Cascaded Feature Network for Semantic Segmentation of RGB-D Images

Di Lin[1] Guangyong Chen[2] Daniel Cohen-Or[1,3] Pheng-Ann Heng[2,4] Hui Huang[1,4*]
[1]Shenzhen University [2]The Chinese University of Hong Kong [3]Tel Aviv University [4]SIAT

Abstract

Fully convolutional network (FCN) has been successfully applied in semantic segmentation of scenes represented with RGB images. Images augmented with depth channel provide more understanding of the geometric information of the scene in the image. The question is how to best exploit this additional information to improve the segmentation performance.

In this paper, we present a neural network with multiple branches for segmenting RGB-D images. Our approach is to use the available depth to split the image into layers with common visual characteristic of objects/scenes, or common "scene-resolution". We introduce context-aware receptive field (CaRF) which provides a better control on the relevant contextual information of the learned features. Equipped with CaRF, each branch of the network semantically segments relevant similar scene-resolution, leading to a more focused domain which is easier to learn. Furthermore, our network is cascaded with features from one branch augmenting the features of adjacent branch. We show that such cascading of features enriches the contextual information of each branch and enhances the overall performance. The accuracy that our network achieves outperforms the state-of-the-art methods on two public datasets.

1. Introduction

Semantic image segmentation is a fundamental problem in computer vision. It enables the pixel-wise categorization of objects [9, 26] and scenes [30, 2]. Recently, deep convolutional neural networks [21, 34, 15] pre-trained on large-scale image data are adopted for semantic segmentation [28, 1, 27, 38, 23]. The emergence of powerful convolutional networks have significantly improved the performances of semantic segmentation.

There has also been an increasing interest in leveraging depth information to assist semantic segmentation. Depth data becomes widespread, as it can be easily captured by commercially cheap sensors. Undoubtedly, depth information is able to improve segmentation, as it captures geometric information that is not captured by the color channels, and can directly enrich the image representation learned by deep networks. In [12, 28, 16], the depth data is added as a fourth channel in addition to the RGB channels as input to the networks. This straightforward approach increased the segmentation performance. More recent works [36, 17] have developed networks that jointly learn from the depth and color modalities, to further improve the segmentation. Although depth data clearly helps to separate between objects/scenes, it has much less semantic information than colors do. Moreover, there is little correlation between depth and color channels [36], which motivates better means to exploit the depth to enhance semantic segmentation.

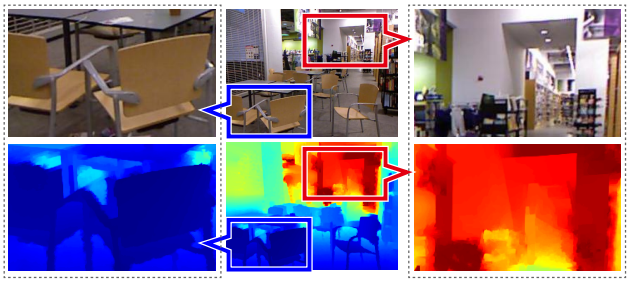


Figure 1: There is correlation between depth and scene-resolution: the near field (highlighted in blue rectangle) consists of high scene-resolution, while the far field (highlighted in red rectangle) has low scene resolution.

In this paper, we present a different approach to exploit depth information. The key idea is to use the depth to split the image into layers representing similar visual characteristic, or the "scene-resolution". We refer to scene-resolution as the resolution of the objects and scenes in general, as observed in the input images[1]. As shown in Figure 1, there is correlation between depth and scene-resolution; lower scene-resolution appears in regions that have higher depth, and higher scene-resolution appears in the near field. In lower scene-resolution regions, objects and scenes densely co-exist, forming more complex correlation between objects/scenes relative to higher scene-resolution

*Hui Huang is the corresponding author of this paper.

[1]We assume the images are with similar resolution, which can be achieved in pre-processing.

