UNIVERSITY OF AMSTERDAM

UvA-DARE (Digital Academic Repository)

3D Neighborhood Convolution: Learning Depth-Aware Features for RGB-D and RGB Semantic Segmentation

Chen, Y.; Mensink, T.; Gavves, E.

[Link to publication](#)

3D Neighborhood Convolution:
Learning Depth-Aware Features for RGB-D and RGB Semantic Segmentation

Yunlu Chen[†], Thomas Mensink[‡], and Efstratios Gavves[†]

[†]University of Amsterdam
[‡]Google Research, Amsterdam
{y.chen3, e.gavves}@uva.nl, mensink@google.com

Abstract

A key challenge for RGB-D segmentation is how to effectively incorporate 3D geometric information from the depth channel into 2D appearance features. We propose to model the effective receptive field of 2D convolution based on the scale and locality from the 3D neighborhood. Standard convolutions are local in the image space (u, v), often with a fixed receptive field of 3x3 pixels. We propose to define convolutions local with respect to the corresponding point in the 3D real world space (x, y, z), where the depth channel is used to adapt the receptive field of the convolution, which yields the resulting filters invariant to scale and focusing on the certain range of depth.

We introduce 3D Neighborhood Convolution (3DN-Conv), a convolutional operator around 3D neighborhoods. Further, we can use estimated depth to use our RGB-D based semantic segmentation model from RGB input. Experimental results validate that our proposed 3DN-Conv operator improves semantic segmentation, using either ground-truth depth (RGB-D) or estimated depth (RGB).

1. Introduction

Most deep networks specialized on semantic segmentation currently follow a fully convolutional architecture [21, 1, 20, 4, 40, 5] on 2D images and return pixel-level classifications. However, as shown in [8, 21, 14], semantic segmentation improves when 3D or depth information is available by specialized hardware. One way is that the local segmentation boundary can be refined by the scene geometry, where there is an occlusion boundary in the 2D image projection. More generally, the global high-level semantics can benefit from the 3D scene distribution by removal of the possible projection ambiguity. In this work, we investigate if and how we can embed 3D scene information into the 2D
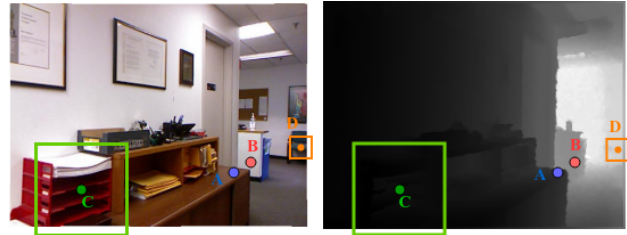

Figure 1. The concept of 3D neighborhood and its difference from 2D neighborhood. *Locality from depth*: A and B are neighbours in 2D image but not in 3D space; *Scale from depth*: D is further away than C, so the 3D neighborhood of D is smaller on 2D image than C. For both cases we can find a explicit cue from the depth value.

convolution in RGB images.

Recently, great progress has been witnessed in deep learning on 3D data, such as voxels [32, 3] and point clouds [24, 25]. Yet, 3D data face problems which prevent it from large-scale or real-time usage. This changes our attention to 2.5D representations in the form of depth maps and RGB-D data for two reasons. On the one hand, processing 2.5D data is almost as computationally efficient as 2D data, contrast to 3D representations that blow up computations. On the other hand, 2.5D RGB-D data can easily be acquired by either low-cost 2.5D commercial depth sensors like Kinect or disparity maps from binocular cameras,What is more, in the absence of sensory depth, monocular depth estimation methods [9, 8, 18, 36, 10] has recently been able to provide reasonable depth maps, even with RGB images alone. The better availability and efficiency of 2.5D data render them an inexpensive and yet effective solution for incorporating geometry information.

