Zig-Zag Network for Semantic Segmentation of RGB-D Images

Di Lin, *Member, IEEE* and Hui Huang, *Senior Member, IEEE*

Abstract—Semantic segmentation of images requires an understanding of appearances of objects and their spatial relationships in scenes. The fully convolutional network (FCN) has been successfully applied to recognize objects' appearances, which are represented with RGB channels. Images augmented with depth channels provide more understanding of the geometric information of the scene in an image. In this paper, we present a multiple-branch neural network to utilize depth information to assist in the semantic segmentation of images. Our approach splits the image into layers according to the "scene-scale". We introduce the context-aware receptive field (CARF), which provides better control of the relevant context information of learned features. Each branch of the network is equipped with CARF to adaptively aggregate the context information of image regions, leading to a more focused domain that is easier to learn. Furthermore, we propose a new zig-zag architecture to exchange information between the feature maps at different levels, augmented by the CARFs of the backbone network and decoder network. With the flexible information propagation allowed by our zig-zag network, we enrich the context information of feature maps for the segmentation. We show that the zig-zag network achieves state-of-the-art performances on several public datasets.

Index Terms—RGB-D images, semantic segmentation, convolutional neural networks

1 Introduction

Semantic image segmentation is a fundamental problem in computer vision. It enables the pixel-wise categorization of objects [1], [2] and scenes [3], [4]. Recently, deep convolutional neural networks (CNNs) [5], [6], [7] pre-trained on large-scale image data have been adopted for semantic segmentation [8], [9], [10], [11], [12]. The emergence of powerful convolutional networks have significantly improved the performances of semantic segmentation.

As depth data captured by low-cost sensors becomes widespread, there is increasing research on leveraging it to assist in semantic segmentation. Compared to color information, depth data captures geometric information of images, which is used to learn useful image representations. To employ depth data for semantic segmentation, conventional methods [8], [13], [14] associate it as an additional channel to the RGB channels as input to networks. Recent works [15], [16] have modeled the relationship between depth and color modalities to improve segmentation. Although depth data clearly helps to separate objects and scenes, it has much less semantic information than colors [15]. This motivates the search for better means to exploit the depth to enhance semantic segmentation.

Instead of using depth data to extract semantic information for segmenting images, we proposed a cascaded feature network (CFN) [17] that uses depth data to split the image into layers representing similar scene-scale. We referred to a scene-scale as the scale of objects and scenes in general, as observed in the input images.[1] As shown in Fig. 1, there is correlation between depth and scene-scale; smaller scene-scales appear in regions with greater depth, and larger scene-scales appear in the near field. In smaller scene-scale regions, objects and scenes densely coexist, forming more complex correlation between objects and scenes relative to larger scene-scale regions. To represent the complex characteristics of smaller scene-scale regions, we introduced context-aware receptive fields, which are computed based on super-pixels determined by the underlying scene structures. We used small super-pixels to subdivide images, allowing the CFN to learn more focused local characteristics of image regions. Then we propagated the local information to small scene-scales, and employed larger super-pixels to aggregate the local information as the context feature map for complex characteristics. The CFN [17] provides better control of the information propagation between image regions at different scene-scales. It avoids over diverse information for large scene-scales, while providing distilled context information for smaller scene-scales. In this paper, we further improve the CFN and CARF from the two perspectives of resolution recovery and region information adjustment.

Resolution Recovery. The fully convolutional network (FCN) [18], [19] with multiple branches has been used to generate distinct features for distinct regions of interest, which are applicable to different scene-scales. Rather than using independent branches that only influence the regions

• *The authors are with the College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China. E-mail: {ande.lin1988, hhzhiyan}@gmail.com.*

Manuscript received 19 Sept. 2018; revised 17 Apr. 2019; accepted 10 June 2019. Date of publication 18 June 2019; date of current version 2 Sept. 2020. (Corresponding author: Hui Huang).
Recommended for acceptance by M. Bennamoun and Y. Guo.
Digital Object Identifier no. 10.1109/TPAMI.2019.2923513

1. We assume the images have similar resolution, which can be achieved in pre-processing.

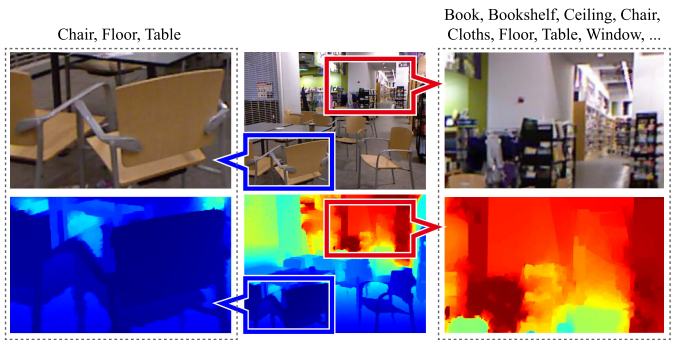

Fig. 1. Correlation between depth and scene-scale: the near field (highlighted in blue rectangle) consists of a large scene-scale, while the far field (highlighted in red rectangle) has a small scene-scale.

of the corresponding scene-scales, CFN [17] employs a cascaded architecture to enable communication between branches. However, this architecture is built on top of the highest-level convolutional feature map of the FCN, which has the lowest resolution and losses visual details. The prevalent methods [20], [21], [22], [23] have used the decoder architecture to compute high-resolution feature maps. The decoder progressively adds high-level feature maps to low-level feature maps, propagating the semantic information to the image regions with richer details. However, the conventional decoder architecture does not feed low-level feature maps back to high-level feature maps to learn better semantic information. This is essential to relatively small scene-scales that requires visual details of scenes/objects.

Region Information Adjustment. To compute the representation of object/scene relationships, numerous segmentation networks [10], [11], [20], [21], [24], [25] have enriched the context information of convolutional features using a set of regular receptive fields, which are context-oblivious in the sense that they do not consider their extent with respect to the underlying image structure. We previously proposed the CARF [17] to compute the context feature for super-pixels separately. However, it does not adjust the information propagation between different super-pixels. Existing methods [26], [27], [28] adaptively combine adjacent super-pixels based on their relationship, which is computationally efficient. Nevertheless, the information propagation [17], [26], [27], [28] is guided by the pairwise relationship between adjacent super-pixels, regardless of high-order relationships for richer context information. Most works [14], [17], [26], [27], [28], [29] have neglected the problematic super-pixels, which inevitably involve noisy regions in partitions of objects/scenes.

Our Approach. We address the above two problems in the context of RGB-D image segmentation. First, we present a zig-zag network to connect the backbone and decoder architectures, which, as shown in Fig. 2, produce convolutional feature maps at different levels. At adjacent levels, we input the low-level feature maps of the backbone architectures, along with the relatively higher-level augmented feature maps of the decoder architectures, to the zig-zag architecture. The zig-zag architecture has multiple branches equipped with CARFs, to compute context feature maps based on the input feature maps. As we show in Fig. 2, the zig-zag architecture enables communication between the backbone and decoder architectures. It allows the low- and high-level feature maps to exchange context information, constructing richer context information for all scene-scales.

Second, we propose a two-stage weighting scheme for the CARF to adjust the information propagation between super-pixels. In the first stage, the local weighting learns the weights for receptive fields within the same super-pixel. It adjusts the importance of each receptive field, selecting the useful information. We show that local weighting alleviates the negative effect of imperfect super-pixels for constructing context feature maps. In the second stage, the high-order weighting enables information propagation between super-pixels. To construct high-order context feature maps, we follow the previous methods [26], [27], [28] to weight adjacent super-pixels to save test time. Here, the network learns local and high-order weights with respect to the context of objects/scenes.

We show that our network enriches the context information and enhances the overall performance. The zig-zag network's performance is demonstrated on two public datasets for semantic segmentation on RGB-D images. Our method achieves the mean intersection-over-union (IoU) values of 51.2 on the NYUDv2 dataset [30] and 51.8 on the SUN-RGBD dataset [31]. We evaluate the performance of CARF on two datasets for the general segmentation task. Using state-of-the-art methods along with CARF, we achieve consistent improvement on the PASCAL VOC 2012 [1] and the Cityscapes test sets [4].

This manuscript extends its ICCV version [17] as summarized below:

- We use a new zig-zag architecture to connect backbone and decoder architectures to yield high-resolution feature maps, which contain rich context information for different scene-scales.
- We apply a two-stage weighting scheme to the CARF to provide the local and high-order context information.
- We conduct more comprehensive studies to evaluate our model.

In Section 2, we revisit related works on semantic segmentation of RGB-D images. In Sections 3, 4 and 5, we present our zig-zag architecture, two-stage weighting scheme for CARF and details of their implementation. In Section 6, we conduct ablation studies to evaluate our model, and compare our model with state-of-the-art methods. We provide our conclusions in Section 7.