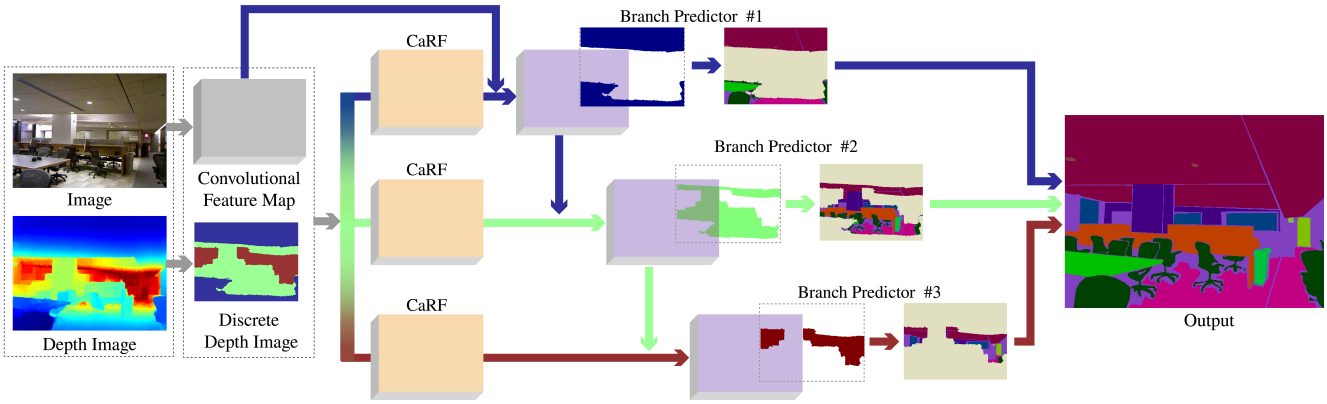


Figure 2: The overview of our cascaded feature network (CFN). Given the color image, we use CNN to compute the convolutional feature map. The discrete depth image is layered, where each layer represents a scene-resolution and is used to match the image regions to corresponding network branches that share the same convolutional feature map. Each branch has context-aware receptive field (CaRF), which produces contextual representation to combine with the feature from adjacent branch. The predictions of all branches are combined to achieve the eventual segmentation result.

regions. Therefore, to better represent and learn the variant object/scene relationships, appropriate features should be constructed for different scene-resolutions.

Regular Receptive Field To compute the representation of object/scene relationships, recently numerous segmentation networks [13, 38, 27, 25, 37, 24] enriched the contextual information of convolutional features using a set of receptive fields. Their receptive fields are in general predefined with regular forms of diverse sizes. However, such regular receptive fields are context-oblivious in the sense that they do not consider their extent with respect to the underlying image structure.

Branched Network Note that fully convolutional network (FCN) [32, 4] with multiple branches has been used to generate distinct features for distinct regions of interest, which are applicable to different scene-resolutions. Specifically, FCN has separate branches that can segment regions with different scene-resolutions. Although different branches are linked by the shared features, each independent branch only influences the shared convolutional features in the regions of the corresponding scene-resolution. It implies that, in the training phase, the shared convolutional features cannot be updated by signals that capture the relationship between the regions of different scene-resolutions. It inevitably limits the context of regions that can indeed effectively update the network.

Our Approach We address the above two problems in the context of RGB-D images segmentation. First, to make the feature more focused on the common visual characteristic of the observed scene, we introduce a context-aware receptive field (CaRF). The CaRF provides a better control on the relevant contextual information of the learned features. Our CaRFs are computed based on super-pixels, which are defined by the underlying scene structures. Thus, the contextual information provided by CaRF can alleviate negative effect of mixing the features of overly small or large regions. Second, we present a cascaded feature network (CFN) with parallel branches, each of which focuses on semantic segmentation of regions of certain scene-resolution. Figure 2 illustrates our CFN architecture. Each branch is equipped with a CaRF. It is trained and operated on a more focused context with similar scene-resolution. The combination of CaRF and cascaded network, enables regions in different scene-resolutions to communicate each other so as to wisely update shared convolutional features.

We show that the cascading of features enriches the contextual information of each branch and enhances the overall performance. The performance of our network is demonstrated on two public datasets. With our presented CFN, the mean intersection-over-union of 47.7 on the NYUDv2 dataset [33] and 48.1 on the SUN-RGBD dataset [35] are achieved, which outperform the state-of-the-art results.

2. Related Work

2.1. FCN for Semantic Segmentation

Fully convolutional network (FCN) [28] has been broadly used in semantic segmentation systems [1, 27, 38, 25, 24, 37]. The stacked pooling operations in FCN, however, inevitably reduce the image resolution, resulting in segmentation information loss on image regions. Some works are proposed to address this problem, for instances, applying atrous convolution to maintain relatively high-resolution information [1], or employing deconvolution operation to recover high-resolution regions from low-resolution ones [31].

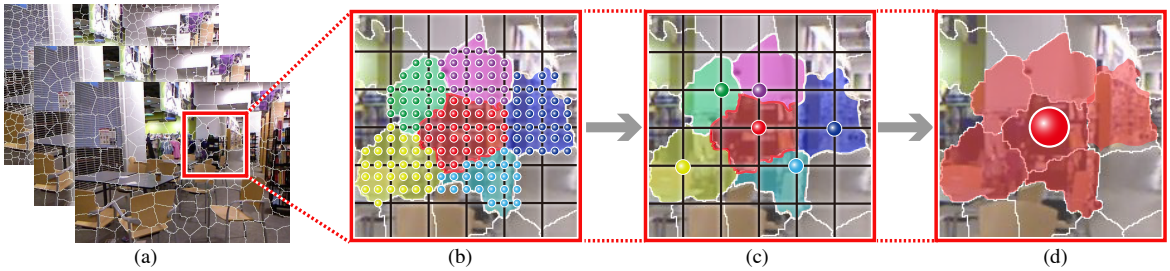


Figure 3: The two-level Context-aware Receptive Field (CaRF): (a) the image partitioned into super-pixels with different sizes; (b) at each node of the coarse grid we aggregate the features that reside in the same super-pixel; (c) the content of adjacent super-pixels is aggregated; (d) the aggregated content in a feature map represents a CaRF. The two-level CaRF is repeatedly applied to the images partitioned by super-pixels with diverse sizes. Note that the feature map has smaller resolution than the image due to down-sampling of network.