A reasonable question, therefore, is how to effectively incorporate depth into a model, such that to learn depth-aware features amenable for scene semantic segmentation. In this paper, we integrate depth into the 2D convolutional operation. We do not just add depth as additional input channels

Figure 2. Visualization of the effective receptive fields [22] sampled at three certain points. Left: original image. Middle: effective receptive field of 3DN-Conv. Right: effective receptive field of standard convolution. Results in six sequential layers of 3DN-Convs.

as usual, but we define a *3D neighborhood* and modify the 2D convolutional filters with it. As illustrated in Figure 1, the two properties of 3D neighborhood are *depth locality* and *scale*. The depth locality comes from the locality in the 3D space; the varying scale in 2D image plane is due to the requirement of scale consistency in 3D. Depth locality and scale are important because they determine the receptive field. Taking the perspective that convolution operation is a *local function*, we manage to learn more 3D-aware features by tuning the receptive field of the 2D convolution according to the 3D neighborhood.

We present 3D Neighborhood convolutions (3DN-Conv) In 3DN-Conv the receptive fields dynamically adapt to the local depth structure. To be specific, the size of the receptive field is inversely proportional to the depth at the same pixel location, determined by the rule of *scale*, and only the region within a certain range of depth are incorporated to the convolution, according to the rule of *depth locality*. The visualization in Figure 2 supports that the effective receptive field of our 3DN-Conv bears awareness of both properties of scale and depth locality. We also use the depth map returned by the proposed Depth Discriminative Feature Network (D-DFN) for semantic segmentation, to obtain depth-aware pixel-level label predictions.

We make the following contributions. *First*, we propose 3D Neighborhood Convolutions (3DN-Conv) a novel spatially variant and depth-aware convolutional operator. The proposed convolution considers depth as a cue for both locality along the camera z-axis and the receptive field scale of the kernel. To the best of our knowledge, we are the first to propose a convolution operation that explicitly considers both aspects. Experiments show that our 3DN-Conv is more effective than other methods to incorporate depth for semantic segmentation. *Second*, we propose a depth estimation model D-DFN that recovers accurate local depth gradients and sharp 3D edges, compared with the state-of-the-art depth estimation algorithms. *Third*, we show that the proposed depth estimation can be successfully combined with RGB-D segmentation models and reach accuracies for which normally one requires depth information by specialized hardware sensors.

2. Related Works

RGB-D semantic segmentation RGB-D segmentation extends RGB semantic segmentation [21, 1, 4, 40, 5, 38]. A widely applied method (e.g. in [21, 8, 6]) is to encode depth into hand-craft HHA features [13]: horizontal disparity, height above ground and angle with gravity. Hazirbas *et al*. [14] uses a separate encoder to process depth channel and fuses the feature maps at every stage. Qi *et al*. [26] built a 3D k-nearest neighbor graph network on point clouds with extracted features from a CNN.

In this work, we focus on using depth to build spatially variant convolutional filters, with the advantage to learn better geometry-aware features by modeling the receptive field of the 2D convolution in accordance with the 3D local neighborhood. Following this trend, Wang and Neumann [34] augments the standard convolution by adding depth similarity as a weight term to consider the locality on dimension of depth. Different from [34], depth can also be a cue for scale [23, 6]: pixels with larger depth values are processed with convolutions with a smaller receptive field.

Unlike those works, our 3D neighborhood is the first concept framework to explicitly cover both the scale *and* the locality from depth in theory, which learns better 3D-aware features, as we will show experimentally.

Supervised monocular depth estimation We denote with monocular depth estimation (MDE) the task to recover the depth map from a single RGB image as input. While handcrafted features and probabilistic graphical models are used for MDE in early years [28, 29], recent methods benefit from learned structural deep features for local and global contextual information. Eigen *et al*. [9, 8] proposed a coarse-to-fine network to refine the output depth from a coarse prediction stage by stage. Laina *et al*. [18] introduced residual block design into fully convolutional network. Xu *et al*. [36] fuses multi-scale depth output in different stages by Conditional Random Fields (CRFs). Fu *et al*. [10] proposed ordinal regression as multiple binary classifications for pixel-wise depth, which achieves current state-of-the-art performance.