2 Related Work

FCN for Semantic Segmentation. FCNs [8] have been broadly used in semantic segmentation systems [9], [10], [11], [20], [21], [25], [32]. FCNs have stacked down-sampling operations to compute feature maps containing high-level semantic information. However, down-sampling operations inevitably reduce the image resolution, resulting in segmentation information loss on image regions. Some works have addressed this problem. Yu et al. [33] and Chen et al. [9] applied the atrous convolution to maintain relatively high-resolution information, which requires substantial memory space. Noh et al. [34], Badrinarayanan et al. [35] and Ghiasi et al. [36] used deconvolution and unpooling to increase the

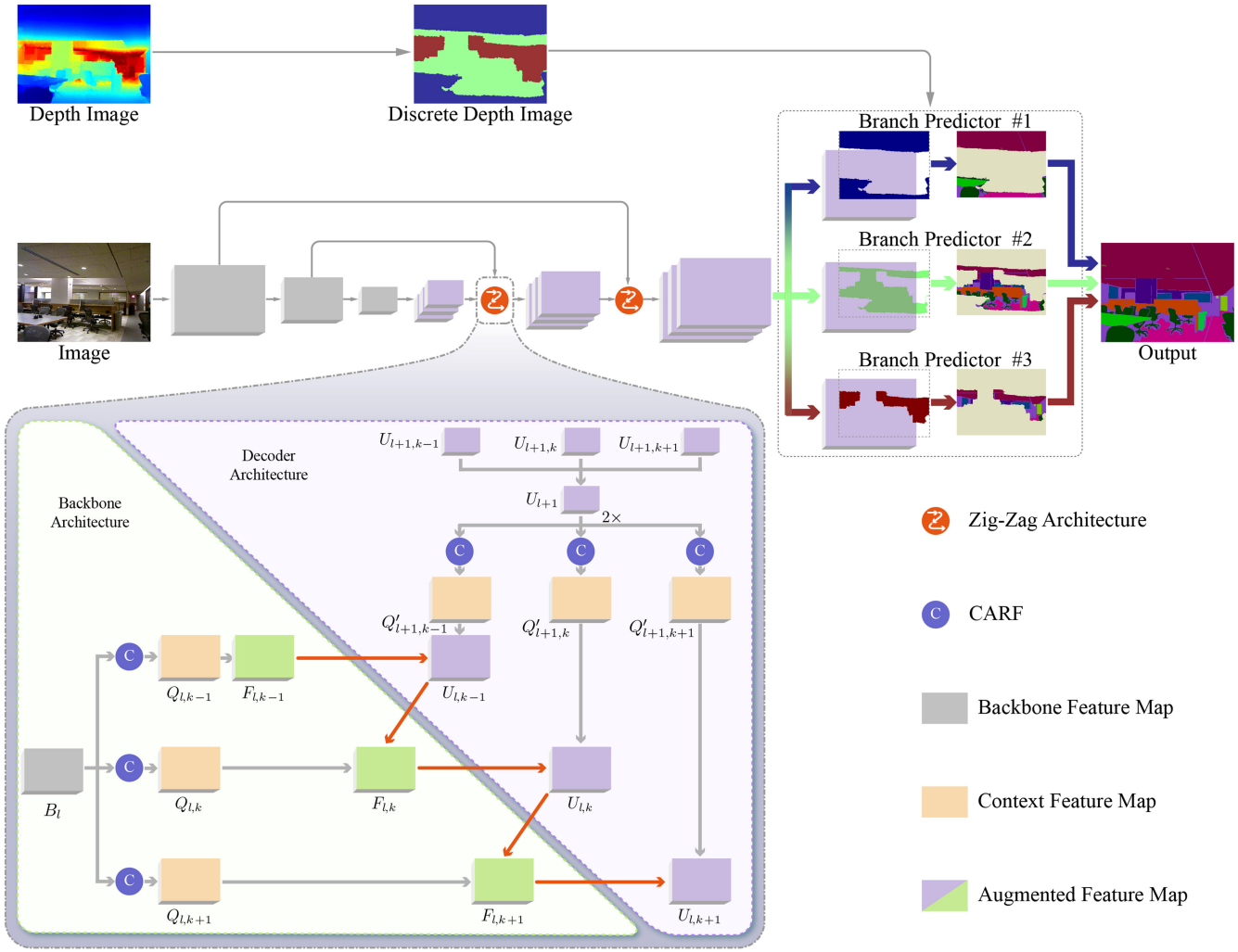

Fig. 2. Overview of our network. Given a color image, we use CNN to compute the convolutional feature maps. These are passed to the zig-zag architectures, which gradually recover their resolutions. Each zig-zag architecture has multiple branches. The discrete depth image is layered, where each layer represents a scene-scale and is used to match the image regions to corresponding network branches. Each branch has the context-aware receptive field (CARF), which produces context feature map to combine with the feature from an adjacent branch. The predictions of all branches are merged to achieve the eventual segmentation result. Please see Fig. 3 for details of the CARF.

resolution of convolutional feature maps that have fewer feature channels, e.g., the last convolutional feature maps of VGG-16 [6] and ResNet-101 [7]. However, these methods cannot reuse the high-resolution feature maps, which provide object details for segmentation. Recent works [20], [22], [37], [38] have used the encoder-decoder (ED) architecture to gradually propagate the semantic information of low-resolution feature maps to shallower network layers, producing high-resolution feature maps with richer information.

Context information of multiple receptive fields is used as well to alleviate problematic prediction. Several studies [9], [10], [11], [25] have integrated graphical models to capture the context of multiple pixels. From another perspective, Lin et al. [20], Zhao et al. [21] and Chen et al. [22] utilized convolutional/pooling kernels of diverse sizes to construct spatial pyramid (SP) architecture, which captures different receptive fields of images to effectively enrich the context information. Chen et al. [23] further used the ED architecture along with different atrous convolutions to produce high-resolution feature maps having rich context information. However, the traditional decoder architecture [20], [22], [37], [38] is incapable of capturing the context of image regions in different scene-scales. This problem is addressed in our paper.

Our method also makes use of the convolutional features extracted from receptive fields of different sizes. In contrast to [20], [21], [22], [23], which used regular kernels, we control the size of super-pixels to capture receptive fields, which are more aware of the relationships between image regions. Similarly, super-pixels were used in [14], [26], [28], [32] to group the convolutional features from a set of receptive fields. Different from our method, these studies do not use the relationship of a wider range of super-pixels to construct context feature maps.

Semantic Segmentation of RGB-D Images. Semantic segmentation of RGB-D images has been studied for more than a decade [13], [14], [15], [30], [39]. Different from traditional semantic segmentation of RGB images [1], [3], [4], an additional depth channel is available now, which provides a better understanding of the geometric information of the scene images. Many prior studies have harnessed useful information from the depth channel. Silberman et al. [30] proposed

an approach to parse the spatial characteristics, such as support relations, using the RGB image along with the depth cue. Gupta et al. [39] used the depth image to construct a geometric contour cue to benefit both object detection and segmentation of RGB-D images.

CNN/FCN has been used recently to learn features from depth to help in segmenting RGB-D images. Couprie et al. [40] proposed to learn CNN using the combination of RGB and depth image pairs such that the convolutional feature maintains depth information. Gupta et al. [13] and He et al. [14] encoded the depth image as an HHA image [39], which maintains each pixel's horizontal disparity, height above ground, and angle of the local surface normal. Networks trained on different modalities, e.g., RGB and HHA images, were fused by Long et al. [8] to boost segmentation accuracy. Compared to direct fusion of segmentation scores as in [8], the network proposed by Wang et al. [15] produces better segmentation results by harnessing deeper correlation of RGB and depth image pairs.

There are works [29], [41], [42] using depth data to model the 3D-spatial relationships of objects in a CNN/FCN. In our scenario, depth information plays a more significant role in guiding feature learning for the regions of different scene-scales. The depth image is layered to identify the scene-scale of the region. An effective design of a neural network structure is thus facilitated to consider the characteristic of the region in a specific scene-scale. This technique can be applied to benefit feature learning from different data modalities, as shown in the results.

3 Zig-Zag Network

To compute the high-resolution feature map for semantic segmentation of RGB-D images, we present a zig-zag network to incorporate the backbone architecture [17] and decoder architecture [23]. Fig. 2 provides an overview of the zig-zag network. Initially, we use the backbone FCN to extract convolutional feature maps at different levels (see the gray blocks in Fig. 2). At adjacent levels, we feed the backbone feature maps and the higher-level augmented feature maps (see the purple blocks in Fig. 2) to the zig-zag architecture. The zig-zag architecture has CARF to process the backbone feature maps and the augmented feature maps, and use a decoder architecture to yield higher-resolution augmented feature maps. The highest-resolution augmented feature maps are used for the segmentation task.

As shown in Fig. 2, the backbone architecture has multiple branches to process the backbone feature maps at the lth level. Each branch is equipped with a CARF. Along the horizontal direction, the CARFs take the backbone feature maps as input, producing context feature maps (orange blocks in Fig. 2) at different scene-scales. We use context feature maps to produce augmented feature maps (green blocks in Fig. 2), which are fed to the decoder architecture. Along the vertical direction, the decoder architecture has another set of branches equipped with CARFs. It computes higher-resolution context feature maps, based on the augmented feature map (smaller purple blocks in Fig. 2) at the $(l+1)$th level. The decoder architecture combines the higher-resolution context feature maps with the augmented feature maps of the backbone architecture, yielding the higher-resolution augmented feature map (larger purple blocks in Fig. 2) at the lth level. At the kth scene-scale, we pass the augmented feature maps of the decoder architecture to the $(k+1)$th branch of the backbone architecture, enriching the context information for the $(k+1)$th scene-scale. Note that here the backbone and decoder architectures exchange augmented feature maps in a zig-zag manner, strengthening the context information for all scene-scales.