Contextual information of multiple receptive fields is used as well to alleviate the problematic prediction. Several works [1, 27, 38, 25] integrate graphical models to capture the context of multiple pixels. From another perspective, Lin et al. [24] and Zhao et al. [37] utilize the convolutional/pooling kernels with diverse sizes to capture different receptive fields of images. In this way, the contextual information is effectively enriched. Our method also makes use of the convolutional features extracted from receptive fields with different sizes. In contrast to [24, 37] that use different kernels, we control the size of super-pixel to capture receptive fields, which are more aware of the relationships between image regions. Similarly, the works [29, 16, 22] use super-pixel to group the convolutional features from a set of receptive fields. Nonetheless, different from ours, these works do not combine neighboring super-pixels, which may result in loss of relationship between super-pixels.

2.2. Semantic Segmentation of RGB-D Images

Semantic segmentation of RGB-D image has been studied for more than a decade [33, 11, 12, 36, 16]. Different from traditional semantic segmentation of RGB images [9, 30, 2], additional depth channel is available now. It allows better understanding of the geometric information of the scene images. Many prior works harness useful information from the depth channel. Silberman et al. [33] propose an approach to parse the spatial characteristics, such as support relations, by using RGB image along with the depth cue. Gupta et al. [11] use depth image to construct geometric contour cue to benefit both object detection and segmentation of RGB-D images.

Recently, CNN/FCN is used for learning features from depth to help the segmentation of RGB-D images. Couprie et al. [3] propose to learn CNN using the combination of RGB and depth image pairs such that the convolutional feature maintains depth information. Gupta et al. [12] and He et al. [16] encode depth image as HHA image [11], which maintains each pixel's horizontal disparity, height above ground, and the angle of the local surface normal. The networks trained on different modalities, e.g., RGB and HHA image, are fused by Long et al. [28] to boost the segmentation accuracy. Compared to direct fusion of segmentation scores as in [28], the network proposed by Wang et al. [36] produces better segmentation result by harness deeper correlation of RGB and depth image pairs.

In our scenario, depth information plays a more significant role in guiding the feature learning for the regions of different scene-resolutions. The depth image is layered to identify the scene-resolution of the region. An effective design of neural network structure is thus facilitated to consider the characteristic of the region in specific scene-resolution. This technique can be applied to benefit feature learning from different data modalities, as shown in results.

3. Context-aware Receptive Field

The receptive fields of common networks are predefined. Here, we present a *Context-aware Receptive Field* (CaRF) where the receptive field is spatially-variant and defined its extent according the local context. The idea is to aggregate convolutional features of local context into richer features that learn better the relevant content.

The contextual information generated by CaRF is controlled by adjusting the sizes of the super-pixels. For the regions in low scene-resolution, we select larger super-pixels that include more objects and scenes information, while in higher scene-resolution, we switch to finer super-pixels so as to avoid too much diverse information; see also Figure 3(a). The adaptive size of the super-pixels helps to capture the complex object/scene relationship in different regions. The relevant context comprises of the local neighborhoods of a super-pixel as shown in Figure 3(d). That is, an entry $M(h, w)$ in the feature map M is an aggregation of all the convolutional features that are within the super-pixel that contains (h, w) and its adjacent super-pixels.

Using such context-aware receptive fields rather than fixed regular ones, leads to better segmentation. In our ex-

periments, we apply CaRF on top of the convolutional feature of the network to gather more contextual information. As we shall show, the addition of CaRF makes a decent improvement in the semantic segmentation performance.

Note that the CaRFs overlap. Thus, to save significant computations of repeatedly integrating the same regions, the CaRFs are computed in two levels as elaborated below.

Given an image I, we utilize the toolkit [7] 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 Figures 3(b-c), at first, we sum features that reside on the same super-pixels. This context-aware summation produces a feature map $R \in \mathbb{R}^{C\times H\times W}$:

$$R(c,h,w) = \sum_{(h',w')\in\Phi(S_i)} F(c,h',w'), \tag{1}$$

where $(h,w) \in \Phi(S_i)$. $F \in \mathbb{R}^{C\times H\times W}$ denotes the common shared convolutional features of the widely-used CNN architectures, such as *fc7* of FCN-VGG [28] or *res5c* of ResNet [15], as illustrated in Figure 2. C is the number of channels with index c, and H and W are the height and width of the feature map. The spatial coordinate (h,w) uniquely corresponds to a center of 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. The local feature $R(c,h,w)$ remains the same for the set $\Phi(S_i)$.

