Estimating Depth from Monocular Images as Classification Using Deep Fully Convolutional Residual Networks

Yuanzhouhan Cao, Zifeng Wu, Chunhua Shen
School of Computer Science, The University of Adelaide, Australia

***Abstract*—Depth estimation from single monocular images is a key component in scene understanding and has benefited largely from deep convolutional neural networks (CNN) recently. In this article, we take advantage of the recent deep residual networks and propose a simple yet effective approach to this problem.**

We formulate depth estimation as a pixel-wise classification task. Specifically, we first quantize the continuous depth values into a few discrete bins and label the bins according to their depth range. Then we solve the depth prediction problem as classification by training a fully convolutional deep residual network to predict the depth label of each pixel. Performing discrete depth label classification instead of continuous depth value regression allows us to predict a confidence in the form of probability distribution. We further apply fully-connected conditional random fields (CRF) as a post-processing step to enforce local smoothness interactions, which improves the results. We evaluate our approach on both indoor and outdoor datasets and achieve state-of-the-art performance.

CONTENTS

I. INTRODUCTION

Depth estimation is one of the most fundamental tasks in computer vision. Many other computer vision tasks such as object detection, semantic segmentation, scene understanding, can benefit considerably from accurate estimation of depth information. Traditional depth estimation models enforce geometric assumptions and typically rely on hand-crafted features such as SIFT, PHOG, GIST, texton, etc. Recently, computer vision has witnessed a series of breakthrough results introduced by deep convolutional neural networks (CNNs). In this work, we aim to estimate the depth value of each pixel from single monocular images as in [1], [2], [3]. Depth estimation from monocular images is challenging because it is a highly ill-posed problem. One thus usually has to enforce image priors to avoid trivial solutions. A common approach to enforce image priors is to learn these priors from a large amount of labelled data. We follow the fully convolutional networks (FCNs) architecture [2] which has been proven to be a desirable choice for dense prediction problems due to the ability of taking as inputs arbitrarily sized images and outputs convolutional spatial maps.

The success of deep networks can be partially attributed to the rich features captured by the stacked layers. These features are invariant to small local image transformations and thus are desirable for high-level tasks such as image classification and object detection. Recent evidence has shown that the number of stacked layers are of crucial importance. Many vision tasks including depth estimation benefit from an increased number of layers. However, stacking more layers does not necessarily improve performance as the training can become very difficult due to the problem of vanishing gradients. In order to increase the number of layers while maintaining high performance, we employ the deep residual networks originally proposed by He et al. [4] for image-level classification. The layers are formulated as learning residual functions with reference to the layer inputs.

The invariance to local image transformations in CNNs is desirable for classification on one hand. On the other hand, the invariance and the low-resolution of the prediction map can also blur the results of dense prediction, which may not be desirable. In order to cope with the problem of low-resolution prediction maps, Eigen et al. [5] proposed a multi-scale architecture that first predicts a coarse global output and then refines it using a finer-scale local network. Long et al. [6] upsample and concatenate the scores from inter-mediate feature maps. Li et al. [7] and Wang et al. [3] apply hierarchical conditional random fields (CRF) to take various potentials into consideration. In this article, we follow the work in [8] and apply fully-connected CRF to recover the detailed local structure as post processing.

Most existing methods [5], [7], [3], [2] formulate depth estimation as a structured regression task due to the fact of depth values being continuous. However, the possible depth values of different pixels distribute differently and, ideally should be handled differently. It may not be straightforward to take this issue into account in a regression model. In this work, we propose to quantize continuous metric depth values into multiple bins and formulate depth estimation as a *classification* task. This quantization strategy enables us to predict the confidence in the form of probability distribution. Moreover, we can use the predicted confidence in our fully-connected CRFs seamlessly. Pixel depth estimation with low confidence can be improved by other pixels that are connected to it. An overview of our proposed depth estimation model is illustrated in Fig. 1. In practice, we also apply the online bootstrapping proposed in [9], [10] to eliminate the pixels that are easy to label, and at the same time, to mine the hard training examples/pixels.

To sum up, we highlight the main contributions of this work as follows.

- We extend the deep residual network architecture to fully convolutional networks and apply them on single monocular images for depth estimation, which outperform the state-of-the-art results on both indoor and outdoor benchmark datasets.
- We formulate depth estimation as a pixel-level classification task and demonstrate that the proposed formulation can outperform regression that predicts continuous metric depth values.
- We apply fully-connected CRFs as post-processing to compensate for the invariance of deep CNNs to spatial transformations, which helps upsample the prediction map to input size and further improve the performance.

The remaining content of the paper is organized as follows. Section II reviews some relevant work. Then we present the proposed method in Section III. Experiment results are presented in Section IV. Finally, the paper is concluded in Section V.

II. Related Work

Previous methods on depth estimation are mainly based on geometric models. For example, the works of [11], [12], [13] rely on box-shaped models and try to fit the box edges to those observed in the image. These methods are limited to only model particular scene structures and therefore are not applicable for general-scene depth estimations. More recently, non-parametric methods [14] are explored. These methods consist of candidate images retrieval, scene alignment and then depth inference using optimizations with smoothness constraints. These methods are based on the assumption that scenes with semantically similar appearances should have similar depth distributions when densely aligned.

Other methods attempt to exploit additional information. To name a few, the authors of [15] estimate depth through user annotations. The work of [16] performs semantic label prediction before depth estimation. The works of [17], [3] have shown that jointly perform depth estimation and semantic labelling can help each other. Given the fact that the extra source of information is not always available, most of recent work formulates depth estimation as a Markov Random Field (MRF) [18], [19], [20] or Conditional Random Field (CRF) [21] learning problem. These methods managed to learn the parameters of MRF/CRF in a supervised fashion from a training set of monocular images and their corresponding ground-truth depth images. The depth estimation problem then is formulated as a maximum a posteriori (MAP) inference problem on the CRF model.