More formally, given a color image $I \in \mathbb{R}^{H\times W\times 3}$ as input of the backbone FCN, we compute the backbone feature maps $\{B_l | l = 1, ..., L\}$. For the feature map $B_l \in \mathbb{R}^{H\times W\times C}$, we use a K-branch structure to construct context feature maps $\{Q_{l,k} | k = 1, ..., K\}$, where $Q_{l,k} \in \mathbb{R}^{H\times W\times C}$. Note that the 1st branch is for the largest scene-scale. Given a depth image $D \in \mathbb{R}^{H\times W}$, we project each pixel to one of the K branches. Each branch deals with a set of pixels that have depth values within a certain range. As illustrated in Fig. 2, the kth branch outputs the feature map $F_{l,k} \in \mathbb{R}^{H\times W\times C}$ as:

$$F_{l,k} = U_{l,k-1} + Q_{l,k}, \quad k = 1, ..., K, \tag{1}$$

where $U_{l,k-1} \in \mathbb{R}^{H\times W\times C}$ denotes the augmented feature map at the lth level. We set $U_{l,0} = 0$. The augmented feature map $F_{l,k}$ is in a combination form, which is modeled by summing the augmented feature map $U_{l,k-1}$ and the context feature map $Q_{l,k}$. Note that $U_{l,k-1}$ contains the high-level semantic context information for enhancing $F_{l,k}$. At the lth level, we compute augmented feature maps $\{F_{l,k} | k = 1, ..., K\}$ for K scene-scales.

As shown in Fig. 2, the decoder architecture also has CARFs to compute the context feature maps $\{Q'_{l+1,k} | k = 1, ..., K\}$. We sum the augmented feature map $F_{l,k}$ and the higher-level context feature map $Q'_{l+1,k} \in \mathbb{R}^{H\times W\times C}$, yielding a new feature map $U_{l,k} \in \mathbb{R}^{H\times W\times C}$ as:

$$U_{l,k} = F_{l,k} + Q'_{l+1,k}, \quad l = 1, ..., L. \tag{2}$$

Here, $U_{l,k}$ is influenced by $F_{l,k}$ having lower-level information. We compute the context feature maps $Q'_{l+1,k}$ based on the high-level feature maps $U_{l+1} \in \mathbb{R}^{H\times W\times C}$ denoted as:

$$U_{l+1} = \sum_{k=1}^{K} U_{l+1,k}. \tag{3}$$

We set $U_{L+1} = 0$ and therefore $U_{L,k} = F_{L,k}$. We apply deconvolutional kernels to enlarge the resolution of U_{l+1} before computing $Q'_{l+1,k}$. Note that the feature map U_{l+1} aggregates information of all scene-scales. With CARFs, the high-level semantic information can be propagated to all of the network branches for different scene-scales.

Finally, the feature map $U_{1,k}$ is fed to the predictor for segmentation. Given all the pixels assigned to the kth scene-scale, we denote their class labels as a set y_k, which is determined as:

$$y_k = f(U_{1,k}). \tag{4}$$

The function $f(:)$ is the softmax predictor widely used for pixel-wise categorization. We denote the class label of the pixel at location (x, y) as $y_k(x, y)$. Combining the prediction results of all of the branches forms the final segmentation y on the image I.

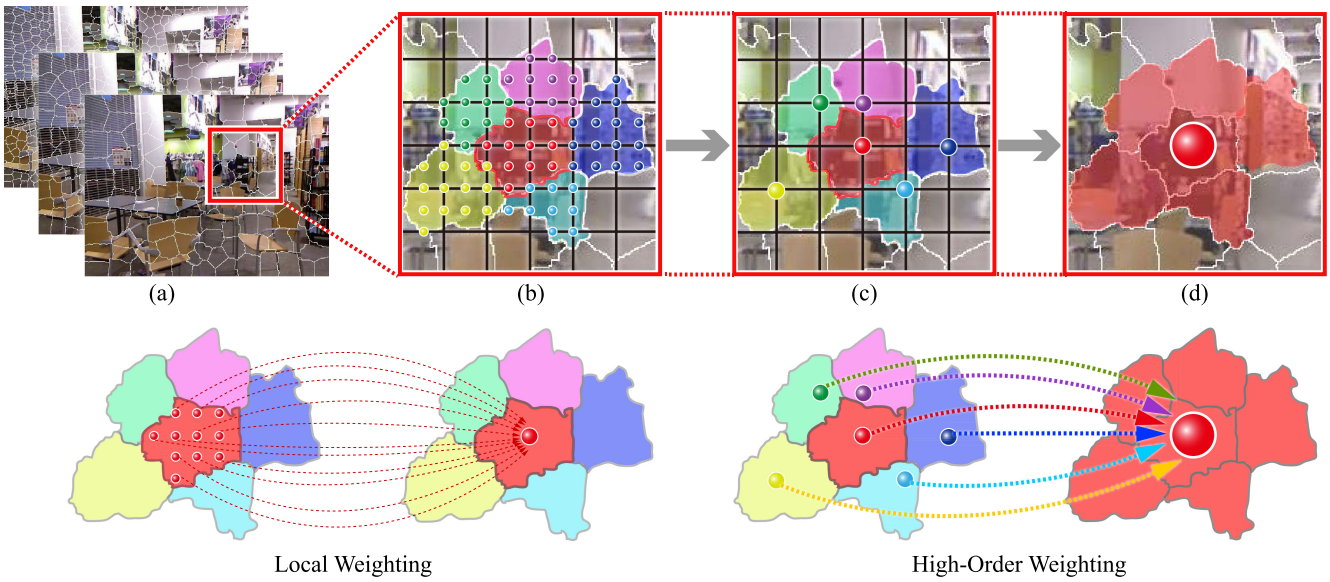

Fig. 3. Two-stage weighting scheme of CARF: (a) image partitioned into super-pixels with different sizes; (b) each neuron of the convolutional feature map is augmented by local weighting, which uses the information of neurons residing in the same super-pixel; (c) after local weighting, the neurons residing in each super-pixel are augmented; (d) each neuron is further augmented by high-order weighting, which uses the content of adjacent super-pixels, to form the context feature map. The two-stage weighting is repeatedly applied to the images partitioned by super-pixels of diverse sizes. Note that the feature map has smaller resolution than the image due to down-sampling of the network.

Next, we elaborate on the CARF for computing context feature maps. For clarity, we omit the notations l and k from this point onward.

4 Context-Aware Receptive Field

The receptive fields of common networks are pre-defined. Here, we present a CARF where the receptive field is spatially-variant and its extent is defined according to the local and high-order context. The idea is to aggregate convolutional features into richer features that better learn the relevant content.

The context information generated by the CARF is controlled by adjusting the sizes of the super-pixels. For regions of low scene-scale, we choose larger super-pixels that include more object and scene information, while at a higher scene-scale, we choose finer super-pixels to avoid too much diverse information; see also Fig. 3a. The adaptive size of the super-pixels helps capture the complex object/scene relationships in different regions. The relevant context comprises the neighborhoods of a super-pixel as shown in Fig. 3d; i.e., a neuron $Q(x, y, c)$ in the feature map Q is an aggregation of all the receptive fields within the super-pixel, which contains (x, y) and its adjacent super-pixels.

Our CARF encodes the local and high-order context information provided by super-pixels into augmented feature maps. We use a two-stage weighting scheme to compute CARFs, as discussed below.

Local Weighting. Given an image I, we utilize the toolkit [43] to generate a set of non-overlapping super-pixels denoted as $\{S_i\}$, satisfying $\bigcup_i S_i = I$ and $S_i \bigcap S_j = \varnothing, \forall i, j$. As shown in Figs. 3b and 3c, at the first stage, we augment on the neurons residing in the same super-pixel. This local augmentation produces a feature map $M \in \mathbb{R}^{H\times W\times C}$, where the neuron $M(x, y, c) \in \mathbb{R}$ is formulated as:

$$M(x, y, c) = w^l(x, y, c) \cdot B(x, y, c), \tag{5}$$

where B is the backbone feature map. The local weight map $w^l \in \mathbb{R}^{H\times W\times C}$ is computed as:

$$w^l(x, y) = \sigma(W * [B(x, y), B_i]), \tag{6}$$

where $(x, y) \in \Phi(S_i)$. The spatial coordinate (x, y) uniquely corresponds to a center of the regular receptive field in the image space. Thus, $\Phi(S_i)$ defines a set of centers of regular receptive fields that are located within the super-pixel S_i. W represents a set of 1×1 convolutional kernels. σ is the sigmoid activation function. $[\cdot, \cdot]$ represents the concatenation operation. $B_i \in \mathbb{R}^C$ aggregates the neurons residing in the same super-pixel S_i. It is formulated as:

$$B_i(c) = \sum_{(x,y)\in\Phi(S_i)} B(x, y, c). \tag{7}$$

In Eq. (7), the feature B_i represents the overall property of S_i. As formulated in Eq. (6), the neurons residing in S_i are combined with B_i. With this, each neuron perceives the information of other neurons in S_i. The combined feature is used to learn the weight map w^l that accounts for the relationship between neurons in the same super-pixel. In Eq. (5), w^l adjusts the neurons of the feature map B, selecting useful information for the high-order weighting process.