At the second level (Figures 3(c-d)), we aggregate the features of R that are associated with adjacent super-pixels to model a new feature map $M \in \mathbb{R}^{C\times H\times W}$:

$$M(c,h,w) = R(c,h,w) + \sum_{S_j\in\mathcal{N}(S_i)} \sum_{(h',w')\in\Phi(S_j)} \lambda_j R(c,h',w'), \tag{2}$$

where $(h,w) \in \Phi(S_i)$. Here $S_j \in \mathcal{N}(S_i)$ means super-pixel S_i and S_j are adjacent, and $\lambda_j = \frac{1}{|\Phi(S_j)|}$ with that $|\Phi(S_j)|$ denotes the number of regular receptive field centers located within the super-pixel S_j. Again, the entry $M(c,h,w)$ remains the same for the set $\Phi(S_i)$, as the identical adjacent super-pixels provide the same context.

This process forms the contextual representation M used below, where each entry $M(h,w)$ represents a CaRF.

4. Cascaded Feature Network

We present a deep *Cascaded Feature Network* (CFN) for semantic segmentation of RGB-D images. CFN has multiple network branches for the segmentation in different scene-resolutions. The multiple-branch CFN allows distinct CaRF to provide specify contextual information for a certain scene-resolution. More importantly, the cascaded structure of CFN enables the information propagated from one branch to help the adjacent branch. In what follows, we elaborate the construction of CFN.

The architecture of the CFN is illustrated in Figure 2. Assume CFN has K branches, each of which accounts for the segmentation in a certain scene-resolution. The 1^{st} branch is for the highest scene-resolution. Given a depth image D, 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. Given a color image I as input, the k^{th} branch outputs the feature F_k as

$$F_k = F_{k-1} + M_k, \;\; k = 1, ..., K, \tag{3}$$

where K is the number of branches, and M_k is the contextual representation formulated in Eq. (2). We define $F_0 = F$, where F is the shared convolutional feature defined in Eq. (1). The feature F_k is in a combination form, which is modeled by adding the feature F_{k-1} with the contextual representation M_k produced by CaRF.

The feature F_k is fed to a predictor for segmentation. Given all the pixels assigned to the k^{th} branch, we denote their class labels as a set y_k, which is determined as:

$$y_k = f(F_k). \tag{4}$$

The function $f(:)$ is softmax predictor that is widely used for pixel-wise categorization. For the pixel that has the location (h,w), we denote $y_k(h,w)$ as its class label.

Combining the prediction results of all branches forms the final segmentation y on the image I.

Network Training We denote y^* as the ground-truth annotation of the image I. Using Eq. (4), we compute the segmentation of the image I. To train CFN for segmentation, the overall objective function is defined as:

$$J(F_1, ..., F_K) = \sum_{k=1}^{K} J_k(F_k), \tag{5}$$

where

$$J_k(F_k) = \sum_{(h,w)\in\Omega_k} L(y_k^*(h,w), y_k(h,w)). \tag{6}$$

J_k is the objective function for the k^{th} branch. The Ω_k denotes the set of pixels handled by the k^{th} branch. The function L is softmax loss for penalizing pixel-wise categorization error. The network training is done by minimizing the objective formulated as Eq. (5).

We utilize the standard back-propagation (BP) algorithm [21] to train CFN. In BP stage, the feature in Eq. (6) are updated in each iteration. To update the feature F_k, we use the definition of Eq. (3) and compute the gradient of

objective function J with respect to F_k as:

$$\frac{\partial J}{\partial F_k} \propto \frac{\partial J_k}{\partial F_k} + \frac{\partial J_{k+1}}{\partial F_{k+1}} \frac{\partial F_{k+1}}{\partial F_k} = \frac{\partial J_k}{\partial F_k} + \frac{\partial J_{k+1}}{\partial F_{k+1}}. \quad (7)$$

The update signal of F_k functions as the compromise between back-propagated information of the feature F_k and F_{k+1}. The update signal $\frac{\partial J_k}{\partial F_k}$ accounts for the k^{th} branch. With the cascaded structure connecting two branches, the signal $\frac{\partial J_{k+1}}{\partial F_{k+1}}$ of the $(k+1)^{th}$ branch influences the update of F_k in training phase. As each adjacently indexed branches communicate via the cascaded structure, we find that any two branches can be balanced in an effective way.