Adaptive Convolutions Our work also relates to generic adaptive convolutions, where local image filters are defined based on a set of different basis functions [17], features of the previous layer [33, 2], the correlation of the input or features [35, 30], or a different modality [34]. In particular, the most similar variants to our methods are the spatial sampling location methods, where filters are locally adaptive based on the spatial structure of the data itself [7, 42, 39, 27]. In contrast to these works that learn the local filter from the data, our convolution filter is defined using depth as privileged information.

3. 3D Neighborhood Convolution

In this section, we describe our 3D Neighborhood Convolution (3DN-Conv) models.

3.1. 3D neighborhood in RGB-D coordinates

Every 2D pixel location on the image frame corresponds to local neighborhood in the 3D space. While a convolutional filter normally has a predefined receptive field, we argue that the spatial extent of the convolutional filter should depend on the 3D neighborhood around the real-world point projected into a particular pixel. To define the 3D neighborhood around an image location, we consider the 3D cube around real-world point $\mathbf{p} = (p_x, p_y, p_z)$, with a radius of σ. Which we subsequently approximate from 2D image coordinates and depth (p_u, p_v, d), resulting in depth aware 2D filters in image space.

Under the constraints of a classic pin-hole camera model, the image coordinates of real-world point $\mathbf{p}$ are:

$$\begin{cases} p_u = \mu \dfrac{p_x}{p_z}, \\ p_v = \mu \dfrac{p_y}{p_z}, \\ d = p_z, \end{cases} \qquad (1)$$

where μ relates to the camera focal length. A σ neighborhood in 3D around $\mathbf{p}$ can be approximated by the following 2D image neighborhood:

$$\begin{cases} p'_u \in [\mu \frac{p_x - \sigma}{p_z}, \mu \frac{p_x + \sigma}{p_z}] = [p_u - \mu \frac{\sigma}{d}, p_u + \mu \frac{\sigma}{d}], \\ p'_v \in [\mu \frac{p_y - \sigma}{p_z}, \mu \frac{p_y + \sigma}{p_z}] = [p_v - \mu \frac{\sigma}{d}, p_v + \mu \frac{\sigma}{d}]. \end{cases} \qquad (2)$$

Which shows that a 3D cube with size σ corresponds to an image based convolution operator with a receptive field of:

$$\Delta p_u = \Delta p_v = \mu \frac{\sigma}{d}, \qquad (3)$$

which suggest that the receptive field of the 2D kernel should be inversely proportional to the depth d, instead of $\Delta p_u = \Delta p_v$ being a predefined filter size of the convolution. The required depth value d can be estimated from the z-buffer of the RGB-D image.

The value of depth d is not influenced by camera projection, yet a direct measurement of the 3D world, therefore:

$$\Delta_d = \sigma, \text{ and } d' \in [d - \sigma, \quad d + \sigma]. \qquad (4)$$

This shows that a local neighborhood on the depth channel is defined by the radius of the 3D neighborhood.

In the rest of this section we use the insights from Eq. 3 and Eq.4 for the design of our 3DN-Conv operator.

3.2. Depth locality

For clarity of presentation, we first formulate the standard convolutional filter for pixel i as follows:

$$\mathbf{y}_i = f \left(\mathbf{b} + \sum_{j \in \mathcal{N}_i} \mathbf{W}_{n_{ji}} \mathbf{x}_j \right), \qquad (5)$$

where $\mathbf{y}_i$ denotes the D dimensional output vector for pixel i, and $\mathbf{x}_j$ denotes the C dimensional input vector for pixel j. Furthermore, $f(\cdot)$ denotes the activation function (*e.g.* ReLU), $\mathbf{b}$ denotes the bias ($\mathbf{b} \in \mathbb{R}^{1 \times D}$), $j \in \mathcal{N}_i$ enumerates over the spatial neighborhood around pixel i, n_{ji} denotes the relative position of j with regard to i inside the neighborhood to select the relevant slice of the filter $\mathbf{W}$, where for each relative location we learn a filter $\mathbf{W} \in \mathbb{R}^{D \times C}$.