High-Order Weighting. At the second stage (see Figs. 3c and 3d), we aggregate the features of M that are associated with adjacent super-pixels to model a new feature map $Q \in \mathbb{R}^{C\times H\times W}$:

$$Q(x, y, c) = w_i^h(c) \cdot M(x, y, c) + \sum_{S_j\in\mathcal{N}(S_i)} w_j^h(c) \cdot \sum_{(x',y')\in\Phi(S_j)} \frac{M(x', y', c)}{|\Phi(S_j)|}, \tag{8}$$

where $(x, y) \in \Phi(S_i)$. Here $S_j \in \mathcal{N}(S_i)$ means that the super-pixel S_i and S_j are adjacent. In Eq. (8), $w_i^h \in \mathbb{R}^C$ is the weight for S_i. We compute w_i^h as:

$$w_i^h(c) = \lambda_i \cdot \sum_{(x,y)\in\Phi(S_i)} \frac{w^h(x, y, c)}{|\Phi(S_i)|}. \quad (9)$$

We apply successive convolutional operations on the feature map M to compute the high-order weight map $w^h \in \mathbb{R}^{H\times W\times C}$. In this way, we learn the high-order weight map from a wide range of image regions, rather than the pair of adjacent super-pixels. $|\Phi(S_i)|$ denotes the numbers of regular receptive field centers located within the super-pixel S_i.

In Eq. (8), w_i^h controls the information of $M(x, y, c)$ residing at S_i, and is used to construct $Q(x, y, c)$. $Q(x, y, c)$ has access to the information of the adjacent super-pixels that are adapted by the high-order weight map. It forms the context feature map used below where each neuron $Q(x, y, c)$ represents a CARF.

5 Implementation Details

Preparation of Image Data. The original RGB images are used as a data source. In addition, we encode each single-channel depth image as a three-channel HHA image [13], [39], which maintains the geometric information of the pixels. The sets of RGB and HHA images are used to train segmentation networks. When preparing the images for network training, we use the four common strategies of flipping, cropping, scaling and rotating to augment the training data.

CARF Settings. The number of scene-scales is pre-defined before using CARFs. We obtain the global range of depth value from all of the depth maps provided by the datasets. For example, the depth value of the NYUDv2 dataset varies from 0 to 102.7 meters. The global range is then divided by the number of branches. Each pixel in the image is assigned to the corresponding scene-scale with respect to its depth value. The super-pixels are controllable in our CARF components. For lower scene-scale, the CARF uses larger super-pixels to capture richer context information. Following this principle, we use larger sizes to broaden the super-pixels. On average, it takes about 3 seconds to compute super-pixels for each image.

Zig-Zag Network Construction. We use four TITAN XP display cards, each with 12 GB memory, as the main devices for all experiments. We modify the Caffe platform [44] to construct our network, which is based on FCN [8]. The network structure, which has been pre-trained on ImageNet [45], i.e., ResNet-101 [7], serve as the backbone architecture on which we build our zig-zag network. Specifically, we use the ResNet-101 network layers *res2*, *res3*, *res4* and *res5* as $\{B_1, B_2, B_3, B_4\}$, which are applied with the zig-zag network to produce the high-resolution feature map. The ResNet-101 network is used for internal study of our zig-zag network. For comparisons with state-of-the-art methods, we use the deeper ResNet-152 [7] to improve segmentation. Given the pre-computed super-pixels, it takes about 35 ms/image to train the network. Given the trained network, we need about 3.023 seconds to test an image. Note that the testing time is contributed by the computation of super-pixels (about 3 seconds/image) and forward propagation of the network (about 23 ms/image).

We optimize the segmentation network using the standard SGD solver. The network is fine-tuned with a learning rate of 1e-10 for 80K mini-batches. After that, we decay the learning rate to 1e-11 for the next 50K mini-batches. The size of each mini-batch is set to 8 by default. As suggested in [8], we use a heavy momentum of 0.99 to achieve stable optimization on relatively small-scale data.

6 Results and Evaluation

To show the efficacy of the zig-zag network and evaluate its performance, we test it on two public datasets: NYUDv2 [30] and SUN-RGBD [31]. The NYUDv2 dataset is more widely used for analysis. We therefore conduct most of our evaluation on it, while using the SUN-RGBD dataset to extend the comparison to state-of-the-art methods. Our CARF is applicable to an array of networks for general segmentation tasks, and thus we further study the effect on segmentation accuracy by using the CARF along with different networks. We evaluate the results on the PASCAL VOC 2012 [1] and Cityscapes test sets [4].

The NYUDv2 dataset [30] contains 1,449 RGB-D scene images. Among them, 795 images were split for training and 654 images for testing. In [13], a validation set of 414 images, was selected from the original training set. We follow the segmentation annotations provided in [39], where all of the pixels are labeled by 40 classes.

Following the common way of evaluating semantic segmentation schemes [20], [21], we perform multi-scale testing. Four scales $\{0.6, 0.8, 1, 1.1\}$ are used to resize the testing image before feeding it to the network. The output scores of the four re-scaled images are then averaged for the final prediction. We report on the semantic segmentation performance in terms of pixel accuracy, mean accuracy and mean intersection-over-union (IoU).

Sensitivities to Partitions of Depth and Color Images. We examine the effect on segmentation accuracy by controlling the number of network branches. We experiment with different numbers $\{1, 2, 3, 4, 5, 6\}$, where each number is used to partition depth images into different levels. The input to the zig-zag network includes the RGB image for segmentation and the partitioned depth image for splitting image regions for different branches. We empirically set the sizes of super-pixels as 1600, 3000, 4200, 6000, 10000 and 12000 for the six applicable branches. For each number of branches, we report the segmentation accuracy on the NYUDv2 validation set in Fig. 4a.

We note that the single-branch zig-zag network achieved a lower score than the scores of other networks having two or more branches. As only one CARF is used in the single-branch network, specific context feature maps can not be achieved for different scene-scales. We find that three-branch zig-zag network achieved the best result. We also observe that further increasing the number of branches, e.g., using four-, five- or six-branch networks, causes a performance drop. In these cases, larger super-pixels are used. This suggests that too large super-pixels are not suitable to use, as they may overly diversify the object/scene classes and lose focus on the stable patterns that should be learned by the zig-zag network.

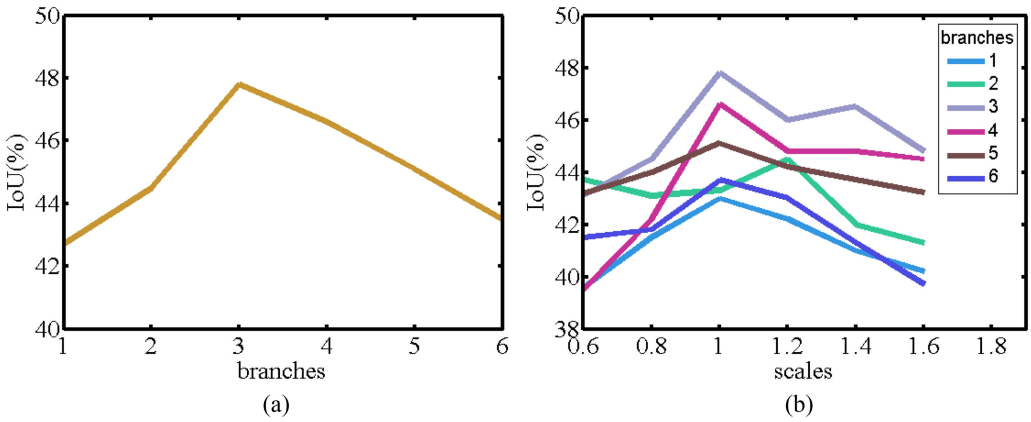

Fig. 4. Sensitivities to the number of branches (a) and the scale of super-pixels (b). Performances are evaluated on the NYUDv2 validation set. Segmentation accuracy is reported in terms of IoU (%).

We investigate the sensitivity to different partitions of color images. This is done by controlling the sizes of super-pixels. Again, we experiment with zig-zag networks with different numbers of branches. We apply the standard sizes of super-pixels {1600, 3000, 4200, 6000, 10000, 12000} to the applicable network branches. For each branch, we use different scales {0.6, 0.8, 1.0, 1.2, 1.4, 1.6} to resize the super-pixels. With various sizes of super-pixels, we report the segmentation scores of different networks (see Fig. 4b).

By increasing the scale, we enlarge the super-pixels for each network branch. As shown in Fig. 4b, a larger scale generally improves the performances of all networks. This is because the high-order weighting of the CARF can use the larger super-pixels to enrich the context information. We also find that a too large scale degrades the performance. We note that a larger super-pixel includes more receptive fields. However, too many receptive fields form complex relationships, which are difficult to learn by the local weighting of the CARF for producing useful information.

According to the validation results shown above, we use a three-branch zig-zag network and the standard sizes of super-pixels (1600, 3000, 4200) by default in the following experiments.

Ablation Study of Two-Stage Weighting. The CARF defines the adaptive extent of the receptive field and plays a critical role in adjusting the context information for different scene-scales. We use the local and high-order weighting to compute the CARF. Below, various key components of the weighting scheme are removed to examine the effect on the segmentation performance. The results are shown in Table 1.

TABLE 1
Ablation Experiments of Using Local and High-Order Weighting Schemes for Computing CARFs

local	high-order	pixel acc.	mean acc.	IoU
		69.7	53.3	43.8
✓		71.1	54.5	44.6
✓	✓	**73.4**	**57.5**	**47.8**

Performances are evaluated on the NYUDv2 validation set. Segmentation accuracy is reported in terms of pixel accuracy, mean accuracy and IoU (%).