In the k^{th} branch, the update signal is passed from the combined feature F_k to the contextual representation M_k. It influences the update of the local regions of the share convolutional feature. To update the feature $R_k(c, h, w)$, which represents a local region of the share convolutional feature F, we use the definition of Eq. (2) and compute the gradient of objective function J with respect to $R_k(c, h, w)$ as:

$$\frac{\partial J}{\partial R_k(c,h,w)} \propto \frac{\partial J}{\partial M_k(c,h,w)} \frac{\partial M_k(c,h,w)}{\partial R_k(c,h,w)} + \sum_{S_{jk} \in \mathcal{N}(S_{ik})} \sum_{(h',w') \in \Phi(S_{jk})} \lambda_{jk} \frac{\partial J}{\partial M_k(c,h',w')} \frac{\partial M_k(c,h',w')}{\partial R_k(c,h,w)}, \quad (8)$$

where $(h, w) \in \Phi(S_{ik})$. As modeled by Eq. (8), the update of the local feature $R_k(c, h, w)$ is impacted by the signal of its neighborhoods satisfying $S_{jk} \in \mathcal{N}(S_{ik})$ and $(h', w') \in \Phi(S_{jk})$. Though this communication is defined on spatially-adjacent local regions, the non-adjacent ones can affect each other along a path of adjacent members.

With cascaded structure, one branch can receive the signals from other branches. Further, with the adjacent relationship defined by CaRF, the signals from other branches can be diffused to any local region in a branch. As a result, the share convolutional feature F can be updated by signals that capture the relationship between local regions in different branches.

5. Implementation Details

Preparation of Image Data The original RGB images are used as data source. In addition, we encode each single-channel depth image as a 3-channel HHA image introduced in [11, 12], 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 the network training, we use four common strategies, i.e., flipping, cropping, scaling and rotating of the image, to argument the training data.

Settings of CFN and CaRF CFN has multiple branches to handle different scene-resolutions. The number of branches is pre-defined before constructing the network. Each branch accounts for a certain range of depth value. We obtain the global range of depth value from all the depth maps provided by the datasets. For example, the depth value of NYUDv2 dataset varies from 0 to 102.7 meters. The global range is then divided by the number of branches. Given a pixel in the image, it is assigned to the corresponding branch with respect to its depth value. In our experiment, we compare the results of 1-, 2-, 3-, 4- and 5-branch CFNs. The super-pixels are controllable in our CaRF components. For lower scene-resolution, CaRF uses larger super-pixels to capture richer contextual information. Following this principle, we enlarger the scale, which is a parameter of the toolkit [7], to broaden the super-pixels. We empirically set the scales as $\{$1600, 3000, 4200, 6000, 10000$\}$ for the five branches, respectively.

Network Construction We modify the Caffe platform [18] to construct our network. Our network is based on the FCN [28]. The network structure pre-trained on ImageNet [5], i.e., VGG-16 [34], serves as the architecture on top of which we build our CFN. We apply atrous convolution [1] to achieve 8-stride network. We use RefineNet-152 [24], which is based on the prevalently deeper ResNet-152 [15], to further improve segmentation performance when we compare our CFN to state-of-the-art methods. We optimize the segmentation network using BP algorithm. The network is fine-tuned with a learning rate of 1e-10 for 60K mini-batches. After that, we decay the learning rate to 1e-11 for the next 40K min-batches. The size of each min-batch is set to 8. As suggested in [28], we use a heavy momentum 0.99 so as to achieve stable optimization on relatively small-scale data.

6. Results and Evaluation

To show the efficacy of our CFN and evaluate its performance, we tested on two public datasets: NYUDv2 [33] and SUN-RGBD [35]. 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 with state-of-the-art methods.

The NYUDv2 dataset [33] contains 1,449 RGB-D scene images. Among them, 795 images are split for training and 654 images are for testing. In [12], a validation set that comprises of 414 images, is selected from the original training set. We follow the segmentation annotations provided in [11], where all pixels are labeled by 40 classes.

Following the common way to evaluate semantic segmentation schemes [24, 37], we perform the multi-scale testing. Four scales, i.e., $\{0.6, 0.8, 1, 1.1\}$, are used to resize the testing image before feeding it to the network. The

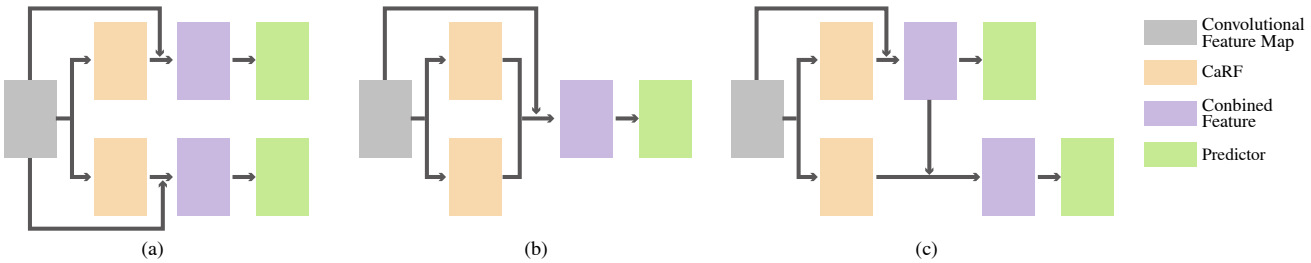


Figure 4: The network can have separate branches (a), combined branches (b) or cascaded branches (c). For clarity, we illustrate it with two branches only. Each network can be extended to have more branches.