With the popularity of deep convolutional neural networks (CNN) since the work of [22], some works attempt to solve the depth estimation problem using deep convolutional networks and achieved outstanding performance. Liu et al. [2] presented a deep convolutional neural field model for depth estimation. It learns the unary and pairwise potentials of continuous CRF in a unified deep network. The model is based on fully convolutional networks (FCN) with a novel superpixel pooling method. Note that the CRF there considers only short-range dependencies. In their recent work of [23], they have proposed task-specific loss functions for learning the CRFs' parameters. It enables direct optimization of the quality of the MAP estimates during the course of learning. Eigen et al. [5] proposed a multi-scale architecture for predicting depths, surface normals and semantic labels. The multi-scale architecture is able to capture many image details without any superpixels or low-level segmentation.

Li et al. [7] predict the depths and surface normals of a color image by regression on deep CNN features. It first performs depth estimation on superpixels, then refine the depth estimation from superpixel level to pixel level by inference on a hierarchical CRF. Similarly, the work of [3] also formulated depth estimation in a two-layer hierarchical CRF to enforce synergy between global and local predictions. Experiment results in the aforementioned work reveal that depth estimation benefit from: (a) an increased number of layers in deep networks; (b) obtaining fine-level details. In this work, we take advantage of the successful deep residual networks [4] and formulate depth estimation as a dense prediction task. We also apply fully-connected CRF [24] as post-processing.

The aforementioned CNN based methods formulate depth estimation as a structured regression task due to the continuous property of depth values. However for different pixels in a single monocular image, the possible depth values have different distributions. Depth values of some pixels are easy to predict while others are more difficult. The output of continuous regression lacks this confidence. In [25], Pathak et al. presented a novel structured regression framework that applies constrains on the output space to capture the confidence of predictions. In [26], Kendall et al. proposed a Bayesian neural network for semantic segmentation. It offers a probabilistic interpretation of deep learning models by inferring distributions over the network weights. In this work, we convert continuous depth value regression to discrete depth label classification. With this conversion, we are able to predict depth label with a confidence in the form of a probability distribution. We demonstrate that depth label classification can outperform directly regressing metric depth values. The

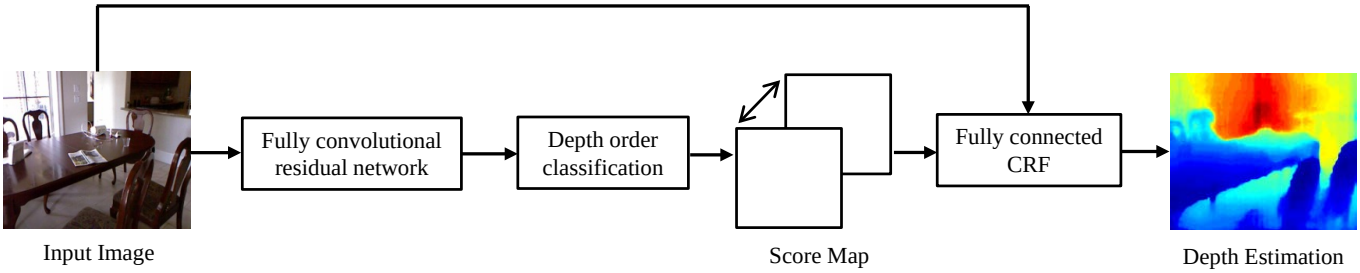

Fig. 1: *An overview of our depth estimation model. It takes as input an RGB image and output score maps. Fully-connected CRFs are then applied to obtain the final depth estimation.*

probability distribution can also be naturally used in our fully-connected CRF as post-processing for further improvement.

III. Proposed Method

In this section, we describe our depth estimation method in detail. We first introduce the network architecture, followed by the introduction of our loss function. Finally, we introduce the fully-connected conditional random fields (CRF) which is applied as post-processing.

A. Network architecture

We formulate our depth estimation as a spatially dense prediction task. When applying CNN to this type of task, the input image is inevitably down-sampled due to the repeated combination of max-pooling and striding. In order to handle this, we follow the fully convolutional network (FCN) which has been proven to be successful in dense pixel labeling. It replaces the fully connected layers in conventional CNN architectures trained for classification with convolution layers. By doing this, it makes the fully convolutional networks capable of taking input of arbitrarily sized images and output a down-sampled prediction map. After applying a simple upsample such as bilinear interpolation, the prediction map are of the same size of input image.

The depth of CNN architecture is of great importance. Much recent work reveals that models based on VGG [27] network outperform those based on shallower AlexNet [22]. However, simply stacking more layers to existing CNN architecture does not necessarily improve its performance due to the notorious problem of vanishing gradients, which hampers convergence from the beginning during training. The recent ResNet model solves this problem by adding skip connections. We follow the recent success of deep residual network with depth up to 152 layers [4], which is about 8× deeper than the VGG network but still having fewer parameters to optimize.

Instead of directly learning the underlying mapping of a few stacked layers, the deep residual network learns the residual mapping. Then the original mapping can be realized by feedforward neural networks with "shortcut connections". Shortcut connections are those skipping one or more layers. In our model, we consider two shortcut connections and the building blocks are shown in Fig. 2. The building block illustrated in Fig. 2(a) is defined as:

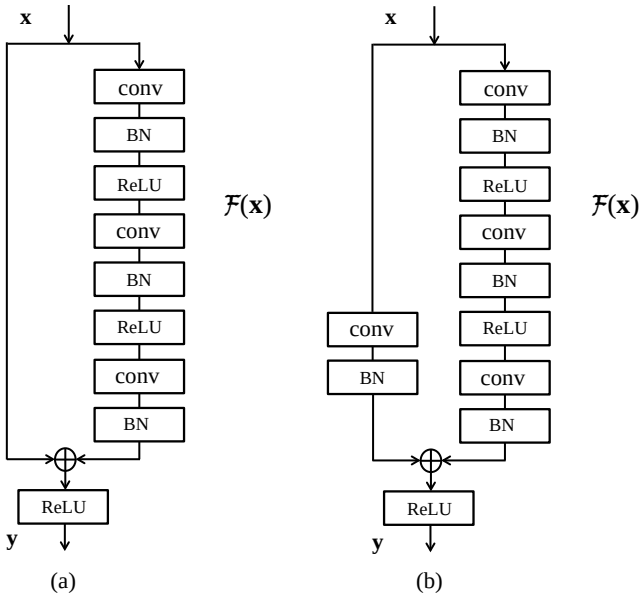

Fig. 2: *Two types of building blocks that can be used in our depth estimation model. (a) building block with identity mapping. (b) building block with linear projection.*

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + \mathbf{x}, \tag{1}$$

where $\mathbf{x}$ and $\mathbf{y}$ are the input and output matrices of stacked layers respectively. The function $\mathcal{F}(\mathbf{x}, \{W_i\})$ is the residual mapping that need to be learned. Since the shortcut connection is an element-wise addition, the dimensions of $\mathbf{x}$ and $\mathcal{F}$ need to be same.

The building block illustrated in Fig. 2(b) is defined as:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}, \{W_i\}) + W_s\mathbf{x}. \tag{2}$$

Comparing to the shortcut connection in Eq. (1), a linear projection W_s is applied to match the dimensions of $\mathbf{x}$ and $\mathcal{F}$.

The overall network architecture of our depth estimation model is illustrated in Fig. 3. The input image is fed into a convolution layer, a max pooling layer followed by 4 convolution blocks. Each convolution block starts with a building block with linear projection followed by different numbers

of building blocks with identity mapping. In this article, we consider two deep residual network architectures with 101 and 152 layers respectively. For network architecture with 101 layers, the number of building blocks with identity mapping in the four convolution blocks, i.e., n_1, n_2, n_3, n_4 in Fig. 3 are 2, 3, 22 and 2 respectively. As for the network architecture with 152 layers, the numbers are 2, 7, 35 and 2. The last five layers are an average pooling layer, three fully-connected layers with channels 1024,512 and N, and a softmax layer, where N is the number of ground-truth labels. Batch normalization and ReLU layers are performed between these convolution layers. Downsampling is performed by pooling or convolution layers that have a stride of 2. That is the first convolution layer, the first max pooling layer, and the second block with linear projection in our network structure. As a result, the output map are downsampled by a factor of 8. During prediction, we perform a bilinear interpolation on the prediction map to make it the same size of input image.

B. Loss function

Since depth values are continuous in real world, traditional methods formulate monocular image depth estimation as a structured regression task. For a given monocular image, CNNs with regression predict the depth value of each pixel.

In this work, we formulate monocular image depth estimation as a classification task, which is simpler to implement than regression. Specifically, we uniformly quantize the continuous depth values into multiple bins in the log-space. Each bin covers a range of depth values and we label the bins according to the range, i.e., the label id of a pixel indicates its distance. During training, there may be tens of thousands of labelled pixels to predict per image and many of them can easily be discriminated from others. In practice, we apply the online bootstrapping proposed in [9] which forces networks to focus on hard training pixels.

We use the pixel-wise multinomial logistic loss function with an "information gain" matrix specifying the contribution of all label pairs, specifically:

$$L = -\frac{1}{\sum_{i=1}^{N} 1\{P(D_i^*|z_i) < t\}} \times \sum_{i=1}^{N}\sum_{D=1}^{B} 1\{P(D_i^*|z_i) < t\} H(D_i^*, D)\log(P(D|z_i)) \tag{3}$$

where $D_i^* \in [1, \ldots, B]$ is the ground-truth depth label of pixel i and B is the total number of discretization bins. $P(D|z_i) = e^{z_{i,D}}/\sum_{d=1}^{B} e^{z_{i,d}}$ is the probability of pixel i labelled with D. $z_{i,d}$ is the output of fully-connected layer in the network.

The "information gain" matrix H is a $B \times B$ symmetric matrix with elements $H(p,q) = \exp[-\alpha(p-q)^2]$ and α is a constant. It encourages depth label that is closer to ground-truth has higher contribution. The indicator function $1\{\cdot\}$ equals to one when the condition inside holds and equals to zero other wise. $t \in (0, 1]$ is a threshold. The indicator function guarantees that only hard pixels contribute to training. In practice, in order to keep a reasonable number of pixels per image, we increase the threshold t accordingly if the current model performs well.

During prediction, we set the depth value of each pixel to be the center of its corresponding bin. The conversion from depth value regression to depth label classification enables us not only to predict the depth label of a pixel, but also the probability distribution among different depth labels. Furthermore, we can utilize the probability distribution to post-process the predicted depth map via fully-connected CRFs.

C. Fully connected conditional random fields

A deep convolutional network typically does not explicitly take the dependency among local variables into consideration. It does so only implicitly through the field of view. That is why the size of field of view is important in terms of the performance of a CNN. As a result, we apply the fully connected CRF proposed in [24] as post-processing. It combines low-level features with depth label scores computed by our deep depth classification network. Specifically, the energy function of the fully-connected CRF is the sum of unary potential U and pairwise potential V:

$$E(\mathbf{D}) = \sum_i U(D_i) + \sum_{i,j} V(D_i, D_j), \tag{4}$$

where $\mathbf{D}$ is the predicted depth labels of pixels and i, j are pixel indices. We use the logistic loss of pixel defined in Eq. (3) as the unary potential, which is

$$U(D_i) = L(D_i) = -\log(P(D_i|z_i)).$$

The pairwise potential is defined as

$$\sum_{i,j} V(D_i, D_j) = \Delta(D_i, D_j) \sum_{s=1}^{M} w_s \cdot k^s(\mathbf{f}_i, \mathbf{f}_j),$$

where $\Delta(D_i, D_j)$ is a penalty term on the labelling. Since the label here indicates depth, we enforce a relatively larger penalty for labellings that are far away from ground-truth. For simplicity, we use the absolute difference between two label values to be the penalty: $\Delta(D_i, D_j) = |D_i - D_j|$. There is one pairwise term for each pair of pixels in the image no matter how far they are from each other, i.e., the model's factor graph is fully connected.