In the first case, we remove the local and high-order weights. This means that the CARF degraded to the version proposed in [17], which achieves a segmentation score of 43.8 IoU. By adding local weights, we enable the selection of information for each super-pixel, increasing the segmentation score to 44.6 IoU. Furthermore, we use the high-order weighting scheme to construct the context feature map. The full weighting scheme achieves a segmentation score of 47.8 IoU, which outperforms the CARF without two-stage weighting by a margin of 4 points.

Similar to the use of various sizes of super-pixels in the CARF, stacking multiple two-stage weighting layers also change the extent of the receptive field. In Fig. 5, we compare the segmentation accuracy of using different numbers of two-stage weighting layers. Here, we use the three-branch network along with the standard sizes of super-pixels (1600, 3000, 4200). Again, we use different scales {0.6, 0.8, 1.0, 1.2, 1.4, 1.6} to resize the super-pixels for each branch. Given small scales (0.6 and 0.8) of super-pixels, using two or three layers of two-stage weighting slightly improves the segmentation accuracy. However, we note that multiple layers produce more feature maps at the cost of extra computation time and storage space. When using relatively larger scales (1.0, 1.2, 1.4 and 1.6), we find that multiple layers lead to negligible improvement, and even performance degradation. This is because large super-pixels and multiple layers significantly enlarge the receptive fields, which contain complex information.

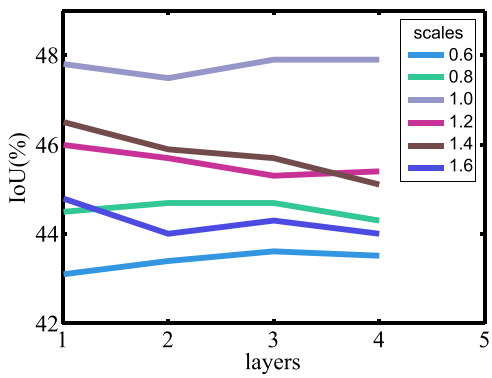

Fig. 5. Sensitivity to the number of two-stage weighting layers. Performances are evaluated on the NYUDv2 validation set. Segmentation accuracy is reported in terms of IoU (%).

TABLE 2
Strategies of Using the CARF, Evaluated on the NYUDv2 Validation Set

strategy	method	pixel acc.	mean acc.	IoU
w/o super-pixel	Chen et al. [9]	66.0	49.0	38.6
	Zheng et al. [11]	67.1	50.2	40.1
	Zhao et al. [21]	69.2	52.8	43.5
w/ super-pixel	He et al. [14]	67.4	50.6	40.3
	Lin et al. [17]	69.7	53.3	43.8
	Liang et al. [28]	72.8	55.7	45.2
	ours	**73.4**	**57.5**	**47.8**

Segmentation accuracy is reported in terms of pixel accuracy, mean accuracy and IoU (%).

Comparisons of Context Representations. We compare different context representations in Table 2. The CARF uses super-pixels to aggregate receptive fields for the context feature map. However, other researchers [9], [11] have constructed the context feature map without super-pixels. Without the CARF, we follow Chen et al. [9] in the use of a conditional random field (CRF) to process the segmentation score map output by the network, leading to 38.6 IoU. Zheng et al. [11] used RNN to model CRF, which enriches the context information of convolutional feature maps. By replacing the CARF with RNN to construct the context feature map, we achieve a segmentation score of 40.1 IoU. Zhao et al. [21] use different pooling kernels to compute pyramid context feature maps at different scales. We experiment with the pyramid pooling method [21] in place of the CARF, where small/large kernels were used for large/small scene-scales. Although the pyramid pooling method accounts for multi-scale context information, it achieves a lower segmentation score of 43.5 IoU than our method. This shows that super-pixels are important to enrich the context feature map.

Several studies [14], [17], [28] have proved that super-pixels provide rich context information. To construct the context representation, He et al. [14] used super-pixels independently. Instead, Lin et al. [17] summed the adjacent super-pixels to encode their relationship into the context feature map, achieving a better result (43.8 IoU) than He et al. [14] (40.3 IoU). Liang et al. [28] used long short-term memory (LSTM) to model the relationship between adjacent super-pixels. We adapt the LSTM [28] in place of the CARF. Although LSTM yields a better result (45.2 IoU) than Lin et al. [17], it requires much more memory for hidden states. We extend the context representation in [17] with the two-stage weighting scheme. Our method outperforms all of the compared methods. The performance gap suggests that our methods provides more useful context information for the segmentation task.

TABLE 3
Improvement with CARF

	PASCAL VOC 2012		Cityscapes	
	val set	test set	val set	test set
RefineNet [20]	82.7 → **84.2**	82.4 → **83.7**	71.5 → **72.3**	73.6 → **74.6**
PSPNet [21]	81.4 → **83.3**	85.4 → **86.5**	80.6 → **81.1**	81.2 → **81.7**
DPCNet [46]	84.2 → **86.0**	87.9 → **88.5**	80.9 → **81.5**	82.7 → **83.1**

Performances is evaluated on the PASCAL VOC 2012 [1] and the Cityscapes datsets [4]. Segmentation accuracy is reported in terms of IoU (%).

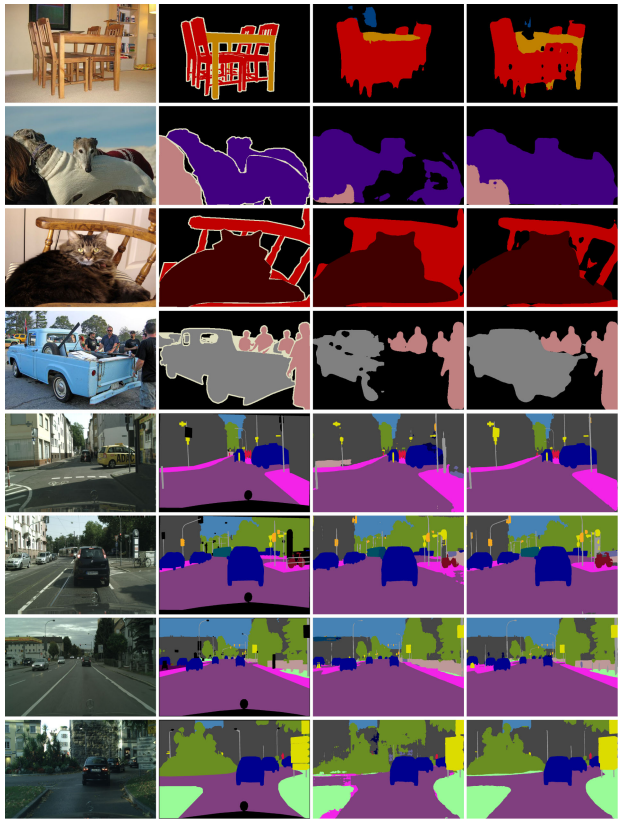

Fig. 6. Sample of the comparison to state-of-the-art DPCNet [46] and ours. Scene images are taken from the PASCAL VOC 2012 [1] (the first four rows) and Cityscapes [4] (the last four rows) validation sets.

Note that the CARF produces context feature maps, which can be used by different networks for the general segmentation task. We equip three widely-used segmentation networks, i.e., PSPNet [21], RefineNet [20] and DPCNet [46], with CARFs. Without the depth image for separate network branches, we sum the context feature maps output by different branches of CARFs for segmentation. We evaluate these networks on the PASCAL VOC 2012 [1] and the Cityscapes datasets [4] (see Table 3). Compared to different baseline models, the CARF generally yields improvement on the segmentation accuracies. Especially, the CARF improves the performance by 0.6–1.3 points on the PASCAL VOC 2012 test set, and by 0.4–1.0 points on the Cityscapes test set. This demonstrates that CARF is applicable to different networks for achieving the performance gain on semantic segmentation.

TABLE 4
Different Strategies of Propagating Context Information

scene-scale		super-pixel		performance		
large	small	large	small	pixel acc.	mean acc.	IoU
←		←		68.1	51.6	41.0
←		→		68.4	52.0	41.5
→		→		69.2	53.0	43.6
→		←		**73.4**	**57.5**	**47.8**

The arrows indicate the order of using scene-scales and super-pixels. Performances are evaluated on the NYUDv2 validation set. Segmentation accuracy is reported in terms of pixel accuracy, mean accuracy and IoU (%).

TABLE 5
Different Multi-Branch Networks, Evaluated on the NYUDv2 Validation Set

strategy	method	pixel acc.	mean acc.	IoU
w/o ZZNet	w/o decoder	67.8	51.2	40.6
	separate branches	68.9	52.7	42.7
	combined branches	70.5	54.0	44.1
	cascaded branches	72.0	55.8	45.3
w/ ZZNet	ours	**73.4**	**57.5**	**47.8**

Segmentation accuracy is reported in terms of pixel accuracy, mean accuracy and IoU (%).

In Fig. 6, we provide the qualitative comparison on the PASCAL VOC 2012 and Cityscapes validation sets.

We also experiment with using CRF to post-process the segmentation results of baseline models. Compared to the CARF that provides high-order context information, CRF focus on the local context of adjacent pixels. Thus, CRF achieves less improvement (0.02 – 0.13 points) on the PASCAL VOC 2012 and Cityscapes test sets.

Strategies of Propagating Context Information. Given a scene-scale, our zig-zag network gradually accumulates the context feature maps produced by the branches at larger scene-scales. Note that we use small super-pixels at large scene-scales, and apply larger super-pixels at smaller scene-scales. We achieve 47.8 IoU on the NYUDv2 validation set (see Table 4). We further compare our zig-zag network to different strategies of propagating context information.