# branches	1	2	3	4	5
mean IoU	31.2	33.8	**35.5**	35.0	34.6

Table 1: Sensitivities to the number of branches, e.g., $\{1, 2, 3, 4, 5\}$. The performances are evaluated on the NYUDv2 validation set. Each segmentation accuracy is reported in terms of mean IoU (%).

output scores of the four re-scaled images are then averaged and processed by dense CRF [20] for the final prediction. Following [28, 24, 12], we report on the segmentation performance in terms of mean intersection-over-union (IoU).

Sensitivities to the Number of Branches First, we report on our investigation of the sensitivity of our model to the number of branches. We tested with different number ($\{1, 2, 3, 4, 5\}$). For every case, we report the segmentation accuracy on the validation set. We train our CFN based on VGG-16 [34] model. The segmentation accuracies on the validation set are reported in Table 1.

The input to CFN includes RGB image for segmentation and depth image for splitting image regions for different branches. The performances of the different CFN configurations are listed in Table 1. We note the single-branch CFN achieves the accuracy score of 31.2, which is lower than the scores of other CFNs that have two or more branches. As only one CaRF is used in the single-branch network, specific contextual representations can not be achieved for different scene-resolutions. We find that 3-branch CFN achieves the best result. We also observe that further increasing the number of branches, i.e., using 4- or 5-branch CFNs, causes a performance drop. In these cases, larger super-pixels are used. It suggests that too large super-pixels are not suitable to use, as they may much diversify the object/scene classes and therefore distract the stable patterns that should be learned by CFN.

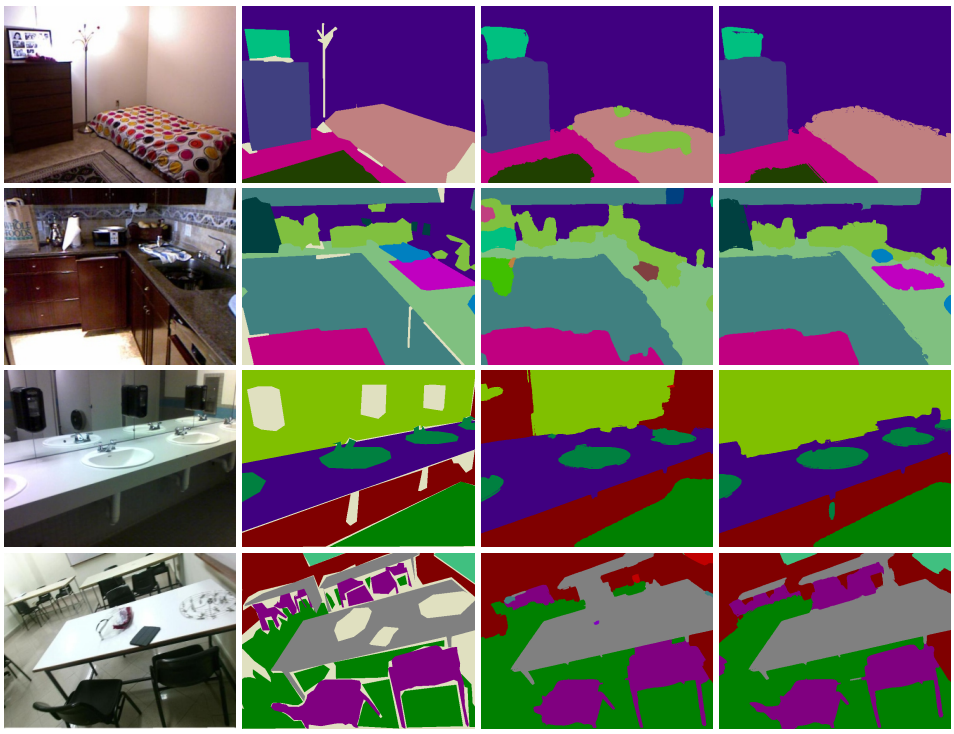
Strategies of Using CaRF CaRF defines the adaptive extent of the receptive field and plays a critical role in adjusting the contextual information for different scene-resolutions in our cascaded network. To demonstrate the importance of CaRF, we conduct an experiment that measure the performance of our CFN without CaRF.

CFN	strategy	mean IoU
single-branch	w/o CaRF	31.8
	w/ CaRF	**32.0**
multiple-branch	w/o CaRF	31.7
	w/ regular-RF	33.8
	w/ CaRF	**36.3**

Table 2: Strategies of using CaRF, evaluated on the NYUDv2 test set. Each segmentation accuracy is reported in terms of mean IoU (%).