Each k^s is the Gaussian kernel depends on features (denoted as $\mathbf{f}$) extracted for pixel i and j and is weighted by parameter w_s. Following [24], we adopt bilateral positions and color terms, specifically, the kernels are:

$$w_1 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\alpha^2} - \frac{\|I_i - I_j\|^2}{2\sigma_\beta^2}\right) + w_2 \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma_\gamma^2}\right). \tag{5}$$

The first kernel is appearance kernel, which depends on both pixel positions (denoted as p) and pixel color intensities (denoted as I). The second kernel is smoothness kernel and only depends on pixel positions. The hyper parameters σ_α, σ_β, σ_γ control the "scale" of the Gaussian kernels.

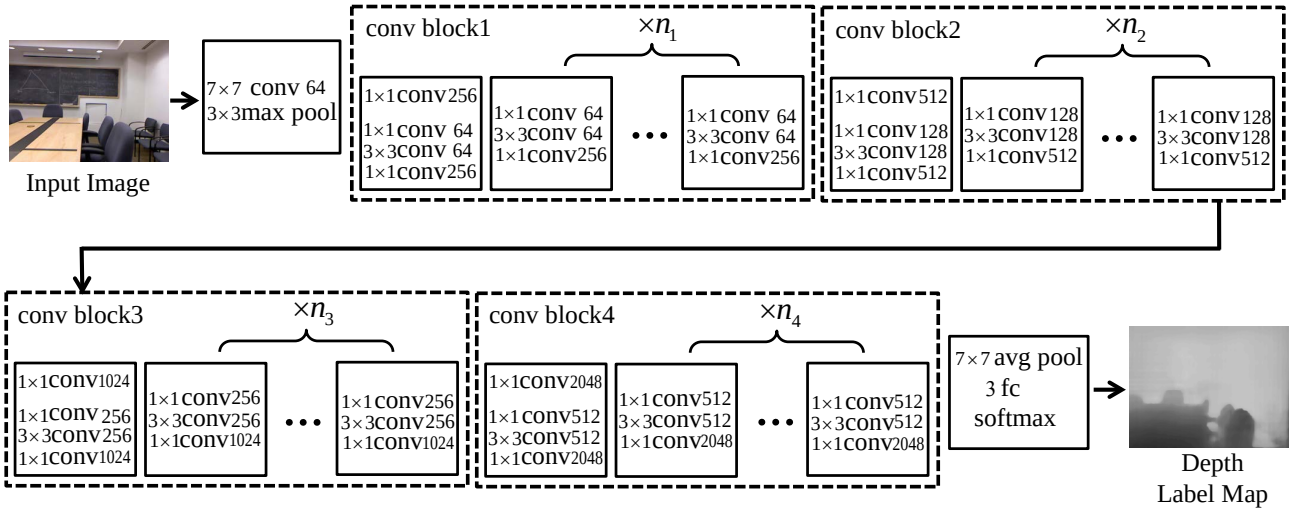

Fig. 3: *Network architecture of our depth estimation model. The input image is fed into a convolution layer, a max pooling layer and 4 convolution blocks. We consider network architectures with 101 and 152 layers. The value of* $[n_1, n_2, n_3, n_4]$ *is* $[2, 3, 22, 2]$ *for the 101-layer network architecture and* $[2, 7, 35, 2]$ *for the 152-layer network architecture. The last 5 layers are an average pooling layer, 3 full-connected layers and a softmax layer. The output map is downsampled by a factor of 8 and we preform bilinear interpolation during prediction.*

IV. Experiments

We test our proposed depth estimation architecture on 3 datasets: the indoor NYUDepth v2 [28] dataset, the outdoor Virtual KITTI [29] dataset and the outdoor Make3D [19] dataset. During training, we apply online data augmentation including random scaling and flipping. We organize our experiments into the following three parts:

(1) We show the effectiveness of our depth discretization scheme and compare our discrete depth label classification with continuous depth value regression.

(2) We evaluate the contribution of different components in our model.

(3) We compare our model with state-of-the-art methods to show that our model performs better in both indoor and outdoor scenes. Several measures commonly used in prior works are applied for quantitative evaluations:

- root mean squared error (rms): $\sqrt{\frac{1}{T}\sum_p (d_{gt} - d_p)^2}$
- average relative error (rel): $\frac{1}{T}\sum_p \frac{|d_{gt} - d_p|}{d_{gt}}$
- average $\log_{10}$ error (log10): $\frac{1}{T}\sum_p |\log_{10} d_{gt} - \log_{10} d_p|$
- accuracy with threshold thr:
  percentage (%) of d_p s.t. $\max(\frac{d_{gt}}{d_p}, \frac{d_p}{d_{gt}}) = \delta < thr$

where d_{gt} and d_p are the ground-truth and predicted depths respectively of pixels, and T is the total number of pixels in all the evaluated images.

A. Depth label classification vs. depth value regression

Discretizing continuous data would inevitably discard some information. In this part, we first show that the discretization degrades the depth estimation model with neglectable amount. We equally discretize the ground-truth depth values of testing images into different number of bins in linear and log space respectively and calculate the three errors as is mentioned above. The result is illustrated in Fig. 5.

We can see from these figures that discretizing depth values in the log space leads to lower error than discretizing in the linear space.

With the number of bins increasing, the errors decrease and converge at some point. Notably, the converged errors are still ~ 10 times lower than the state-of-the-art results in presented Table IV. As for the accuracies, all the accuracies are 100% except for the accuracy with threshold 1.25 when linearly discretizing into 10 bins. So we can convert continuous depth regression to discrete depth label classification and improve depth estimation performance by increasing classification accuracy.

