

# Geometry-Aware Distillation for Indoor Semantic Segmentation

Jianbo Jiao<sup>1</sup>, Yunchao Wei<sup>2\*</sup>, Zequn Jie<sup>3</sup>, Honghui Shi<sup>4</sup>, Rynson Lau<sup>5</sup>, Thomas S. Huang<sup>2</sup>

<sup>1</sup>Department of Engineering Science, University of Oxford

<sup>2</sup>UIUC, <sup>3</sup>Tencent AI Lab, <sup>4</sup>IBM Research, <sup>5</sup>City University of Hong Kong

jianbo@robots.ox.ac.uk, {wychao1987, zequn.nus, shihonghui3}@gmail.com

rynson.lau@cityu.edu.hk, t-huang1@illinois.edu

## Abstract

It has been shown that jointly reasoning the 2D appearance and 3D information from RGB-D domains is beneficial to indoor scene semantic segmentation. However, most existing approaches require accurate depth map as input to segment the scene which severely limits their applications. In this paper, we propose to jointly infer the semantic and depth information by distilling geometry-aware embedding to eliminate such strong constraint while still exploiting the helpful depth domain information. In addition, we use this learned embedding to improve the quality of semantic segmentation, through a proposed geometry-aware propagation framework followed by several multi-level skip feature fusion blocks. By decoupling the single task prediction network into two joint tasks of semantic segmentation and geometry embedding learning, together with the proposed information propagation and feature fusion architecture, our method is shown to perform favorably against state-of-the-art methods for semantic segmentation on publicly available challenging indoor datasets.

## 1. Introduction

Semantic segmentation that infers semantic labels of every pixel in an indoor scene is a fundamental yet challenging problem in computer vision. Obtaining better scene understanding by semantic segmentation benefits many applications like robotics, visual SLAM and virtual/augmented reality. Compared with RGB image based methods, RGB with depth (RGB-D) based methods can leverage additional 3D geometry information from the scene to effectively tackle ambiguities that are challenging for 2D appearance solely approaches, *e.g.*, some *pillows* on a *bed* with similar color as the *bed* (Figure 1).

Prior RGB-D semantic segmentation methods have achieved promising performance by incorporating the depth

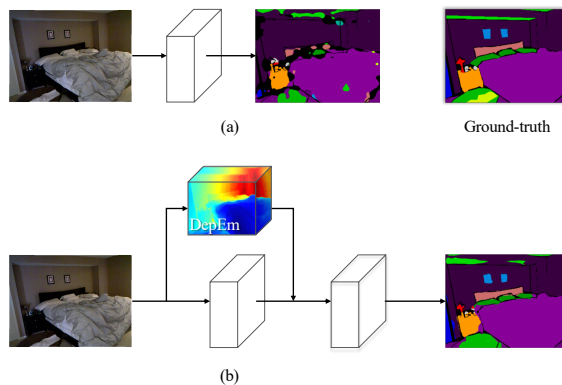


Figure 1. Illustration on the effectiveness of (b) distilling geometry-aware depth embeddings compared to (a) the traditional RGB, for semantic segmentation.

information [33, 12, 34, 13, 11, 6, 32]. There are mainly two types of approaches to achieve the goal: using hand-crafted features or deploying CNN-based models. Earlier works use handcrafted image descriptors like SIFT or HOG to extract features from the RGB images. Some specially designed features (*e.g.*, surface normal [34], depth gradient or spin [33]) for depth description are also used to assist the final segmentation. For CNN-based models, fully convolutional networks (FCN) [28] greatly improve the performance for semantic segmentation due to the highly representative features extracted by a learning manner. In general, two separate FCNs are utilized to extract features from RGB and depth channels, followed by a simple fusion [28, 11] for semantic label prediction.

All the aforementioned approaches require ground-truth depth map associated with the input RGB images. However, collecting depth data from the scene is not convenient compared to RGB image capturing and the alignment between depth and RGB is a challenging problem itself. Therefore, we are interested in such a question: is that possible to incorporate the geometry information for semantic segmentation with only one single RGB image as input?

Some research attempts [37] have been proposed to pre-

\*Corresponding author

dict depth information to help the semantic segmentation task through a multi-task network. The performance gain of semantic segmentation is mainly from simply post-fusion strategy. We argue that the depth-aware features are not well exploited by such approaches and we aim to learn better geometry-aware feature representations implicitly.

In this work, we propose to distill/extract geometry-aware information by learning dense depth embeddings in a joint reasoning framework, for single RGB image semantic segmentation. Instead of directly taking a depth image as input, the proposed model distills depth embeddings that can guide semantic segmentation together with RGB input. In the proposed framework, such learned embeddings are fused with the feature from 2D appearance by a proposed geometry-aware propagation block, which leverages the geometry affinity to guide semantic propagation. Furthermore, we find the segmentation result tends to lack details especially near object boundaries. An incremental cross-scale fusion scheme is proposed, in feature space, to further enrich the structure details consequently. Some objects may have very similar 2D appearance that cannot be well discriminated. With the proposed model, the 3D geometry information can be well embedded into the learned features, leading the prediction to be both semantic consistent and geometrical consistent. As shown in Figure 1 (a), the *pillows* are difficult to be segmented out solely based on 2D features, while with the learned embeddings (DepEm) they can be well classified due to their different 3D geometry information to the surroundings (Figure 1(b)). The shape of the *bed* also benefits from the learned embeddings, which reveals the effectiveness of the distilled geometry information. The key idea of our approach is predicting semantic labels from one single RGB image, while taking 3D geometry information into consideration implicitly. The main contributions of this paper are summarized as:

- We propose a novel approach that distills geometry-aware embeddings via implicit depth inference, which effectively guides scene segmentation over RGB input.
- The proposed joint framework enables effective information fusion between depth and semantic labels and is end-to-end trainable.
- Our model achieves state-of-the-art performance on challenging indoor semantic segmentation datasets of NYU-Dv2 and SUN RGBD.