In the first strategy, we reverse the order by propagating context information from small scene-scales to larger scene-scales. Here, we use small/large super-pixels at small/large scene-sales. Compared to our zig-zag network, we find that the performance significantly degraded to 41.0 IoU. Without the focused local information learned from large scene-scales, the context feature maps at smaller scene-scales contain diverse information, leading to the performance degradation. A similar performance drop (41.5 IoU) takes place in the second case, where we further reverse the order of super-pixels in the first case. Again, the learning of context feature maps of small scene-scales is not conditioned on the focused local information.

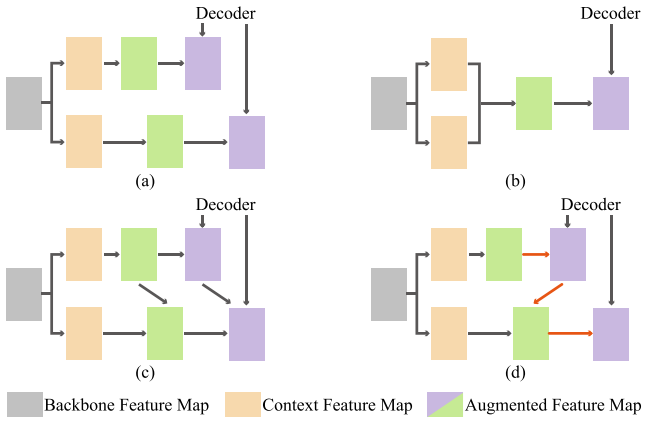

Fig. 7. The network can have (a) separate branches, (b) combined branches, (c) cascaded branches or (d) zig-zag branches. In each subfigure, we illustrate the multiple branches of the backbone architecture and omit the decoder with similar structure. For clarity, we illustrate it with two branches only. Each network can be extended to have more branches.

We further experiment with the third case, where we follow the zig-zag network to propagate information from a large scene-scale to a small scene-scale. However, we use the super-pixels in the opposite order, i.e., small/large super-pixels for small/large scene-scales, as large super-pixels include too diverse information at the beginning, and has a negative impact on all context feature maps. It can be clearly seen that the segmentation performance in the third case lags far behind our zig-zag network.

Comparisons of Multi-Branch Networks. Our zig-zag network connects the backbone and decoder architectures. It exploits multiple branches to handle different scene-scales. In Table 5, we evaluate the performance on segmentation and experiment with different configurations of network branches.

Following the CFN [17], we disable the decoder that produced the high-resolution feature map. It leads to a performance drop of 7.2 points (see "w/o decoder" in Table 5), compared to our full model. Next, we employ separate branches (Fig. 7a) for different scene-scales. The backbone and decoder architectures yield the augmented feature maps, combining them for each scene-scale in an isolated way. Although the CARF provides context information for each scene-scale, the information propagation between branches is lacking.

TABLE 6
Comparisons with Other State-of-the-Art Methods on the NYUDv2 Test Set

model	RGB-input	pixel acc.	mean acc.	IoU	RGB-D-input	pixel acc.	mean acc.	IoU
VGG-16	Long et al. [8]	60.0	42.2	29.2	Eigen et al. [47]	65.6	45.1	34.1
	Kendall et al. [48]	68.0	45.8	32.4	He et al. [14]	70.1	53.8	40.1
	Lin et al. [25]	70.0	53.6	40.6	Lin et al. [17]	70.6	54.2	41.7
ResNet-101	Zhao et al. [21]	72.8	55.9	45.2	Lin et al. [20]	73.3	58.2	46.3
	Lin et al. [20]	73.1	57.3	46.0	Lin et al. [17]	73.8	59.1	46.6
					Lee et al. [49]	75.6	62.2	49.1
					ours	**75.8**	**62.3**	**49.3**
ResNet-152	Lin et al. [20]	73.6	58.9	46.5	Lin et al. [20]	74.6	59.7	47.0
					Lin et al. [17]	74.8	60.4	47.7
					Lee et al. [49]	76.0	62.8	50.1
					ours	**77.0**	**64.0**	**51.2**

Segmentation accuracy is reported in terms of pixel accuracy, mean accuracy and IoU (%).

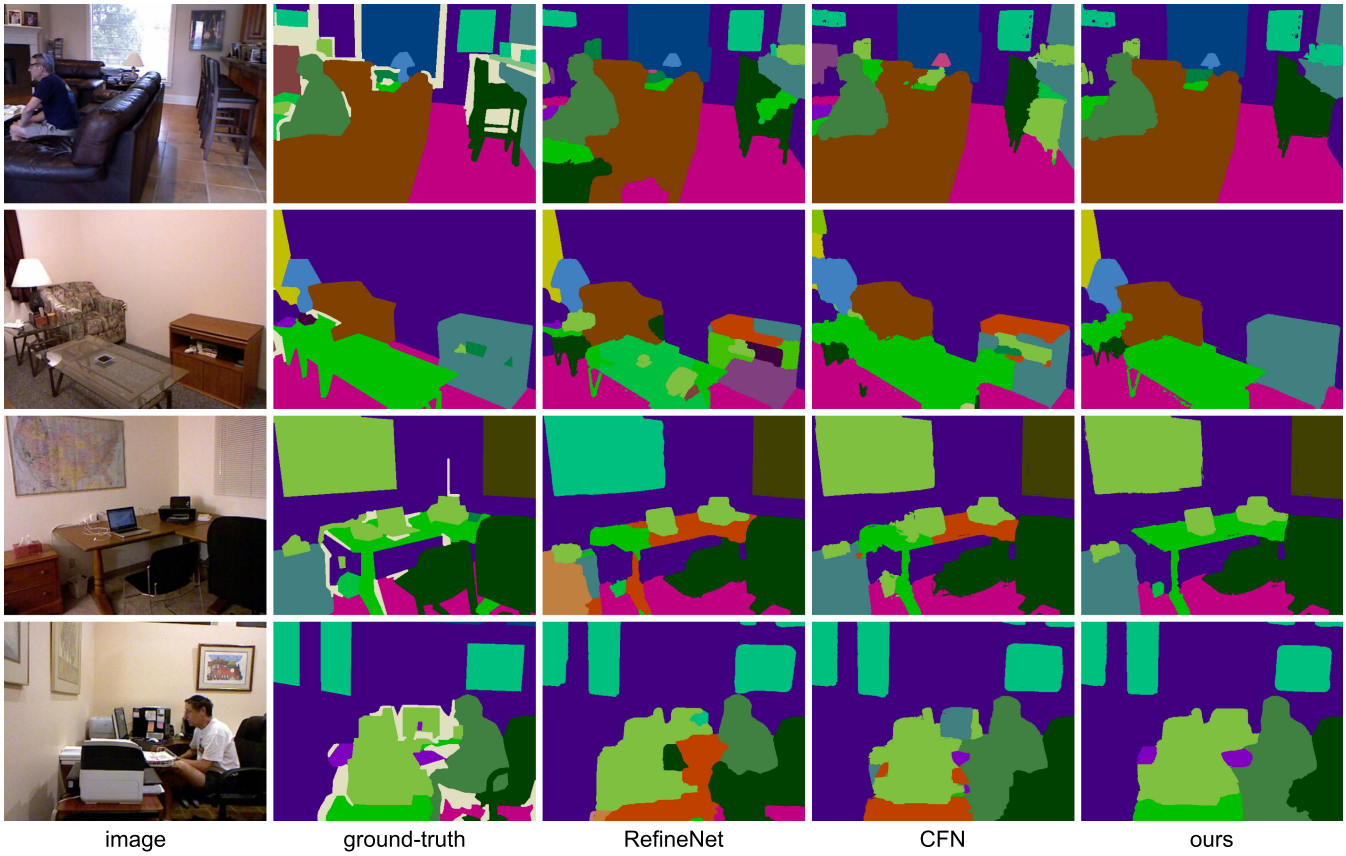

Fig. 8. Sample of the comparison to state-of-the-art models [17], [20] and ours. Scene images are taken from the NYUDv2 dataset [30].

Thus the separate branches produce a lower score (see "separate branches" in Table 5) than the zig-zag network.

The branches can be combined to segment images, as illustrated in Fig. 7b. With the combined branches, all of the scene-scenes share the same context information. The low scene-scales benefit from more global context information provided by broader super-pixels. However, mixing overly complex context information distracts the segmentation on relatively larger scene-scales. See "combined branches" in Table 5.

We further compare the cascaded network [17] with our zig-zag network, as illustrated in Fig. 7c and 7d. The cascaded branches propagate information between adjacent scene-scales. But it does not exchange information to assist in the joint learning of feature maps at different levels, as compared to our full model. See "cascaded branches" in Table 5.