We next compare the depth estimation by depth value regression and depth label classification and show the results in Table I. In this experiment, we apply the network architecture with 101 layers and the parameters are initialized with deep ResNet-101 in [4] which is trained on the ImageNet classification set. We use only one fully-connected layer with channels equal to the number of bins. We train our models on standard training set with 795 images and test on standard 654 test images for fast comparison. For depth value regression, the loss function is standard $L2$ norm which minimizes the squared euclidean norm between predicted and ground-truth depth. The depth values are discretized in the log space. As we can see from the Table I, depth label classification outperforms depth value regression. Moreover, the performance improves with increasing number of discretization bins.

B. Component evaluation

In this part we evaluate the contribution of "information gain" matrix and online bootstrapping in our loss function, as

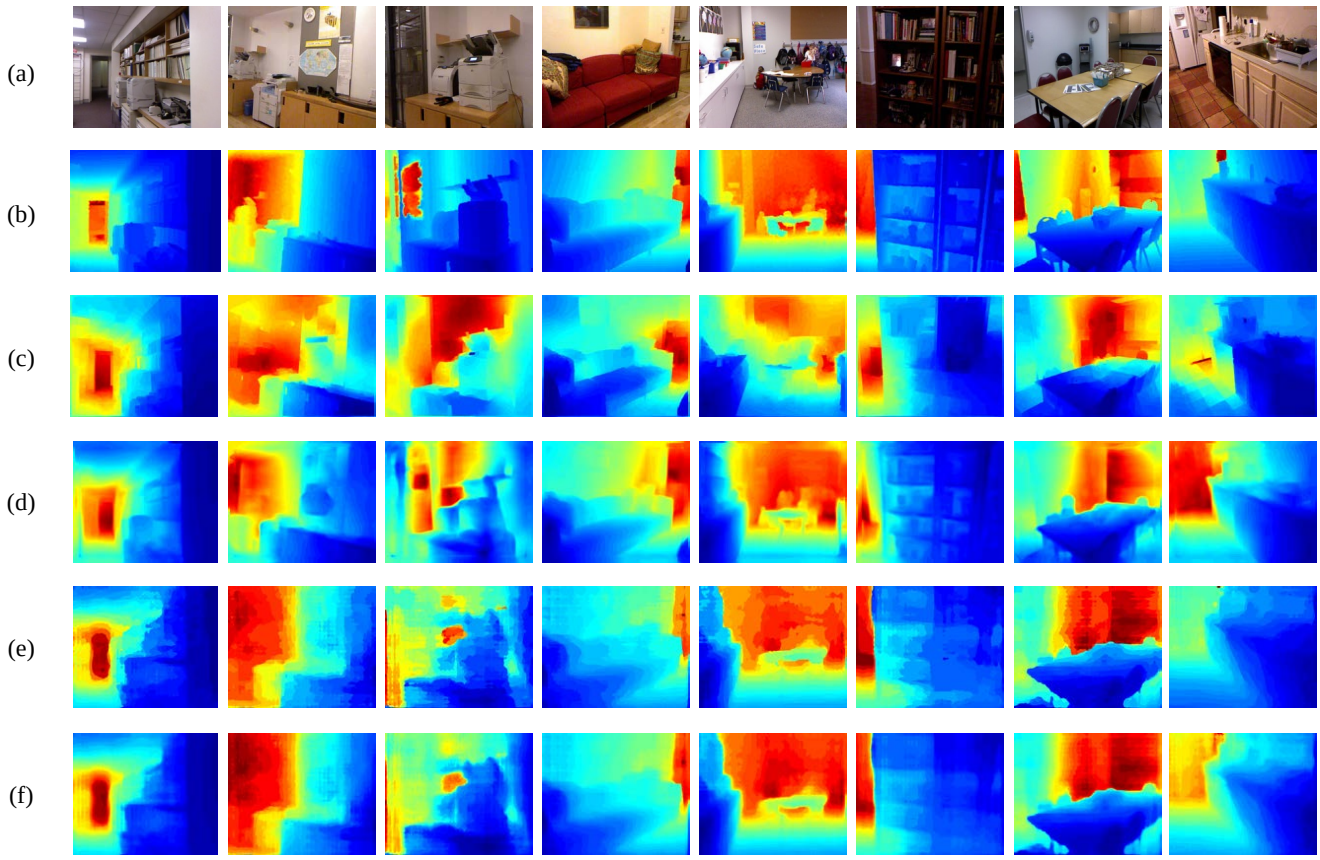

Fig. 4: *Some depth estimation results on the NYUDepth v2 dataset. (a) RGB Input; (b) Ground-truth depth; (c) Results of Liu et al. [2]; (d) Results of Eigen et al. [5]; (e) Results of our model without fully-connected CRFs; (f) Results of our model with fully-connected CRFs.*

TABLE I: *Depth estimation results by continuous depth value regression and discrete depth label classification. The first row is the result by regression. The last 6 rows are results of depth label classification with different number of discretization bins.*

	Accuracy			Error		
	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	rel	log10	rms
Regression	54.2%	83.9%	95.5%	0.249	0.107	0.991
10 bins	60.6%	87.4%	96.2%	0.251	0.098	0.865
20 bins	61.9%	88.1%	96.3%	0.240	0.096	0.851
30 bins	62.2%	88.2%	96.4%	0.240	0.095	0.841
50 bins	62.4%	87.9%	96.4%	0.241	0.095	0.842
70 bins	62.4%	88.2%	96.5%	0.233	0.094	0.842
100 bins	62.3%	88.0%	96.5%	0.239	0.096	0.843

well as the fully connected CRF. Similar to the experiment above, we apply the network architecture with 101 layers and train the models on standard training set with 795 images. The continuous depth values are discretized into 30 bins in the log space. We show the results in Table II. The first row is the result of our depth estimation model trained using the multinomial logistic loss function. Row 2 to row 10 are results of our model trained using the multinomial logistic loss function with an "information gain" matrix where α is the parameter of "information gain" matrix H in Eq. (3). Row 11 to row 13 are results of our model trained using multinomial logistic loss function and online bootstrapping where $t \in (0, 1]$ is a threshold in Eq. (3). Row 14 is the result of our model trained using multinomial logistic loss function together with both "information gain" matrix and online bootstrapping. The last row is the result with fully connected CRF. From the table we can see that both the online bootstrapping and "information gain" matrix help with the depth estimation. Moreover, the "information gain" matrix contributes more than online bootstrapping. And with the fully connected CRF being added as post processing, the depth estimation performance is further improved.