## 2. Related Work

**RGB Semantic Segmentation.** Owe to the great success of deep learning in high-level vision tasks [21, 35], most recent semantic segmentation approaches take advantage of CNNs. In [28] an FCN structure that performs pixel-wise classification by end-to-end training is proposed. Later on,

FCN turned to be the basic structure for most CNN-based methods [5, 25, 24, 9, 1]. Chen *et al.* [5] employed an atrous convolution to expand the receptive field followed by fully connected Conditional Random Fields (CRFs). Besides taking as a post-processing step, CRFs has also been integrated into the networks [40, 2, 4] to enrich more detailed prediction. To overcome the low-resolution limitation of FCNs, some works [30, 3] proposed to use upconvolution (also known as deconvolution) layers to upsample the features layer by layer. In [30], the authors made the first attempt to learn deconvolution network on top of convolutional layers and combined instance-wise segmentation for the final result. Another work [3] further added connections from encoder to decoder by pooling indices. Another type of approach [28, 3, 24] leveraged multi-level/scale features to predict the final result. Li *et al.* [24] proposed a network that iteratively combines multi-level features and demonstrate large improvement.

**RGB-D Semantic Segmentation.** Different from 2D RGB settings, RGB-D semantic segmentation is augmented with the 3D geometry information provided by the depth map. Early works [33, 12, 21, 34] design handcrafted features tailored for RGB with depth information. Extracted features are further fed into another model to do the classification. Similar to recent RGB semantic segmentation, CNN also benefits RGB-D approaches. Some methods [11, 28] treated depth map as an additional channel for the input with RGB image, while more recent works [13, 28, 32, 31, 38] first encoded the depth into a three-dimensional HHA (horizontal disparity, height above ground, and angle with gravity) image. In addition to RGB semantic segmentation, Long *et al.* [28] also reported their performance on RGB-D data, by separately predicting features for the two modalities and fusing for final prediction. Eigen and Fergus [11] leveraged depth and RGB images in a global-to-local framework. Li *et al.* [23] fused the depth and RGB features by LSTM layers. Cheng *et al.* [6] used two separate locality-sensitive DeconvNets to combine HHA and RGB features and recover sharp boundaries. Park *et al.* [31] extended the RefineNet [24] for RGB-D semantic segmentation. Qi *et al.* [32] proposed a 3D graph neural network that builds on 3D point cloud from the depth map, to predict the semantic labels of each pixel. These methods all taking the ground-truth depth map as input.

Alternatively, some efforts have been made to leverage 3D geometry information without feeding ground-truth depth into the model. Wang *et al.* [37] proposed a joint framework to predict both depth and semantic maps, followed by a hierarchical CRF. Only the final layer of a CNN is used to predict the semantics and the hierarchical CRF is computationally expensive. Hoffman *et al.* [16] proposed to hallucinate different modality during training but for de-

tection task. Kokkinos [20] proposed a CNN called UberNet that jointly handles several vision tasks (*e.g.*, boundaries, surface normals, semantic segmentation, *etc.*), which achieved competitive performance as well as high efficiency. In this way, semantic segmentation benefits from several vision tasks including the surface normal encoding geometry information. However, information sharing between different tasks are not well explored.

### 3. Geometry-Aware Distillation

This section presents the proposed framework on geometry-aware distillation to implicitly improve the semantic segmentation performance. The whole network is trained end-to-end by a joint objective function.

#### 3.1. Learning Depth-Aware Embedding

The goal of this work is to leverage the geometry (depth here) information for semantic segmentation without explicitly requiring depth annotation as inputs. An intuitive approach for such a purpose is to first predict a depth map from the input RGB image and then incorporate the depth information into the traditional RGB-D segmentation pipeline [13, 11]. Instead of taking such sequential and *ad-hoc* solution, we propose to learn a depth-aware embedding from the RGB image and simultaneously perform semantic segmentation. We define the depth-aware embedding as the representation encoding both depth information and pixel affinities at the semantic level.

Concretely, given an RGB image  $I$  with pixels  $I_i \in \mathbb{R}^{R,G,B}$ , the depth-aware embedding is from a learnable projection function  $g(I_i)$  that transforms the RGB pixels into a higher-dimension space with embedded corresponding features. Then the embedding learning can be modeled as an optimization problem:

$$\min_g \sum_{i=1}^n E(g(I_i); D_i^*) + s(I_i), \quad (1)$$

where  $E(x, x^*)$  is a data fitting term and  $D^*$  is the ground-truth offering depth information to be embedded through the projection. The second term  $s(x) = E(g'(x), x^*)$  is a semantic one aiming to embed the semantic information, where  $g'(\cdot)$  partially shares weights with  $g(\cdot)$ . Here  $n$  is the total number of pixels. In order to obtain a good projection  $g$ , we parameterize it by a deep neural network model and the embedding can be optimized by backpropagation. Thus,  $g$  is defined as  $f_\theta$  where  $f$  is a deep CNN with parameters  $\theta$ . Then the optimization (Eq. 1) is re-formulated as,

$$\min_\theta \sum_{i=1}^n E(f_\theta(I_i); D_i^*) + s_\theta(I_i), \quad (2)$$

where  $s$  is parameterized by the same network model  $\theta$ .

### 3.2. Geometry-Aware Guided Propagation

After learning the embeddings, we deploy them to improve semantic segmentation. Here we propose a geometry-aware propagation (GAP) approach to leverage the learned embeddings as guidance. In this way, the depth embedding acts as an affinity guidance providing geometry information for better grouping the semantic features beyond 2D appearance space. Given a point  $i$  in the embedding space with its neighboring point  $j \in \mathcal{N}(i)$ , for the corresponding feature point  $p_j$  at location  $j$  in the score map used to predict semantic labels, the propagation output  $q_i$  at location  $i$  can be formulated as,

$$q_i = \frac{\sum_j W_{ij}(G_{em})p_j}{\sum_j W_{ij}}, \quad (3)$$

where  $G_{em} = f_\theta(I_i)$  is the learned depth embedding and  $W_{ij}$  is the propagation weights derived from the geometry guidance  $G_{em}$ . Since  $W_{ij}$  represents the geometric affinity in embedding space, here we define it as a dot-product of decoupled embeddings as,

$$W_{ij} = \eta(G_{em}^i) \cdot \varphi(G_{em}^j), \quad (4)$$

where  $\eta$  and  $\varphi$  decouple the original embedding into two sub-embeddings, respectively. In order to cope with the dimension variation during the propagation, the semantic feature is further projected to an embedding space accordingly by  $\delta(p_j)$ . In particular, the propagation weights are designed by several convolution units which can be automatically learned by backpropagation. Specially, the original semantic feature is added back to the propagated result, to avoid interruption during the whole propagation. Then the proposed GAP block is defined as,

$$q_i = \frac{\sum_j \eta(G_{em}^i) \cdot \varphi(G_{em}^j) \cdot \delta(p_j)}{\sum_j W_{ij}} + p_i. \quad (5)$$