Comparisons with State-of-the-art Methods. In Table 6, we compare our zig-zag network with state-of-the-art methods that are also based on deep neural networks. According to

TABLE 7
Class-Wise Semantic Segmentation Accuracy on the NYUDv2 Test Set

	wall	floor	cabinet	bed	chair	sofa	table	door	window	bkshelf
Lin et al. [17]	77.2	83.0	58.1	70.6	61.3	62.7	**51.2**	36.5	45.2	**46.0**
Lee et al. [49]	79.7	87.0	60.9	**73.4**	**64.6**	65.4	50.7	**39.9**	49.6	44.9
ours	**80.5**	**87.6**	**63.0**	72.3	63.9	**68.7**	51.1	37.6	**52.1**	44.7
	picture	counter	blind	desk	shelf	curtain	dresser	pillow	mirror	mat
Lin et al. [17]	57.3	64.8	**64.7**	23.3	10.9	54.1	50.0	44.2	51.4	38.2
Lee et al. [49]	**61.2**	67.1	63.9	28.6	14.2	59.7	49.0	**49.9**	54.3	39.4
ours	60.0	**69.2**	63.1	**30.5**	**15.6**	**60.3**	**49.3**	47.3	**58.7**	**42.6**
	cloths	ceiling	books	refridg	tv	paper	towel	shower	box	board
Lin et al. [17]	24.1	65.3	31.8	56.4	60.0	31.6	41.8	34.0	**13.1**	50.8
Lee et al. [49]	26.9	69.1	35.0	**58.9**	63.8	**34.1**	41.6	38.5	11.6	**54.0**
ours	**30.4**	**70.0**	**37.8**	56.2	**67.1**	32.5	**44.2**	**39.1**	12.5	52.6
	person	stand	toilet	sink	lamp	bathtub	bag	othstr	othfurn	othprop
Lin et al. [17]	77.5	42.8	61.5	**65.7**	41.9	53.5	**22.6**	26.5	16.4	37.0
Lee et al. [49]	80.0	45.3	65.7	62.1	**47.1**	57.3	19.1	30.7	**20.6**	39.0
ours	**82.6**	**47.1**	**68.2**	63.8	45.2	**61.4**	21.5	**34.7**	18.3	**44.8**

Segmentation accuracy is reported in terms of IoU (%).

TABLE 8
Comparisons with Other State-of-the-Art Methods on the SUN-RGBD Test Set

model	RGB-input	pixel acc.	mean acc.	IoU	RGB-D-input	pixel acc.	mean acc.	IoU
VGG-16	Chen et al. [9]	69.7	43.6	27.4	Long et al. [8]	74.3	47.3	35.1
	Kendall et al. [48]	71.2	45.9	30.7	Hazirbas et al. [50]	76.6	48.5	37.8
ResNet-101	Zhao et al. [21]	78.6	55.3	44.6	Lin et al. [20]	80.7	58.9	46.5
	Lin et al. [20]	80.4	57.8	45.7	Lin et al. [17]	80.9	59.6	47.0
				ours	**82.7**	**61.3**	**48.6**	
ResNet-152	Lin et al. [20]	80.6	58.5	45.9	Lin et al. [20]	81.1	59.8	47.3
					Lee et al. [49]	81.5	60.1	47.7
					Lin et al. [17]	82.4	60.7	48.1
					ours	**84.7**	**62.9**	**51.8**

Segmentation accuracy is reported in terms of pixel accuracy, mean accuracy and IoU (%).

the training and testing data, the compared methods are divided into two groups. All the methods are evaluated on the NYUDv2 test set.

In the first group, the methods use only RGB images for segmentation. The *RGB-input* column of Table 6 shows the performances of these methods. We find that the deep network proposed by Lin et al. [20] achieves the best accuracy in this group. This network is based on ResNet-152 [7], which is much deeper than the previous methods [8], [25], [48] using VGG-16 [6] and ResNet-152 [7]. It suggests that using a deeper network can improve segmentation accuracy.

In the second group, the methods take both RGB and depth images as input. The performances are shown in the *RGB-D-input* column of Table 6. We note that each depth image can be encoded as a three-channel HHA image, which maintains richer geometric information as shown in [13], [39]. Following Long et al. [8], we used HHA images in place of RGB images to train the segmentation network. Given an image, a segmentation network trained on HHA images was used to compute a score map, which is fused with the score map derived from the network trained on RGB images. The fusion strategy is implemented by averaging the score maps. Compared to the network [20], [21] that uses RGB images only, the network using both RGB and HHA images improves the segmentation accuracy. As the comparison between network structures are based on the same backbones (e.g., ResNet-101 and ResNet-152), we conclude that the performance gap is solely attributed to using HHA images for assisting segmentation.

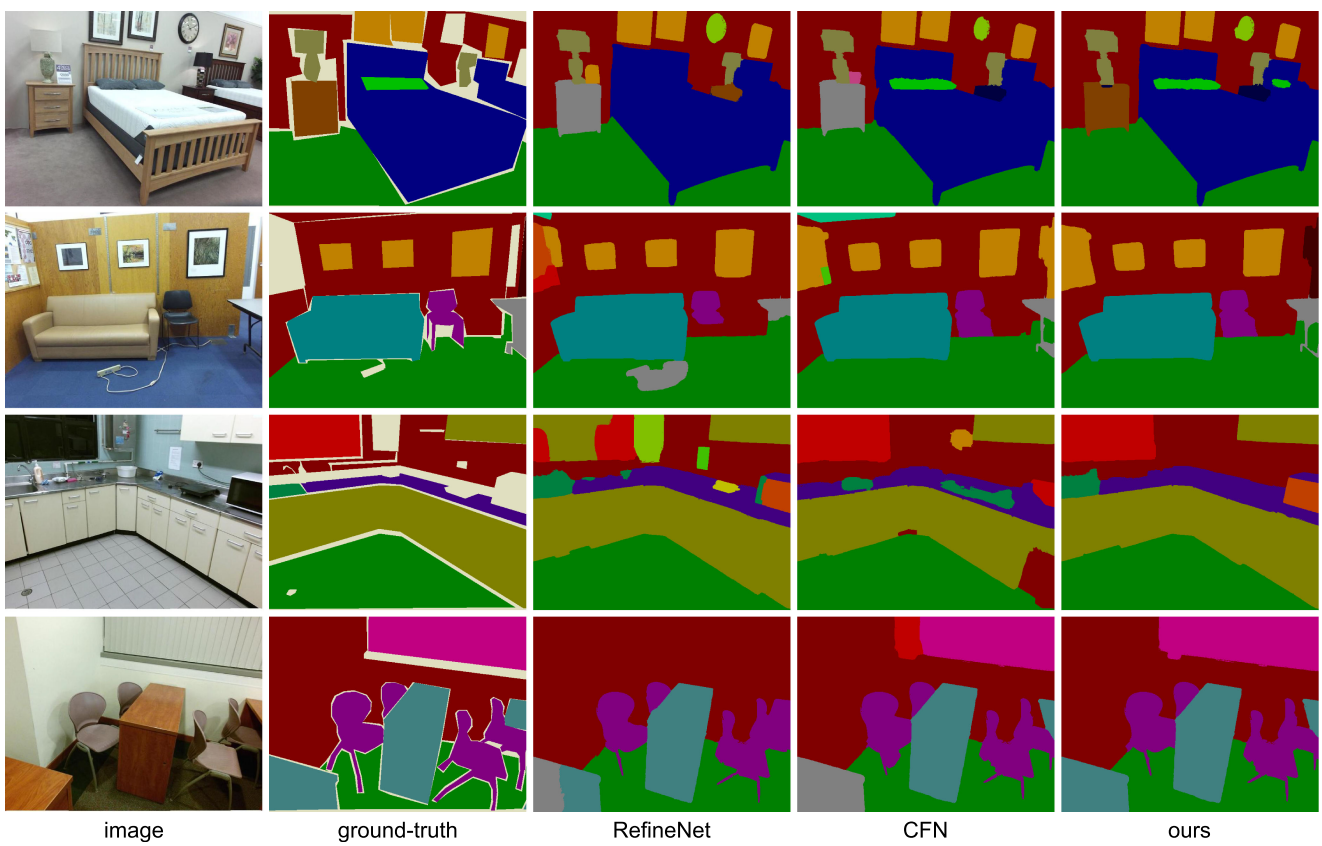

Fig. 9. Sample of the comparison to state-of-the-art models [17], [20] and ours. Scene images are taken from the SUN-RGBD dataset [31].

TABLE 9
Class-Wise Semantic Segmentation Accuracy on the SUN-RGBD Test Set

	wall	floor	cabinet	bed	chair	sofa	table	door	window	bkshelf
Lin et al. [20]	85.2	**94.6**	58.9	71.4	76.7	60.5	**67.9**	37.1	68.5	42.7
Lin et al. [17]	83.1	93.5	62.5	72.6	77.9	61.2	65.8	36.3	**71.6**	43.9
ours	**86.3**	91.7	**65.9**	**76.1**	**80.2**	**67.9**	67.6	**40.7**	71.5	**49.8**
	picture	counter	blind	desk	shelf	curtain	dresser	pillow	mirror	mat
Lin et al. [20]	**59.5**	37.1	36.1	16.0	14.3	70.3	38.1	**48.2**	34.1	13.7
Lin et al. [17]	56.7	38.6	37.2	**20.0**	11.8	72.4	39.6	45.0	31.2	16.5
ours	59.1	**40.2**	**39.4**	19.8	**25.5**	**74.7**	**42.3**	48.1	**35.9**	**18.2**
	cloths	ceiling	books	refridg	tv	paper	towel	shower	box	board
Lin et al. [20]	35.2	68.8	**51.5**	33.7	56.1	26.1	29.2	10.9	**31.0**	58.6
Lin et al. [17]	37.8	71.3	46.1	36.1	58.4	31.8	**33.4**	12.6	27.1	52.7
ours	**43.9**	**75.2**	49.8	**40.6**	**60.7**	**33.5**	31.0	**25.1**	30.6	**60.2**
	person	stand	toilet	sink	lamp	bathtub	bag			
Lin et al. [20]	47.5	14.0	70.6	67.1	35.6	51.2	33.0			
Lin et al. [17]	50.2	19.8	72.1	66.8	40.3	50.5	35.4			
ours	**55.0**	**22.4**	**78.8**	**69.3**	**43.8**	**52.4**	**43.1**			

Segmentation accuracy is reported in terms of IoU (%).