Modeling depth locality From Eq. 4 we know that the locality of depth should be within a range of $[d_i - \sigma, d_i + \sigma]$. To use this as *effective* filter size, we use a function that decays with depth as the local window to reweigh the local convolution kernel, which is equivalent to reweighing the local input features.

We evaluate different functions for reweighing, see Figure 3. Intuitively a step function, which uses a hard threshold on the locality seems an obvious choice, yet this fails in practice. We argue this is because 2D receptive field can gradually grow along with more convolution kernels, while receptive field on depth can hardly grow with 2D convolution. By noting that the effective receptive field follows the shape of a Gaussian distribution, see *e.g.* [22], we choose to model the window function by a 1-D Gaussian function $N(d_i, \sigma)$, where σ, the size of the expected 3D neighborhood, is used as the standard deviation, the size of the Gaussian window. This yields the following convolutional filter:

$$\mathbf{y}_i = f \left(\mathbf{b} + \sum_{j \in \mathcal{N}_i} \mathcal{L}_{ji} \mathbf{W}_{n_{ji}} \mathbf{x}_j \right), \text{ where} \qquad (6)$$

$$\mathcal{L}_{ji} = \exp(\frac{d_j - d_i}{\sigma})^2. \qquad (7)$$

In the following we denote this local convolutional filter with: $\mathbf{W}_{n_{ji}}^{\mathcal{L}} = \mathcal{L}_{ji} \mathbf{W}_{n_{ji}}$.

In standard 2D CNNs the effective receptive field of a kernel grows per layer, due to the aggregation of information and the use of pooling layers. Therefore, we scale σ
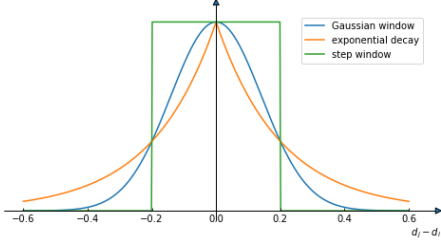
Figure 3. Illustrating different locality functions. A sharp step function (green) loses flexibility to incorporate non-local features. The exponential decay function (orange), used in [34], has too long tails. We use the Gaussian curve (blue), since it is in accordance with the effective receptive field theory [22].

per layer with respect to the size of 2D convolution kernel, so that the effective 3D neighborhood remains similar. According to Eq. 3, the size of the 2D kernel is proportional to the desired size of the 3D neighborhood. However, in different layers in the ConvNet, as the features are usually downsampled in 2D spatial resolution stage by stage, the relative size of the canonical 2D kernel, which we regard as the size of the 3D neighborhood σ, is actually enlarged as the network goes deeper. So σ varies with layer in the network. The exact scaling factor depends on the used architecture and is clarified in the supplementary material.

3.3. Neighborhood scale selection

We know that the receptive field of the 2D convolution Δ_d should be inversely proportional to the local depth value d, see Eq. 3. In order to incorporate this into our design, we choose to bilinearly resample the convolutional patch, $i.e.$, the 2D local neighborhood $\mathcal{N}_i$ to a rescaled version $\mathcal{N}_i^S$. This yields the following convolutional filter:

$$\mathbf{y}_i = f\left(\mathbf{b}_{n_{ji}} + \sum_{j \in \mathcal{N}_i^S} \mathbf{W}_{n_{ji}}^{\mathcal{L}} \mathbf{x}_j\right), \qquad (8)$$

where $\mathcal{N}_i^S$ is the scaled 2D neighborhood, defined by:

$$\mathcal{N}_i^S := \{j | \delta(i,j) \leq r^S\}, \quad \text{where} \qquad (9)$$

$$r^S = \frac{d_0}{d_i} r_0 \qquad (10)$$

in which $\delta(i,j)$ denotes the distance of the two points i and j in 2D space. r^S is the scaled kernel size, and r_0 is the original kernel size and it usually equals to dilation rate. In practice, we need to set a canonical depth value d_0 as a hyperparameter so that the size of the local convolution patch adapts according to Eq. 10.