#### 3.3. Network Architecture

In this section, we propose a specially designed deep CNN architecture to distill the geometry-aware information with guided propagation and pyramid feature fusion, for semantic segmentation.

As shown in Figure 2, the proposed network consists of five components: shared backbone network, semantic segmentation branch, depth embedding branch, geometry-aware propagation block, and skip pyramid fusion block. The proposed network globally follows an encoder-decoder structure, with multi-task predictions. The network weights of the encoder backbone part are shared between the following two tasks. For the decoder part, the upper branch predicts semantic labels while the lower branch learns the depth embeddings by predicting depth map. Features from

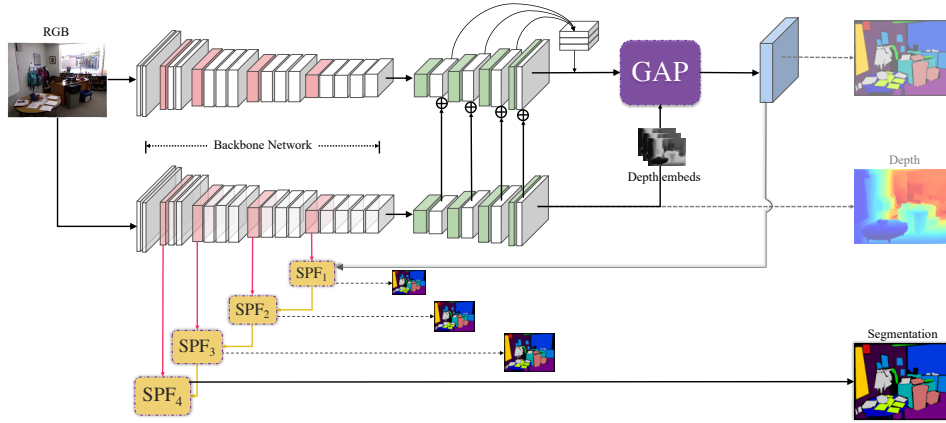


Figure 2. Overview of the proposed network architecture. Top part shows two parallel encoder-decoder networks predicting semantic labels and depth information respectively. The weights of backbone encoders are shared with each other while decoders are task-specific. At the end of decoders, the learned embeddings are used to improve the semantic features by a geometry-aware propagation (GAP) block. In the bottom part, the distilled semantic features (blue block) are further fused with multi-level feature maps from the backbone to improve the final semantic segmentation performance.

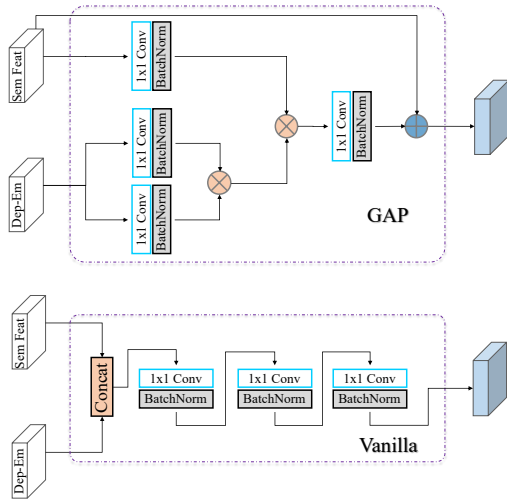


Figure 3. Detailed structure of the proposed GAP block (top) in Figure 2, and a vanilla convolution block (bottom).  $\otimes$  denotes dot-product while  $\oplus$  denotes element-wise summation.

the depth branch are propagated (by summation) to the semantic branch to provide multi-scale depth guidance (Feat-Prop). In the decoders, different scale features are also propagated to enrich the final layer output. Each layer in the decoders is upsampling followed by convolution. A geometry-aware propagation block (GAP) is applied at the end of the semantic branch to improve the quality of semantic features with the learned embeddings as guidance. The distilled output is further refined by combining with multi-level feature maps from the backbone network through the skip pyramid fusion block (SPF). The score map from the bottom SPF block is utilized for the final semantic label prediction. The semantic supervision is performed on both the rightmost distilled features and each level of the side output

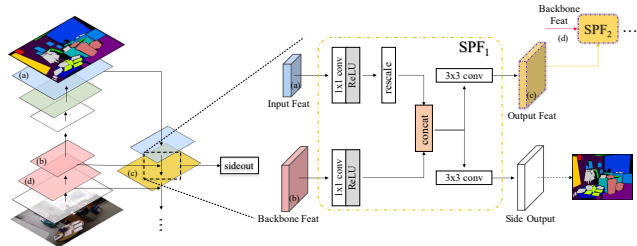


Figure 4. Structure of the proposed SPF block in Figure 2.

from SPFs. The corresponding depth map acts as a supervision for learning the embeddings. The whole network is trained end-to-end by a joint objective function (detailed in the objective function section).

**Geometry-Aware Propagation.** The proposed geometry-aware propagation is implemented by several convolution layers followed by batch normalization and element operations in our network. The detailed structure of the GAP is shown in Figure 3. The depth embedding is first sent into two conv units to achieve the geometric affinity. Then the geometric affinity is treated as a guidance to fuse with the semantic features. Finally, the original semantic features are combined with the fused information for the output, shown as the blue block in Figure 3. The whole propagation maintains the dimension of the semantic features. A vanilla convolution block is also shown for comparison, which fuses the depth and color features without knowing the underlying combining strategy. In contrast, depth is explicitly designed to act as a guidance for feature fusion in the GAP.