Our zig-zag network belongs to the second group. We use RGB and HHA images for training and testing. The zig-zag network based on ResNet-101 achieves an IoU of 49.3. We further use a deeper ResNet-152 [7] backbone network, and achieve a 51.2 IoU. This result is better than state-of-the-art methods. The previous best result was achieved by RDFNet [49]. Based on the same ResNet-152 backbone, RDFNet [49] requires to learn about 218 million parameters. In comparison, our method contains about 206 million learnable parameters. This means that RDFNet has a more complex model architecture than our zig-zag network. In Fig. 8, we show the visual improvement against the state-of-the-art models [17], [20]. This comparison demonstrates that our zig-zag network is compatible with different network structures and improves segmentation accuracy. We provide the accuracies of individual classes in Table 7. Compared to the state-of-the-art methods [17], [49], our zig-zag network achieves better results for most of the classes.

Experiments on SUN-RGBD Dataset. We conduct more experiments on the SUN-RGBD dataset [31], which comprises 10,335 images labeled with 37 classes. We use 5,285 images for training and the rest for evaluation. The SUN-RGBD dataset provides more images than the NYUDv2 dataset [30]. It thus can verify whether our method could effectively handle more diverse scene and depth conditions.

We show the segmentation accuracy of our method in Table 8. Again, the compared methods are divided into two groups. Similar to the previous experiments, we compare our method to the group of methods that consider both RGB and HHA images as input. With a ResNet-152 model trained on RGB and HHA images, the previous best performance was produced by the method of Lin et al. [17]. Using the same model and data, our method yields a better IoU of 51.8, which outperforms the previous best result by a margin of 3.7. The visualization results of our method on the SUN-RGBD dataset [31] can be found in Fig. 9. The accuracies of individual classes are provided in Table 9. Our zig-zag network outperforms other methods in most of the classes.

7 Conclusions

Recent developments in semantic segmentation of images have leveraged the power of convolutional networks that are trained on large datasets. In our work, we use depth information to provide more understanding of the geometric relationship between scenes/objects. It helps to produce features with richer context information for the appropriate scene-scale. We have also presented a zig-zag network to construct context feature maps at different levels. The zig-zag network exchanges useful information between feature maps. It enables flexible modeling of the data with a good balance between image regions in different scene-scales. Our method outperforms recent state-of-the-art methods.

Acknowledgments

We thank the anonymous reviewers and editors for their constructive suggestions. This work was supported in parts by NSFC (61702338), National 973 Program (2015CB352501), Guangdong Science and Technology Program (2015A030312015), Shenzhen Innovation Program (KQJSCX20170727101233642), LHTD (20170003), and the National Engineering Laboratory for Big Data System Computing Technology.

References

[1] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, pp. 303–338, 2010.

[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[3] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 891–898.

[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3212–3223.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.* - Vol. 1, 2012, pp. 1097–1105.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *ICLR*, 2015. [Online]. Available: http://arxiv.org/abs/1409.1556

[7] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[8] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.

[9] 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," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, 2018. [Online]. Available: https://doi.org/10.1109/TPAMI.2017.2699184

[10] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1377–1385.

[11] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. H. Torr, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1529–1537.

[12] D. Lin, J. Dai, J. Jia, K. He, and J. Sun, "Scribblesup: Scribble-supervised convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3159–3167.

[13] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 345–360.

[14] Y. He, W.-C. Chiu, M. Keuper, and M. Fritz, "RGBD semantic segmentation using spatio-temporal data-driven pooling," *CoRR*, vol. abs/1604.02388, 2016. [Online]. Available: http://arxiv.org/abs/1604.02388

[15] J. Wang, Z. Wang, D. Tao, S. See, and G. Wang, "Learning common and specific features for rgb-d semantic segmentation with deconvolutional networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 664–679.

[16] F. Husain, H. Schulz, B. Dellen, C. Torras, and S. Behnke, "Combining semantic and geometric features for object class segmentation of indoor scenes," *IEEE Robot. Autom. Lett.*, vol. 2, no. 1, pp. 49–55, Jan. 2017.

[17] D. Lin, G. Chen, D. Cohen-Or, P.-A. Heng, and H. Huang, "Cascaded feature network for semantic segmentation of RGB-D images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1320–1328.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proc. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[19] J. Dai, Y. Li, K. He, J. Sun, et al., "R-fcn: Object detection via region-based fully convolutional networks," in *Proc. Conf. Neural Inf. Process. Syst.*, 2016, pp. 379–387.

[20] G. Lin, A. Milan, C. Shen, and I. D. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5168–5177. [Online]. Available: https:doi.org/10.1109/CVPR.2017.549

[21] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 6230–6239. doi: 10.1109/CVPR.2017.660.

[22] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *CoRR*, vol. abs/1706.05587, 2017. [Online]. Available: http://arxiv.org/abs/1706.05587

[23] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Comput. Vis. Conf.*, 2018, pp. 833–851. doi: 10.1107/978-3-030-01234-2_49.

[24] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Hypercolumns for object segmentation and fine-grained localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 447–456.

[25] G. Lin, C. Shen, A. van den Hengel, and I. Reid, "Efficient piecewise training of deep structured models for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3194–3203.

[26] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich, "Feedforward semantic segmentation with zoom-out features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3376–3385.

[27] R. Gadde, V. Jampani, M. Kiefel, D. Kappler, and P. V. Gehler, "Superpixel convolutional networks using bilateral inceptions," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 597–613.

[28] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan, "Semantic object parsing with graph lstm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2175–2184.

[29] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun, "3d graph neural networks for rgbd semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017.

[30] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from rgbd images," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 746–760.

[31] S. Song, S. P. Lichtenberg, and J. Xiao, "Sun rgb-d: A rgb-d scene understanding benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 567–576.

[32] D. Lin, Y. Ji, D. Lischinski, D. Cohen-Or, and H. Huang, "Multi-scale context intertwining for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 622–638.

[33] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *ICLR*, 2016. [Online]. Available: http://arxiv.org/abs/1511.07122

[34] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1520–1528.

[35] V. Badrinarayanan, A. Handa, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," *CoRR*, vol. abs/1505.07293, 2015. [Online]. Available: http://arxiv.org/abs/1505.07293

[36] G. Ghiasi and C. C. Fowlkes, "Laplacian pyramid reconstruction and refinement for semantic segmentation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 519–534.

[37] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large kernel matters–improve semantic segmentation by global convolutional network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1743–1751. [Online]. Available: http://doi.org/10.1109/CVPR.2017.189

[38] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 936–944.

[39] S. Gupta, P. Arbelaez, and J. Malik, "Perceptual organization and recognition of indoor scenes from rgb-d images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 564–571.

[40] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, "Indoor semantic segmentation using depth information," *ICLR*, 2013. [Online]. Available: http://arxiv.org/abs/1301.3572

[41] W. Wang and U. Neumann, "Depth-aware CNN for RGB-D segmentation," *ECCV*, pp. 144–161, 2018. doi: 10.1007/978-3-030-01252-6_9.

[42] M. Tatarchenko, J. Park, V. Koltun, and Q.-Y. Zhou, "Tangent convolutions for dense prediction in 3d," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3887–3896.

[43] P. Dollár and C. L. Zitnick, "Structured forests for fast edge detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 1841–1848.

[44] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

[45] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[46] L.-C. Chen, M. Collins, Y. Zhu, G. Papandreou, B. Zoph, F. Schroff, H. Adam, and J. Shlens, "Searching for efficient multi-scale architectures for dense image prediction," in *Proc. Conf. Neural Inf. Process. Syst.*, 2018, pp. 8713–8724.

[47] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2650–2658.

[48] A. Kendall, V. Badrinarayanan, and R. Cipolla, "Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding," in *Proc. Brit. Mach. Vis. Conf.*, 2017. [Online]. Available: https://www.dropbox.com/s/gozsaobbk98azy/0205.pdf?dl=1

[49] S. Lee, S.-J. Park, and K.-S. Hong, "Rdfnet: Rgb-d multi-level residual feature fusion for indoor semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4990–4999.

[50] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers, "Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture," in *Proc. Asian Conf. Comput. Vis.*, 2016, pp. 213–228.

Di Lin received the bachelor's degree in software engineering from Sun Yat-sen University, in 2012, and the PhD degree from the Chinese University of Hong Kong, in 2016. He is an assistant professor with the College of Computer Science and Software Engineering, Shenzhen University. His research interests include computer vision and machine learning. He is a member of the IEEE.

Hui Huang received the PhD degree in applied math from The University of British Columbia, in 2008 and another PhD degree in computational math from Wuhan University, in 2006. She is a distinguished professor of Shenzhen University, where she directs the Visual Computing Research Center in College of Computer Science and Software Engineering. Her research interests include computer graphics and computer vision. She is currently a senior member of the IEEE and ACM, a distinguished member of CCF, an associate editor-in-chief of The Visual Computer and is on the editorial board of Computers & Graphics.