We include an overview in Figure 4 on how we build our 3DN-Conv to model the scale and depth locality. In practice, we first scale the convolution patch size and sample the scaled neighborhood by Eq. 9 and Eq. 10, and then adapt the local filter to incorporate depth locality by Eq. 6 and Eq. 7.

3.4. Leverage RGB edges

The first few layers of standard convolutional networks extract edge features and other low level features. While our 3D-Neighborhood Convolution incorporates features in accordance with their real world spatial location, the limitation of looking at only 3D neighbors could be to lose edge kind features. Especially the edges caused by depth occlusions, since both sides of the occlusion edge are not regarded as the neighborhood in our convolution being aware of 3D neighborhood.

In order to leverage edge and other low-level features, we combine standard 2D convolution filters, denoted by $\mathbf{W}^{\mathcal{E}}$ for edge features with the proposed 3DN-Conv:

$$\mathbf{y}_i = f\left(\mathbf{b} + \sum_{j \in \mathcal{N}_i^S} \mathbf{W}_{n_{ji}}^{\mathcal{L}} \mathbf{x}_j + \mathbf{W}_{n_{ji}}^{\mathcal{E}} \mathbf{x}_j\right). \qquad (11)$$

Note that we deliberately use a separate set of parameters for $\mathbf{W}^{\mathcal{E}}$ without any weight sharing with $\mathbf{W}^{\mathcal{L}}$ to ensure the two kernels learning discriminate features. This joint convolution is only applied in the first stage of convolution layers `conv_1` in the ResNet backbone.

In depth comparison with [34] Our proposed 3DN-Conv bears resemblance to the depth-aware convolution operator proposed by Wang and Neumann [34]. The depth-aware spatial-variant convolution reweighs features to be convolved by an exponential function that measures the difference in depth. We clarify some significant differences: (i) We explicitly model a scaling function for our kernel, while [34] only considers depth similarity. This is the largest difference and traces back to the fact that our motivation being different from [34] as we design our kernel based on modeling the 3D neighborhood. (ii) Our local weight function is a Gaussian depth locality function to resemble the shape of the effective receptive field. While [34] uses an exponential decay function (see Figure 3). (iii) We use varying depth locality window size σ in different layers in the network while in [34] the window size is fixed in the network (see the last paragraph in Section 3.2). (iv) We include the edge convolution component to enhance low-level features.