**Skip Pyramid Fusion.** As much detail information may be lost when the image passes the encoder and decoder, we

try to enrich and recover more details in the final semantic feature maps as follows. Inspired by the feature pyramid networks [26] for object detection, we propose to leverage multi-level features from the encoder backbone through skip connections. Due to the bottleneck feature space between the encoder and decoder is the most sparse one with least details, the final feature maps recovered by the decoder barely contain useful details. Thus we turn to the encoder part to seek more information. The structure of the skip pyramid fusion (SPF) block is shown in Figure 4. The first SPF (*i.e.*, SPF<sub>1</sub>) takes the distilled feature as input, which passes through a 1×1 convolution and is concatenated with the feature map from the encoder backbone after proper resizing. The combined features are propagated to another SPF after a 3×3 convolution. At the same time, each SPF predicts a side output for semantic segmentation.

### 3.4. Objective Function

For semantic segmentation, most methods utilize cross-entropy to measure the difference between the prediction and ground-truth labels. However, for existing semantic segmentation datasets, *e.g.*, NYU-Dv2 [34], SUN RGBD [36], distribution of the semantic labels is dramatically imbalanced. Very few semantic labels dominate the whole dataset, leaving only a few samples for a great number of labels. We plot the distribution of the above two datasets in Figure 5. As shown in the distribution, some categories (*wall*, *floor*, *etc.*) have much more samples than the others (*bath tub*, *bag*, *etc.*). This will bias the learning to those dominant samples and result in low accuracy for the minority categories. To alleviate the data imbalance issues, we extend a recent loss function [27] proposed for object detection, to our semantic segmentation task as follows,

$$L_s = - \sum_i \sum_c (1 - p_{i,c})^2 \times \ell^* \times \log(p_{i,c}), \quad (6)$$

where  $i$  indexes the pixel,  $c \in 1, 2, 3, \dots$  denotes the category.  $p_{i,c}$  is the predicted probability of pixel  $i$  belonging to category  $c$ .  $\ell^*$  is the ground-truth label. By such loss, the hard samples contribute more than the easy ones. For example, if the prediction for one pixel is correct, *e.g.*,  $p = 0.9$ ,  $L_s$  weights less with  $(1 - p)^2 = 0.01$ ; if a pixel is wrongly predicted with  $p = 0.1$ , the weight will be as large as 0.81.

In addition to the semantic supervision, learning the depth-aware embeddings requires supervision from depth domain. Following a state-of-the-art algorithm [22] for depth estimation, we use the berHu loss for our depth supervision defined as,

$$L_d = \sum_i \begin{cases} |d_i - D_i^*|, & |d_i - D_i^*| \leq \delta \\ \frac{(d_i - D_i^*)^2 + \delta^2}{2\delta}, & |d_i - D_i^*| > \delta \end{cases}, \quad (7)$$

where  $d_i$  is the predicted depth derived from the embeddings  $g(I_i)$  for pixel  $i$ ,  $\delta = 0.2 \cdot \max_c(|d_i - D_i^*|)$ . Then the

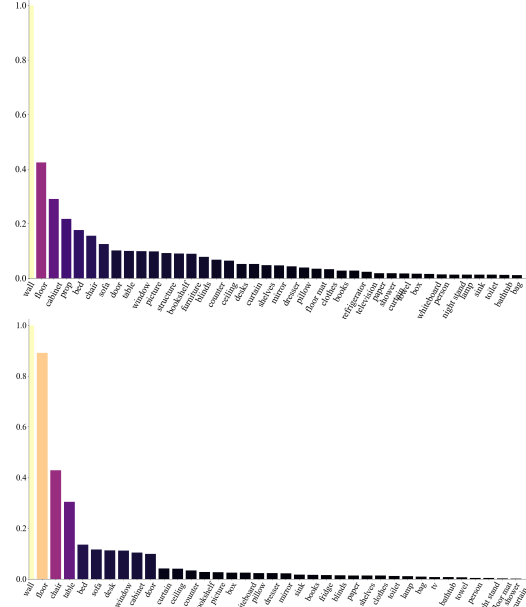


Figure 5. The distribution of semantic labels on NYU-Dv2 (top) and SUN RGBD (bottom). Horizontal axis shows the semantic labels, while vertical axis shows the relative proportion of samples.

loss  $L_d$  acts as the data term in the optimization of embedding learning shown in Equation (1).

Together with the loss  $L_{sk}$  ( $L_s$  at SPF <sub>$k$</sub> ) for semantic prediction at intermediate layers (aggregate for  $K$  layers), our final joint loss function is formulated as,

$$L = L_s + L_d + \sum_{k=1}^K L_{sk}. \quad (8)$$

## 4. Experiments

### 4.1. Datasets and Metrics

We evaluate our method mainly on two public datasets: the popular NYU-Dv2 [34] dataset and a large-scale SUN RGBD [36] dataset. The NYU-Dv2 dataset consists of 1449 image samples with both dense semantic labels and depth information, from 464 different scenes captured by Microsoft Kinect. The standard split by [34] involves 795 images from 249 scenes for training and 654 images from 215 scenes for testing. The semantic labels cover nearly 900 different categories. Following [12], we use the projected 40-category labels in our experiment. The SUN RGBD dataset consists of 10335 RGB-D image pairs also with pixel-wise semantic labels, come from existing RGB-D datasets [34, 17, 39] as well as newly captured data. We use the standard training/testing split [36] of 5285/5050 in our experiment with 37-category semantic labels.