We use RGB images to train different CFNs, and their results are listed in Table 2. First, we investigate the single-branch network. Without CaRF, the single-branch network degrades to the VGG-FCN [28], which yields the score of 31.8. This result is lower than the score of 32.0 produced by single-branch network that has CaRF. We also experimented with different scene-resolutions, and we train multiple-branch (3-branch) CFNs for comparison. Without CaRF, the multiple-branch CFN performs similarly as the single-branch counterpart. By adding CaRFs to different branches, CFN improves the segmentation accuracy to the score of 36.3. These comparisons manifest that CaRFs provide useful contextual information for different scene-resolutions.

We note that enlarging the regular receptive field can gather more contextual information as well. In the case of multiple-branch CFN, we thus use multiple regular receptive fields in place of CaRFs. This is implemented by using diverse average-pooling kernels to handle different scene-resolutions. This manner is similar to the method described in [37]. We hand-tune the average-pooling kernels to the sizes of $\{3, 5, 7\}$ to achieve a reasonable accuracy score of 33.8; see the entry *w/ regular-RF* in Table 2. Nonetheless, this score is still lower than that of multiple-branch CFN having CaRFs. The performance gap suggests that CaRF provides finer means to utilize contextual information than regular receptive field.



(a) Image (b) Ground-truth (c) Baseline (d) CFN

Figure 5: A sample of the comparison to the baseline model [24] and our CFN. The first two and last rows are scenes taken from NYUDv2 [33] and SUN-RGBD [35] dataset, respectively.

strategy	separate-branch	combined-branch	CFN
mean IoU	33.0	34.0	**36.3**

Table 3: Strategies of using CFN, evaluated on the NYUDv2 test set. Each segmentation accuracy is reported in terms of mean IoU (%).

Strategies of Using CFN We exploit cascaded structure to handle different scene-resolutions. In different cases, we evaluate the performance on segmentation and experiment with removing the connecting links between the branches. We compare the performances in Table 3.

Without cascaded structure, each branch accounts for the corresponding scene-resolution in an isolated way, as shown in Figure 4(a). This network has CaRFs integrated in all branches. Though CaRF provides contextual information for each scene-resolution, the information propagation between branches is lacking. It makes the shared convolutional feature oblivious of the relationship between regions in different scene-resolutions. In comparison to our CFN that connects different branches, as shown in Figure 4(c), the separate-branch network yields inferior performance.

The branches can be combined to segment image, as illustrated in Figure 4(b). With combined-branch network, all scene-resolutions share the same contextual information. The low scene-resolutions benefit from the local contextual information, however, mixing the contextual information is not desirable for high scene-resolutions. Thus, the performance of combined-branch network lags behind our CFN.

Comparisons with State-of-the-art Methods In Table 4, we compare our CFN with state-of-the-art methods that are also based on deep neural networks. According to the training and testing data, the methods to compare are divided into two groups.

In the first group, the methods use only RGB images for segmentation. In the column *RGB-input* of Table 4, we report the performances of these methods. We find that the deep network proposed by Lin et al. [24] achieves the best accuracy in this group. This network is based on ResNet-152 [15], which is much deeper than the previous ones used in [28, 19, 25]. It suggests that using deeper network can help improving segmentation accuracy.

In the second group, the methods take both RGB and depth images as input. We report the performances in the column *RGB-D-input* of Table 4. We note each depth image can be encoded as a 3-channel HHA image, which maintains richer geometric information as introduced in [11, 12]. Following Long et al. [28], we use HHA images to train segmentation network, in place of RGB image. Given an image, the segmentation network trained on HHA images is

RGB-input	mean IoU	RGB-D-input	mean IoU
		Gupta et al. [12]	28.6
		Fayyaz et al. [10]	30.9
		Deng et al. [6]	31.5
Long et al. [28]	29.2	Long et al. [28]	34.0
Kendall et al. [19]	32.4	Eigen et al. [8]	34.1
Lin et al. [25]	40.6	He et al. [16]	40.1
Lin et al. [24]	46.5	Lin et al. [24]	47.0
		CFN (VGG-16)	41.7
		CFN (RefineNet-152)	**47.7**

Table 4: Comparisons with other state-of-the-art methods on the NYUDv2 test set. Each segmentation accuracy is reported in terms of mean IoU (%).

used to compute 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. Using this fusion strategy and the network proposed by Lin et al. [24], the previous best result 47.0 is obtained. Compared to the network in [24] that uses RGB images only, the one using both RGB and HHA images improves the segmentation accuracy. As the network structures are based on ResNet-152, we conclude that the performance gap is solely attributed to using HHA images for assisting segmentation.

Our CFN belongs to the second group. We use RGB and HHA images for training and testing. Our CFN that is based on VGG-16 achieves the score of 41.7. Comparing to the previous network proposed by He et al. [16], which also use RGB and HHA image for training VGG-16 model, our CFN produces better results. We further use deeper model introduced in [24], and the score of 47.7 is achieved. This result is better than state-of-the-art methods. In the first two rows of Figure 5, we show the visual improvement against the baseline model of [24]. The comparison demonstrates that our CFN is compatible to different network structures and improves the segmentation accuracy.