C. State-of-the-art comparisons

In this section, we show the result of depth estimation by our deep fully convolutional residual network and compare with recent methods. We apply the network architecture with 152 layers and the parameters are initialized with deep ResNet-152 in [4].

1) NYUDepth v2 data: We train our model using the entire raw training data specified in the official train/test distribution

TABLE III: *State-of-the-art results of Virtual KITTI dataset. The results are reported on regions with ground-truth less than 80 meters.*

	Accuracy			Error		
	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	rel	log10	rms
Ours	70.3%	97.6%	98.7%	0.214	0.082	6.156

TABLE IV: *Comparison with state-of-the-art of NYUDepth v2 dataset. The first 3 row are results by recent depth estimation models. The last row is the result of our approach.*

	Accuracy			Error		
	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	rel	log10	rms
Wang et al. [3]	60.5%	89.0%	97.0%	0.210	0.094	0.745
Liu et al. [2]	65.0%	90.6%	97.6%	0.213	0.087	0.759
Eigen et al. [5]	76.9%	95.0%	98.8%	0.158	-	0.641
Ours	**80.0%**	**95.6%**	**98.8%**	**0.148**	**0.063**	**0.615**

TABLE V: *Comparison with state-of-the-art of Make3D dataset. The first 5 row are results by recent depth estimation models. The last row is the result of our approach.*

	Error (C_1)			Error (C_2)		
	rel	log10	rms	rel	log10	rms
Saxena et al. [19]	-	-	-	0.370	0.187	-
Liu et al. [16]	-	-	-	0.379	0.148	-
Karsch et al. [14]	0.355	0.127	9.20	0.361	0.148	15.10
Discrete-continuous CRF [21]	0.335	0.137	9.49	0.338	0.134	12.60
DCNF [2]	0.287	0.109	7.36	0.287	0.122	14.09
Ours	**0.206**	**0.082**	**6.81**	**0.209**	**0.085**	**8.31**

TABLE II: *Component evaluation. The first row is the result of 30 bins in Table I. α is the parameter of the "information gain" matrix H in Eq. (3). $t \in (0, 1]$ is a threshold in Eq. (3).*

	Accuracy			Error		
	$\delta < 1.25$	$\delta < 1.25^2$	$\delta < 1.25^3$	rel	log10	rms
plain	62.2%	88.2%	96.4%	0.240	0.095	0.841
$\alpha = 0.4$	62.1%	88.2%	96.5%	0.243	0.095	0.818
$\alpha = 0.5$	62.6%	88.3%	96.4%	0.242	0.095	**0.813**
$\alpha = 0.7$	**63.4%**	**89.0%**	**96.9%**	**0.231**	**0.093**	0.814
$\alpha = 1.0$	63.0%	88.6%	96.6%	0.235	0.094	0.827
$\alpha = 1.2$	62.8%	88.1%	96.4%	0.241	0.095	0.831
$\alpha = 1.4$	62.8%	88.5%	96.6%	0.237	0.094	0.828
$\alpha = 1.6$	62.8%	88.3%	96.4%	0.241	0.095	0.834
$\alpha = 1.8$	62.6%	88.2%	96.5%	0.237	0.094	0.836
$\alpha = 2.0$	62.5%	88.5%	96.7%	0.233	0.094	0.835
$t = 0.3$	**62.8%**	**88.2%**	**96.5%**	**0.239**	**0.094**	**0.833**
$t = 0.4$	62.7%	88.2%	96.4%	0.240	0.095	0.844
$t = 0.6$	62.3%	88.0%	96.3%	0.242	0.095	0.845
$\alpha = 0.7$; $t = 0.3$	63.5%	89.0%	96.6%	0.234	0.093	0.819
$\alpha = 0.7$; $t = 0.3$; CRF	64.6%	89.2%	96.8%	0.232	0.091	0.790

and test on standard 654 test images. We discretize the depth values into 100 bins in log space. The results are reported in Table IV. The first row is the result in [3] which jointly performs depth estimation and semantic segmentation. The second row is the result of deep convolutional neural fields (DCNF) with fully convolutional network and super-pixel pooling in [2]. The third row is the result in [5] which performs depth estimation in a multi-scale network architecture. The last row is depth estimation result by our model. As we can see from the table, our deep fully convolutional residual network with depth label classification achieves the best performance of all the 6 metrics. We also show some qualitative results in Fig. 4, from which we can see our method yield better visualizations in general.

2) Virtual KITTI data: Virtual KITTI [29] is a photo-realistic synthetic video dataset which contains 40 high-resolution videos (17008 frames) generated from five different virtual worlds in urban settings under different imaging and weather conditions. These videos are fully annotated at the pixel level with category, instance, flow, and depth labels. We do not consider the variation of weather conditions and randomly split the remaining frames into 3519 for training and 2859 for test. The depth ground-truth of far away pixels are arbitrarily set to 655.35 meters, however the depth value of most pixels are within 80 meters. We thus discretize the regions with ground-truth less than 80 meters into 50 bins in log space and report the depth estimation results in Table III. We also discretize the entire image into 120 bins and show some qualitative estimation results in Fig. 6.