To evaluate the performance of our method, we employ the commonly used metrics in recent works [6, 31, 32, 12,

Table 1. Comparison with state-of-the-arts on NYU-Dv2 dataset. Percentage (%) of pixel accuracy, mean accuracy, and mean IoU are shown for evaluation.

| Method                       | Input | PixAcc.     | mAcc.       | mIoU        |
|------------------------------|-------|-------------|-------------|-------------|
| Gupta <i>et al.</i> [12]     | RGBD  | 60.3        | 35.1        | 28.6        |
| Eigen & Fergus [11]          | RGBD  | 65.6        | 45.1        | 34.1        |
| FCN [28]                     | RGBD  | 65.4        | 46.1        | 34.0        |
| Lin <i>et al.</i> [25]       | RGB   | 70.0        | 53.6        | 40.6        |
| Mousavian <i>et al.</i> [29] | RGB   | 68.6        | 52.3        | 39.2        |
| Cheng <i>et al.</i> [6]      | RGBD  | 71.9        | 60.7        | 45.9        |
| Gupta <i>et al.</i> [13]     | RGBD  | 60.3        | -           | 28.6        |
| Deng <i>et al.</i> [10]      | RGBD  | 63.8        | -           | 31.5        |
| RefineNet [24]               | RGB   | 73.6        | 58.9        | 46.5        |
| 3DGNN [32]                   | RGBD  | -           | 55.7        | 43.1        |
| D-CNN [38]                   | RGBD  | -           | 61.1        | 48.4        |
| RDFNet [31]                  | RGBD  | <u>76.0</u> | <u>62.8</u> | <u>50.1</u> |
| Proposed                     | RGB   | <b>84.8</b> | <b>68.7</b> | <b>59.6</b> |

28]: pixel accuracy (PixAcc.), mean accuracy (mAcc.), and mean intersection over union (mIoU).

## 4.2. Implementation Details

We implement our network with the PyTorch framework on an 8-GPU machine. We use the pre-trained ResNet-50 [14] as our backbone network and four up-convolution blocks for the decoder branches. The network parameters except the backbone are initialized by the method in [15]. We use Adam solver [19] with  $(\beta_1, \beta_2) = (0.9, 0.999)$  to optimize the network. Gradient clipping is utilized for the semantic branch and the SPF blocks. The learning rates are initialized as  $10^{-5}$  for the backbone and  $10^{-2}$  for the other parts, and divided by 10 for every 40 epochs. The batch size is set to 8. During training, the images are first down-sampled to  $320 \times 240$  with data augmentation applied. We use random horizontal flipping, cropping, and image color augmentation (*e.g.*, gamma shift, brightness shift, *etc.*). The predicted semantic segmentation map is up-sampled to the original size for evaluation.

## 4.3. Comparison with State-of-the-art

**NYU-Dv2 Dataset.** The comparison results on the NYU-Dv2 dataset with 40-category are shown in Table 1. It can be observed that our approach leads to substantial improvements over the current state-of-the-arts. Note that most methods on the NYU-Dv2 are RGB-D methods, means the ground-truth depth map is taken as one of the input sources. Although our method only takes the RGB image as input, it performs better than the RGB-D based methods. RefineNet [24] and RDFNet [31] also utilize multi-scale information but combine features only at the backbone by a complex configuration without side supervision. These results in Table 1 also reveal that incorporating depth infor-

mation generally improves the performance.

In addition, to evaluate the performance of our model on the imbalanced distributed data, we also show the results on each category, as in Table 2. From the category-wise results shown in the table, we can see that our method performs better than other methods in most categories. Specially, on some “hard” categories (*e.g.*, *shelves*, *books*, *bag*), our method still achieves a relatively higher IoU. We owe the robustness among almost all the categories to the effectively learned depth embeddings with associated feature sharing/fusion with the 2D color information, and the newly introduced loss function. For the *board* (*whiteboard* in Figure 5) category which is also with few samples and even difficult to be distinguished from the *wall* or *picture* on depth map, our model surpasses others. This in part validates the effectiveness of our distilled geometry-aware information with joint fusion strategy. Notice our model performs not well on some categories like the *person*, *wall*, *floor*, which may due to our joint reasoning property of both 2D and 3D implicitly, as the depth may vary a lot in different scenes compared to the corresponding 2D appearance.

**SUN RGBD Dataset.** We also compare our method with state-of-the-arts on the large-scale SUN RGBD dataset. The results are shown in Table 3. As some methods in Table 1 not reported the performance on SUN RGBD dataset while other methods only reported on the SUN RGBD, the compared methods may vary from Table 1 to Table 3. The comparison on this large-scale dataset again validates the effectiveness of the proposed method, with better performance than the compared methods. Note there are many low-quality depth maps in SUN RGBD dataset caused by the capture device [36, 31], which may affect the auxiliary utility from the depth. From the results we can see, even without manually clipping the data our method can achieve state-of-the-art performance, which indicates that the learned depth-aware embedding is effective in representing 3D information.

**Generalization.** To evaluate the generalization capability of the proposed method, we first fine-tune our model on a recent proposed larger dataset (ScanNet [8]), where the mIoU achieves 56.9 on the *val* set. In addition, we further test on an outdoor scene (CityScapes [7]) for reference, resulting in a mIoU of 71.4 on *val*. While on par with state-of-the-arts, these additional evaluations demonstrate the generalization ability of our model to a certain extent.

## 4.4. Ablation Study

In order to discover the functionality of each component in the proposed network, in this section we conduct an ablation study on the NYU-Dv2 dataset. All the training and testing procedures of each ablative experiment are kept the

Table 2. Comparison with state-of-the-arts on each category of the NYU-Dv2 dataset. Percentage (%) of IoUs are shown for evaluation, with best performance marked in **bold**.