Experiments on SUN-RGBD Dataset We conduct more experiments on the SUN-RGBD dataset [35], which comprises of 10,335 images labeled with 37 classes. We use 5,285 images for training and the rest for evaluation. SUN-RGBD dataset provides more images than the NYUDv2 dataset [33]. It thus can verify whether our CFN is able to effectively handle more diverse scene and depth conditions.

We show the segmentation accuracy of our CFN in Table 5. Again, the compared methods are divided into two groups. Similarly to the previous experiments, we compare our method to the group of methods that consider both RGB and HHA images as input. With VGG-16 model trained on RGB and HHA images, the previous best performance is produced by the method of Hazirbas et al. [14]. Using the same model and data, our CFN yields a better score of 42.5. Again, with a deeper model RefineNet-152 introduced in [24], we are able to achieve the accuracy score of 48.1, which outperforms the state-of-the-art results. The visualization results of our CFN on SUN-RGBD dataset [35] can be found in the last two rows of Figure 5.

RGB-input	mean IoU	RGB-D-input	mean IoU
Noh et al. [31]	22.6		
Long et al. [28]	24.1		
Chen et al. [1]	27.4		
Kendall et al. [19]	30.7	Long et al. [28]	35.1
Lin et al. [25]	42.3	Hazirbas et al. [14]	37.8
Lin et al. [24]	45.9	Lin et al. [24]	47.3
		CFN (VGG-16)	42.5
		CFN (RefineNet-152)	**48.1**

Table 5: Comparisons with other state-of-the-art methods on the SUN-RGBD test set. Each segmentation accuracy is reported in terms of mean IoU (%).

7. Conclusions

Recent developments in semantic segmentation of image have leveraged the power of convolutional networks that are trained on large datasets. In our work, we have shown that with depth information we can further increase the accuracy of the segmentation. The increased performance is attributed to the use of context-aware receptive fields, which have irregular extents that adapt and learn relevant data in the appropriate scene-resolution. We have presented a cascaded feature network that takes advantage of the spatially-variant receptive field to enable a flexible modeling of the data with a good balance between image regions in different scene-resolutions. We have showed that our CFN is efficient and outperforms recent state-of-the-art methods.

In the future, we would like to further explore the potential of the context-aware receptive fields, first for semantic segmentation of RGB images with no depth, and second, for other applications rather than recognition or segmentation. Another research direction is extending context-aware receptive fields to 3D or higher dimension.

Acknowledgments

We thank the reviewers for their constructive comments. This work was supported in part by National 973 Program (2015CB352501, 2015CB351706), NSFC (61522213, 61379090, 61232011, 61233012, U1613219), Guangdong Science and Technology Program (2014TX01X033, 2015A030312015, 2016A050503036), Shenzhen Innovation Program (JCYJ20151015151249564) and Natural Science Foundation of SZU (827-000196).

References

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

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

[3] C. Couprie, C. Farabet, L. Najman, and Y. LeCun. Indoor semantic segmentation using depth information. *arXiv*, 2013.

[4] J. Dai, Y. Li, K. He, J. Sun, et al. R-fcn: Object detection via region-based fully convolutional networks. In *NIPS*, 2016.

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

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

[7] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013.

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

[9] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010.

[10] M. Fayyaz, M. H. Saffar, M. Sabokrou, M. Fathy, R. Klette, and F. Huang. Stfcn: Spatio-temporal fcn for semantic video segmentation. *arXiv*, 2016.

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

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

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

[14] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *ACCV*, 2016.

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

[16] Y. He, W.-C. Chiu, M. Keuper, and M. Fritz. Rgbd semantic segmentation using spatio-temporal data-driven pooling. *arXiv*, 2016.

[17] F. Husain, H. Schulz, B. Dellen, C. Torras, and S. Behnke. Combining semantic and geometric features for object class segmentation of indoor scenes. *IEEE Robotics and Automation Letters*, 2017.

[18] 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 *ACM international conference on Multimedia*, 2014.

[19] A. Kendall, V. Badrinarayanan, and R. Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv*, 2015.

[20] P. Krähenbühl and V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *NIPS*, 2011.

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

[22] X. Liang, X. Shen, J. Feng, L. Lin, and S. Yan. Semantic object parsing with graph lstm. In *ECCV*, 2016.

[23] D. Lin, J. Dai, J. Jia, K. He, and J. Sun. Scribblesup: Scribble-supervised convolutional networks for semantic segmentation. In *CVPR*, 2016.

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

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

[26] 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 *ECCV*, 2014.

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

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

[29] M. Mostajabi, P. Yadollahpour, and G. Shakhnarovich. Feedforward semantic segmentation with zoom-out features. In *CVPR*, 2015.

[30] 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 *CVPR*, 2014.

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

[32] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015.

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

[34] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv*, 2014.

[35] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *CVPR*, 2015.

[36] 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 *ECCV*, 2016.

[37] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *arXiv*, 2016.

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