3) Make3D data: The Make3D dataset contains 534 images depicting outdoor scenes, with 400 images for training and 134 images for test. Due to the limited range and resolution of the depth sensor, the far away pixels are arbitrarily set

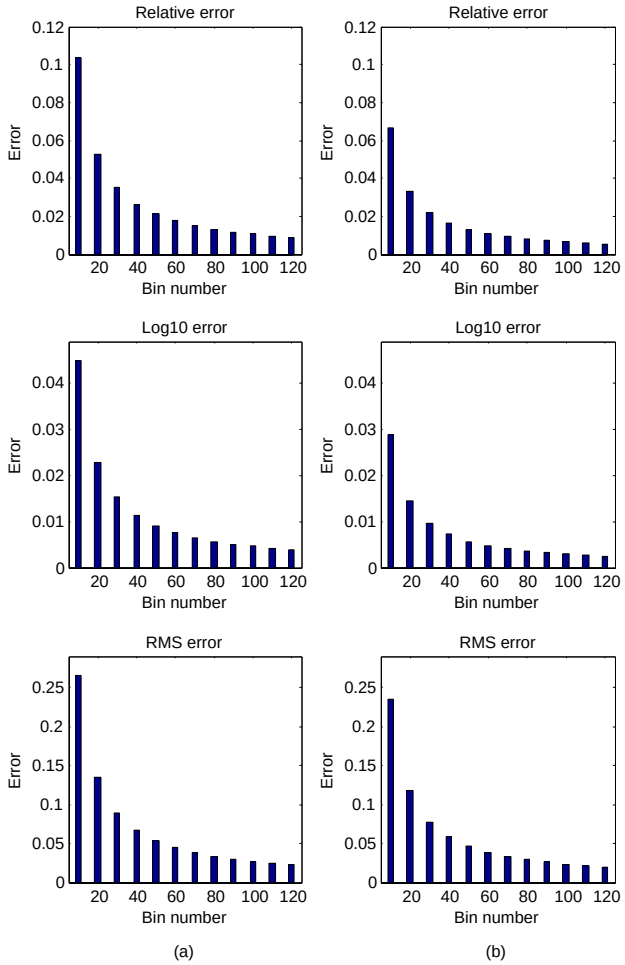

Fig. 5: *Quantitative evaluations of discretized ground-truth depth values. (a): errors of ground-truth depth values discretized in linear space. (b): errors of ground-truth depth values discretized in the log space.*

to 81 meters. Following the conventional setting in [21], we report errors based on two different criteria: (C_1) Errors are computed in the regions with ground-truth less than 70 meters; (C_2) Errors are computed in the entire image. During training, we exclude the pixels with ground-truth over 80 meters to reduce the effect of meaningless candidates in sky regions. We discretize the depth values into 50 bins in log space. In order to overcome overfit, the model is pretrained on Virtual KITTI training images. The results are reported in Table V. As we can see from the table, the results by our approach outperform the state-of-the-art results markedly. Some qualitative results are illustrated in Fig. 7.

V. Conclusion

We have presented a deep fully convolutional residual network architecture for depth estimation from single monocular images. The proposed model makes use of the recent deep residual networks. We also discretize continuous depth values into different bins and formulate depth estimation as a discrete classification problem. We show that it is possible that the discretization even outperforms depth estimation by continuous regression. We also show that a fully-connected CRF post-processing step can further improved the performance.

Note that the proposed network can be further improved by applying the techniques that have been previously explored. For example, it is expected that

- Multi-scale inputs as in [5] would improve our result.
- Concatenating the mid-layers' outputs may better use the low-, mid-layers information as in [30].
- Upsampling the prediction maps as in [6] would be beneficial too.

We leave these directions in our future work.

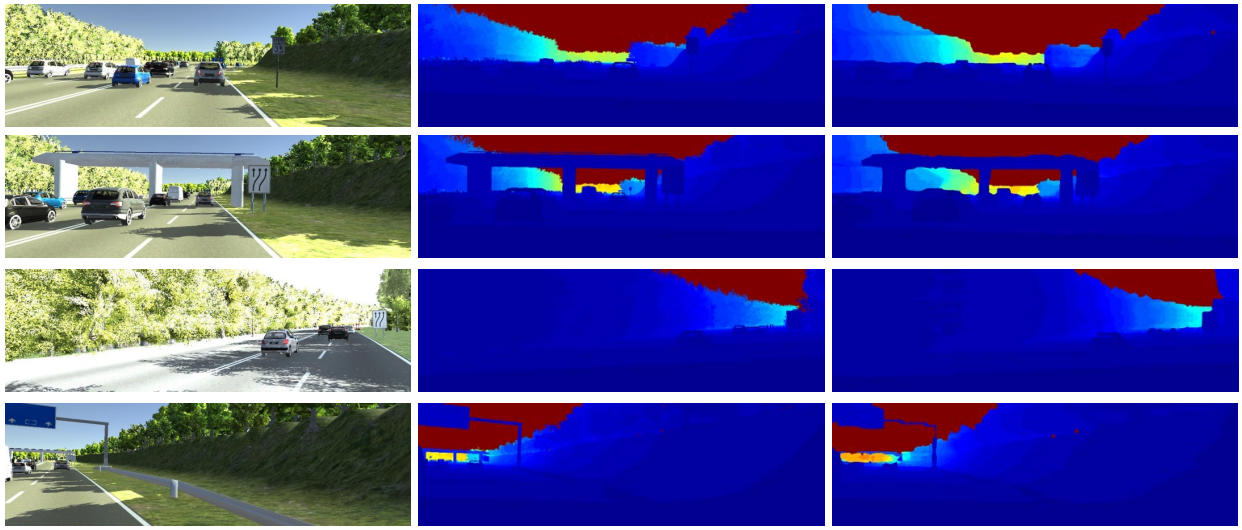

Fig. 6: *Some depth estimation results on the Virtual KITTI dataset. (a) RGB Input; (b) Ground-truth depth; (c) Estimated depth by our approach.*