| Method                   | wall        | floor       | cabinet     | bed         | chair       | sofa        | table       | door        | window      | bookshelf   | picture     | counter     | blinds      | desk        | shelves     | curtain     | dresser     | pillow      | mirror      | floormat    |
|--------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| FCN [28]                 | 69.9        | 79.4        | 50.3        | 66.0        | 47.5        | 53.2        | 32.8        | 22.1        | 39.0        | 36.1        | 50.5        | 54.2        | 45.8        | 11.9        | 8.6         | 32.5        | 31.0        | 37.5        | 22.4        | 13.6        |
| Gupta <i>et al.</i> [13] | 68.0        | 81.3        | 44.9        | 65.0        | 47.9        | 47.9        | 29.9        | 20.3        | 32.6        | 18.1        | 40.3        | 51.3        | 42.0        | 11.3        | 3.5         | 29.1        | 34.8        | 34.4        | 16.4        | 28.0        |
| Deng <i>et al.</i> [10]  | 65.6        | 79.2        | 51.9        | 66.7        | 41.0        | 55.7        | 36.5        | 20.3        | 33.2        | 32.6        | 44.6        | 53.6        | 49.1        | 10.8        | 9.1         | 47.6        | 27.6        | 42.5        | 30.2        | 32.7        |
| Cheng <i>et al.</i> [6]  | 78.5        | <b>87.1</b> | 56.6        | 70.1        | <b>65.2</b> | 63.9        | 46.9        | 35.9        | 47.1        | 48.9        | 54.3        | 66.3        | 51.7        | 20.6        | 13.7        | 49.8        | 43.2        | 50.4        | 48.5        | 32.2        |
| RDFNet [31]              | <b>79.7</b> | 87.0        | 60.9        | 73.4        | 64.6        | 65.4        | 50.7        | 39.9        | 49.6        | 44.9        | 61.2        | <b>67.1</b> | 63.9        | 28.6        | 14.2        | 59.7        | 49.0        | 49.9        | 54.3        | 39.4        |
| Proposed                 | 71.4        | 75.2        | <b>71.3</b> | <b>77.1</b> | 53.3        | <b>69.5</b> | <b>51.4</b> | <b>63.7</b> | <b>68.2</b> | <b>57.3</b> | <b>61.4</b> | 53.1        | <b>77.1</b> | <b>55.2</b> | <b>52.5</b> | <b>70.4</b> | <b>64.2</b> | <b>51.6</b> | <b>68.3</b> | <b>61.3</b> |

| Method                   | clothes     | ceiling     | books       | fridge      | tv          | paper       | towel       | shower      | box         | board       | person      | nightstand  | toilet      | sink        | lamp        | bathub      | bag         | ot. struct. | ot. furn.   | ot. props.  |
|--------------------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| FCN [28]                 | 18.3        | 59.1        | 27.3        | 27.0        | 41.9        | 15.9        | 26.1        | 14.1        | 6.5         | 12.9        | 57.6        | 30.1        | 61.3        | 44.8        | 32.1        | 39.2        | 4.8         | 15.2        | 7.7         | 30.0        |
| Gupta <i>et al.</i> [13] | 4.7         | 60.5        | 6.4         | 14.5        | 31.0        | 14.3        | 16.3        | 4.2         | 2.1         | 14.2        | 0.2         | 27.2        | 55.1        | 37.5        | 34.8        | 38.2        | 0.2         | 7.1         | 6.1         | 23.1        |
| Deng <i>et al.</i> [10]  | 12.6        | 56.7        | 8.9         | 21.6        | 19.2        | 28.0        | 28.6        | 22.9        | 1.6         | 1.0         | 9.6         | 30.6        | 48.4        | 41.8        | 28.1        | 27.6        | 0           | 9.8         | 7.6         | 24.5        |
| Cheng <i>et al.</i> [6]  | 24.7        | 62.0        | 34.2        | 45.3        | 53.4        | 27.7        | 42.6        | 23.9        | 11.2        | 58.8        | 53.2        | 54.1        | <b>80.4</b> | 59.2        | 45.5        | 52.6        | 15.9        | 12.7        | 16.4        | 29.3        |
| RDFNet [31]              | 26.9        | <b>69.1</b> | 35.0        | 58.9        | 63.8        | 34.1        | 41.6        | 38.5        | 11.6        | 54.0        | <b>80.0</b> | 45.3        | 65.7        | <b>62.1</b> | <b>47.1</b> | 57.3        | 19.1        | 30.7        | 20.6        | <b>39.0</b> |
| Proposed                 | <b>53.1</b> | 58.1        | <b>42.9</b> | <b>62.2</b> | <b>71.7</b> | <b>40.0</b> | <b>58.2</b> | <b>79.2</b> | <b>44.1</b> | <b>72.6</b> | 55.9        | <b>55.0</b> | 72.5        | 50.8        | 33.6        | <b>72.3</b> | <b>46.3</b> | <b>50.6</b> | <b>54.1</b> | 37.8        |

Table 3. Comparison with state-of-the-arts on the SUN RGBD.

| Method                  | Input | PixAcc.     | mAcc.       | mIoU        |
|-------------------------|-------|-------------|-------------|-------------|
| SegNet [3]              | RGB   | 72.63       | 44.76       | 31.84       |
| Lin <i>et al.</i> [25]  | RGB   | 78.4        | 53.4        | 42.3        |
| Bayesian-SegNet [18]    | RGB   | 71.2        | 45.9        | 30.7        |
| RefineNet [24]          | RGB   | 80.6        | 58.5        | 45.9        |
| Cheng <i>et al.</i> [6] | RGBD  | -           | 58.0        | -           |
| 3DGNN [32]              | RGBD  | -           | 57.0        | 45.9        |
| D-CNN [38]              | RGBD  | -           | 53.5        | 42.0        |
| RDFNet [31]             | RGBD  | <u>81.5</u> | <u>60.1</u> | <u>47.7</u> |
| Proposed                | RGB   | <b>85.5</b> | <b>74.9</b> | <b>54.5</b> |

same. Taking the semantic-only without depth information as a baseline, the performance of each component is shown in Table 4. The results in the table show that, by utilizing the new loss function ( $L_s$ ) for the network training, semantic segmentation performance is improved by a large margin. This mainly due to its specially designed configuration for the hard categories with very few samples. Another observation is that, incorporating depth information enables a boost in the performance considerably, which reveals the effectiveness of reasoning 2D and 3D information together. Although the strategy of using ground-truth depth as input (encoded by HHA [13]) shows the effectiveness of the depth information, the proposed approach of learning depth-aware embeddings (DepEm) further boosts the performance. Feature propagation (FeatProp) from the depth branch to the semantic branch enables a thorough RGB-D fusion implicitly in the feature space. By introducing the geometry-aware propagation scheme, the performance is significantly improved. For the two fusion solutions (Figure 3), geometry-aware propagation (GAP) performs better than the vanilla convolution (VanConv). We owe this to the

Table 4. Ablation study of the proposed model on the NYU-Dv2.

| Model                              | PixAcc. | mAcc. | mIoU |
|------------------------------------|---------|-------|------|
| sem-only                           | 64.7    | 44.1  | 36.0 |
| sem+ $L_s$                         | 70.4    | 50.3  | 40.6 |
| sem+ $L_s$ +HHA                    | 72.0    | 53.2  | 42.4 |
| sem+ $L_s$ +DepEm                  | 75.6    | 57.5  | 44.7 |
| sem+ $L_s$ +DepEm+FeatProp         | 77.9    | 60.0  | 45.4 |
| sem+ $L_s$ +DepEm+FeatProp+VanConv | 81.6    | 64.9  | 55.1 |
| sem+ $L_s$ +DepEm+FeatProp+GAP     | 83.4    | 65.6  | 56.0 |
| sem+ $L_s$ +DepEm+FeatProp+GAP+SPF | 84.8    | 68.7  | 59.6 |

geometric affinity distilled from the depth embedding. The final SPF blocks for multi-level fusion with the encoder features give another rise in the performance.

#### 4.5. Analysis on Depth Supervision

Although our model does not need any depth information as input during test time, the depth supervision is still necessary for the network training. In this section, we analyze the possibility of semi-supervision from depth, *i.e.*, with only partial depth information during training. We perform the experiments on the NYU-Dv2 dataset. The original depth training data is rearranged into four different subsets that contain 20%, 40%, 60%, and 80% of the full training set. All these subsets are constructed by random selection from the original set. For the network training, as depth information may not exist for a training sample, in such cases, we freeze the learning of the depth branch and only perform inference. The other parts of the network are trained by the same strategy as the previous experiments. The results are shown in Table 5. Using 0% of depth data means only the semantic branch is reserved (*i.e.*, sem+ $L_s$ ), while for 20% – 100% all the other components are included (*i.e.*, the whole model). The results reveal that depth information is important to help the semantic segmentation when performing as

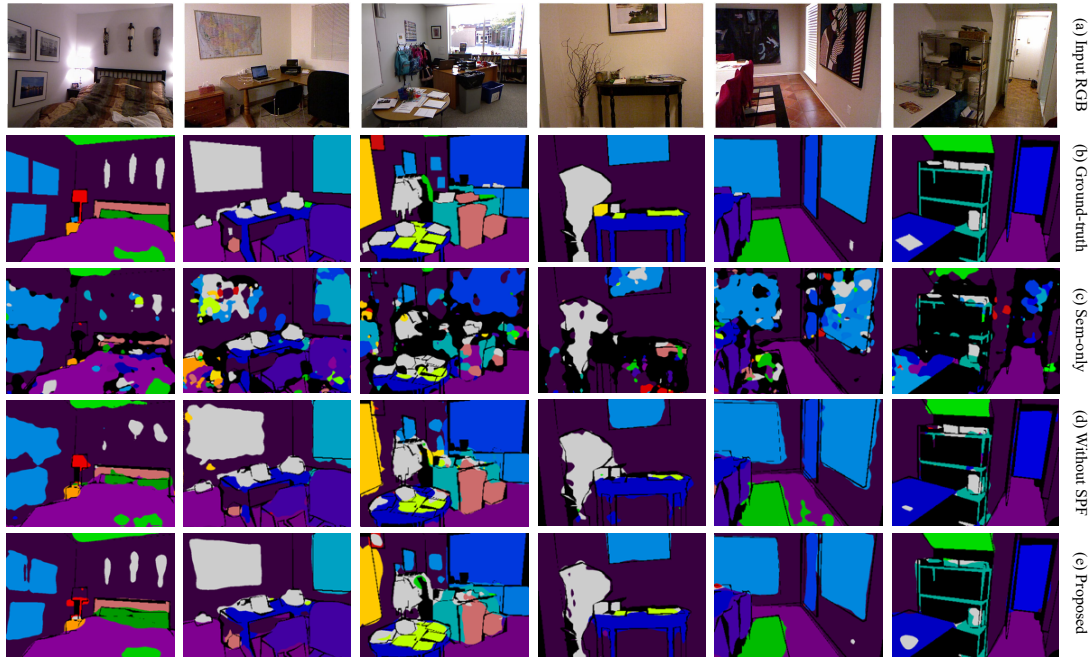


Figure 6. Qualitative performance on NYU-Dv2 dataset. (a) and (b) are the input and ground-truth, respectively. Our method (e) is compared to the result with only semantic branch (c), and the result without the SPFs (d).

Table 5. Performance evaluation for semi-supervision from depth.

| Data size | 0%   | 20%  | 40%  | 60%  | 80%  | 100% |
|-----------|------|------|------|------|------|------|
| PixAcc.   | 70.4 | 71.2 | 77.2 | 80.0 | 83.1 | 84.8 |
| mAcc.     | 50.3 | 51.0 | 58.3 | 68.2 | 70.3 | 68.7 |
| mIoU      | 40.6 | 41.1 | 45.6 | 51.6 | 55.2 | 59.6 |

a supervision. In addition, more supervision from the depth data leads to a better performance. Note that even with only 20% of the depth supervision, our model is able to result in a slightly better performance than the baseline without depth, which demonstrates the effectiveness of our model to learn the depth-aware embeddings for semantic segmentation.