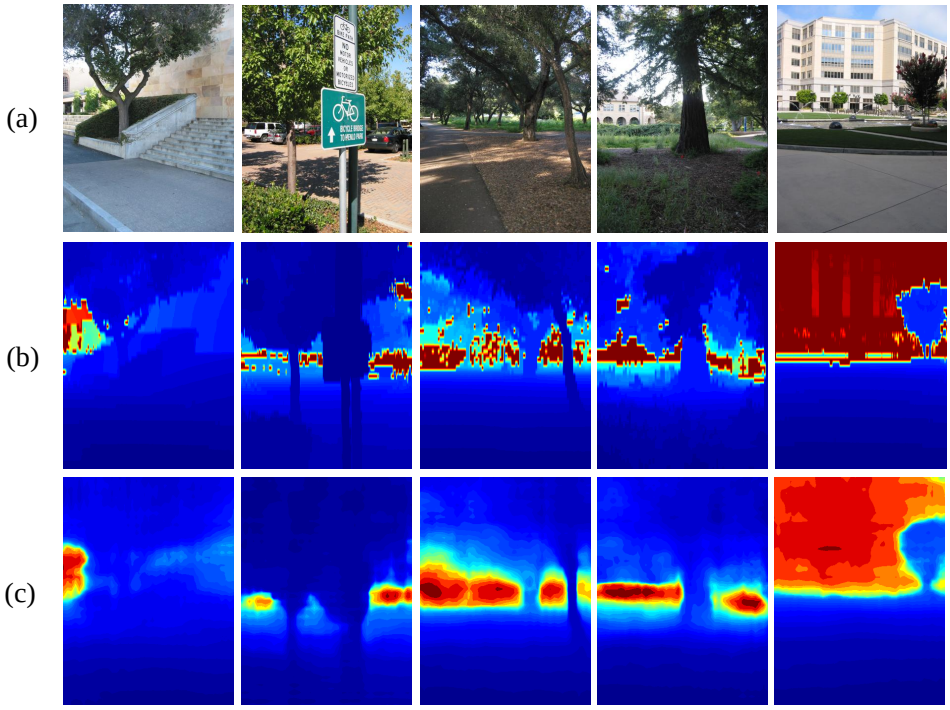

Fig. 7: *Some depth estimation results on the Make3D dataset. (a) RGB Input; (b) Ground-truth depth; (c) Estimated depth by our approach.*

REFERENCES

[1] F. Liu, C. Shen, and G. Lin, "Deep convolutional neural fields for depth estimation from a single image," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2015. [Online]. Available: http://arxiv.org/abs/1411.6387

[2] F. Liu, C. Shen, G. Lin, and I. D. Reid, "Learning depth from single monocular images using deep convolutional neural fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2016.

[3] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille, "Towards unified depth and semantic prediction from a single image," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, June 2015.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016.

[5] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2015.

[6] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015.

[7] 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 *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015.

[8] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *Proc. IEEE Int. Conf. Learn. Rep.*, 2015.

[9] Z. Wu, C. Shen, and A. van den Hengel, "High-performance semantic segmentation using very deep fully convolutional networks," 2016. [Online]. Available: https://arxiv.org/pdf/1604.04339.pdf

[10] ——, "Bridging category-level and instance-level semantic image segmentation," 2016. [Online]. Available: https://arxiv.org/abs/1605.06885

[11] V. Hedau, D. Hoiem, and D. Forsyth, "Thinking inside the box: Using appearance models and context based on room geometry," in *Proc. Eur. Conf. Comp. Vis.*, 2010, pp. 224–237.

[12] A. Gupta, M. Hebert, T. Kanade, and D. M. Blei, "Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces," in *Proc. Adv. Neural Inf. Process. Syst.*, 2010.

[13] A. G. Schwing and R. Urtasun, "Efficient exact inference for 3d indoor scene understanding," in *Proc. Eur. Conf. Comp. Vis.*, 2012.

[14] K. Karsch, C. Liu, and S. B. Kang, "Depthtransfer: Depth extraction from video using non-parametric sampling," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2014.

[15] B. C. Russell and A. Torralba, "Building a database of 3d scenes from user annotations." in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2009.

[16] B. Liu, S. Gould, and D. Koller, "Single image depth estimation from predicted semantic labels," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2010.

[17] L. Ladicky, J. Shi, and M. Pollefeys, "Pulling things out of perspective," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2014.

[18] A. Saxena, A. Ng, and S. Chung, "Learning Depth from Single Monocular Images," *Proc. Adv. Neural Inf. Process. Syst.*, 2005.

[19] A. Saxena, M. Sun, and A. Y. Ng, "Make3D: Learning 3d scene structure from a single still image," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2009.

[20] A. Saxena, S. H. Chung, and A. Y. Ng, "3-d depth reconstruction from a single still image," *Int. J. Comp. Vis.*, 2007.

[21] M. Liu, M. Salzmann, and X. He, "Discrete-continuous depth estimation from a single image," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2014.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[23] F. Liu, G. Lin, and C. Shen, "Discriminative traininig of deep fully-connected continuous CRFs with task-specific loss," 2016. [Online]. Available: https://arxiv.org/abs/1601.07649

[24] P. Krähenbühl and V. Koltun, "Efficient inference in fully connected crfs with gaussian edge potentials," in *Proc. Adv. Neural Inf. Process. Syst.*, 2011.

[25] D. Pathak, P. Krähenbühl, S. X. Yu, and T. Darrell, "Constrained structured regression with convolutional neural networks," 2016. [Online]. Available: http://arxiv.org/abs/1511.07497/

[26] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," 2015. [Online]. Available: http://arxiv.org/abs/1511.02680

[27] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Proc. Int. Conf. Learn. Rep.*, 2015.

[28] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Proc. Eur. Conf. Comp. Vis.*, 2012.

[29] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtual worlds as proxy for multi-object tracking analysis," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016.

[30] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015.