#### 4.6. Qualitative Performance

We show some qualitative results of our method on the NYU-Dv2 dataset for semantic segmentation in Figure 6. For comparison, we also include the visual results of without the depth information (sem-only) and without the skip pyramid feature fusion (without SPF). From the result we can see that by learning the depth-aware embeddings, the geometry information is well distilled. For example, the *pillows* have very similar patterns to the *bed*, which cannot be easily discriminated by just the 2D appearance (c), while with the depth embeddings (d, e) they can be well separated to corresponding categories. Similar examples can be found in the *dustbin* of the third column and the *door* in the last column. Furthermore, when incorporating the SPF blocks

(e) for fusion with the backbone features that are closer to the image domain, more context information and object details are recovered. For example the *pictures* and *decorations* on the *wall* of the first column, the tiny socket in the second last example, and the *windows*, *paintings* across all the shown examples, *etc.*

## 5. Conclusions

In this paper, we presented a new framework that takes full advantage of the 3D geometry information by distilling depth-aware embedding implicitly for single RGB image semantic segmentation. The geometric distillation and semantic label prediction are jointly reasoned by decoupling a shared backbone network. The learned embeddings are used as a guidance to improve the semantic features by a geometry-aware propagation architecture. The distilled features are further fed back to the shared backbone to fuse with multi-level context information by skip pyramid fusion blocks. Our model captures both 2D appearance and 3D geometry information by only taking one single RGB image as input. Experiments on indoor RGB-D semantic segmentation benchmarks demonstrate that our model achieves favorable performance against state-of-the-art methods.

**Acknowledgments:** We acknowledge the support of EPSRC Programme Grant Seebibyte EP/M013774/1 and IARPA D17PC00341. YW is supported by IBM-ILLINOIS Center for Cognitive Computing Systems Research (C3SR).



## References

- [1] Iasonas Kokkinos, Adam W Harley, Konstantinos G. Derpanis. Segmentation-aware convolutional networks using local attention masks. In *ICCV*, 2017. 2
- [2] Anurag Arnab, Sadeep Jayasumana, Shuai Zheng, and Philip HS Torr. Higher order conditional random fields in deep neural networks. In *ECCV*, 2016. 2
- [3] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE TPAMI*, 39(12):2481–2495, 2017. 2, 7
- [4] Siddhartha Chandra and Iasonas Kokkinos. Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian crfs. In *ECCV*, 2016. 2
- [5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 2
- [6] Yanhua Cheng, Rui Cai, Zhiwei Li, Xin Zhao, and Kaiqi Huang. Locality-sensitive deconvolution networks with gated fusion for rgb-d indoor semantic segmentation. In *CVPR*, 2017. 1, 2, 6, 7
- [7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016. 6
- [8] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 6
- [9] Jifeng Dai, Kaiming He, and Jian Sun. Boxesup: Exploiting bounding boxes to supervise convolutional networks for semantic segmentation. In *ICCV*, 2015. 2
- [10] Zhuo Deng, Sinisa Todorovic, and Longin Jan Latecki. Semantic segmentation of rgb-d images with mutex constraints. In *ICCV*, 2015. 6, 7
- [11] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015. 1, 2, 3, 6
- [12] Saurabh Gupta, Pablo Arbelaez, and Jitendra Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *CVPR*, 2013. 1, 2, 5, 6
- [13] Saurabh Gupta, Ross Girshick, Pablo Arbeláez, and Jitendra Malik. Learning rich features from rgb-d images for object detection and segmentation. In *ECCV*, 2014. 1, 2, 3, 6, 7
- [14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015. 6
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015. 6
- [16] Judy Hoffman, Saurabh Gupta, and Trevor Darrell. Learning with side information through modality hallucination. In *CVPR*, 2016. 2
- [17] Allison Janoch, Sergey Karayev, Yangqing Jia, Jonathan T Barron, Mario Fritz, Kate Saenko, and Trevor Darrell. A category-level 3d object dataset: Putting the kinect to work. In *Consumer Depth Cameras for Computer Vision*, pages 141–165. 2013. 5
- [18] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. In *BMVC*, 2015. 7
- [19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6
- [20] Iasonas Kokkinos. Ubernet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, 2017. 3
- [21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012. 2
- [22] Iro Laina, Christian Rupprecht, Vasileios Belagiannis, Federico Tombari, and Nassir Navab. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV)*, 2016. 5
- [23] Zhen Li, Yukang Gan, Xiaodan Liang, Yizhou Yu, Hui Cheng, and Liang Lin. Lstm-cf: Unifying context modeling and fusion with lstms for rgb-d scene labeling. In *ECCV*, 2016. 2
- [24] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *CVPR*, 2017. 2, 6, 7
- [25] Guosheng Lin, Chunhua Shen, Anton Van Den Hengel, and Ian Reid. Efficient piecewise training of deep structured models for semantic segmentation. In *CVPR*, 2016. 2, 6, 7
- [26] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017. 5
- [27] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *ICCV*, 2017. 5
- [28] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 1, 2, 6, 7
- [29] Arsalan Mousavian, Hamed Pirsiavash, and Jana Košecká. Joint semantic segmentation and depth estimation with deep convolutional networks. In *3D Vision (3DV)*, 2016. 6
- [30] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *ICCV*, 2015. 2
- [31] Seong-Jin Park, Ki-Sang Hong, and Seungyong Lee. Rdfnet: Rgb-d multi-level residual feature fusion for indoor semantic segmentation. In *ICCV*, 2017. 2, 6, 7
- [32] Xiaojuan Qi, Renjie Liao, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. 3d graph neural networks for rgb-d semantic segmentation. In *ICCV*, 2017. 1, 2, 6, 7
- [33] Xiaofeng Ren, Liefeng Bo, and Dieter Fox. Rgb-(d) scene labeling: Features and algorithms. In *CVPR*, 2012. 1, 2
- [34] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgb-d images. In *ECCV*, 2012. 1, 2, 5

- [35] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [2](#)
- [36] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015. [5](#), [6](#)
- [37] Peng Wang, Xiaohui Shen, Zhe Lin, Scott Cohen, Brian Price, and Alan L Yuille. Towards unified depth and semantic prediction from a single image. In *CVPR*, 2015. [1](#), [2](#)
- [38] Weiyue Wang and Ulrich Neumann. Depth-aware cnn for rgb-d segmentation. In *ECCV*, 2018. [2](#), [6](#), [7](#)
- [39] Jianxiong Xiao, Andrew Owens, and Antonio Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *ICCV*, 2013. [5](#)
- [40] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *ICCV*, 2015. [2](#)