4. Learning Depth for RGB-D Segmentation

In RGB-D segmentation, depth provides extra information such as 3D boundaries to eliminate some ambiguities in projected 2D images. Unfortunately, it is not always possible to have sensory depth as this requires specialized equipment. For this reason, we propose to first estimate depth, followed by semantic segmentation using our 3DN-Conv described in the previous section. The estimated depth map should be locally correct, including fine local details, sharp depth boundaries and consistent depth gradients, in order to guide the semantic segmentation model.
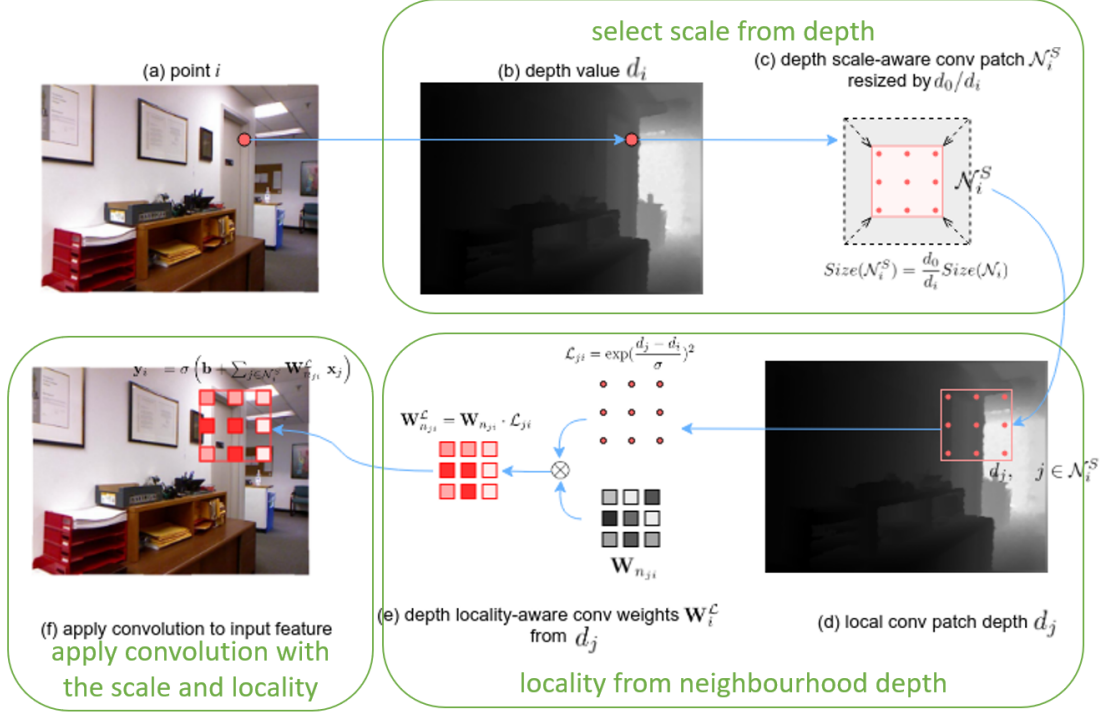
Figure 4. A practical overview of 3DN-Conv. The depth *scale*-aware conv patch $\mathcal{N}_i^S$ and the depth *locality*-aware conv weights $\mathbf{W}_i^{\mathcal{L}}$ (step (c) and step (e)) are calculated from depth (step (b), by Eq. 9 and Eq. 10, and step (d), by Eq. 6 and Eq. 7).
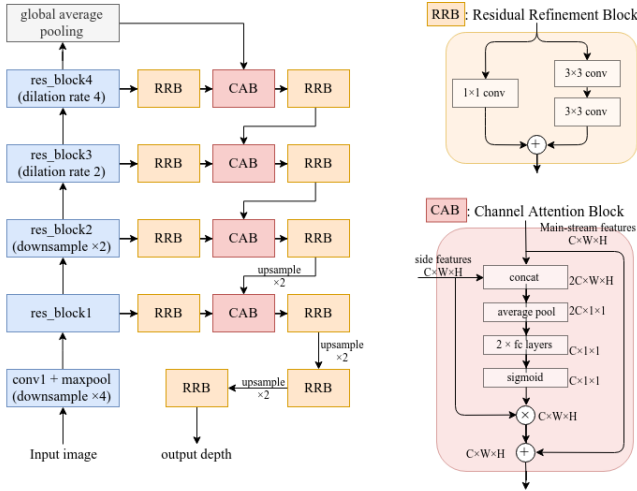

Figure 5. Illustration of Depth Discriminative Feature Network (D-DFN) for depth prediction. We experimentally show that estimating depth from RGB and then use our depth-aware 3DN-Conv for semantic segmentation improves over using RGB segmentation.

4.1. Depth Discriminative Feature Network

The task of depth estimation shares some common properties with semantic segmentation. For one, both are pixelwise prediction tasks. Also, both tasks require not only recognizing both global contextual information as well as local details, but also a successful fusion strategy for combining the two. It stands to reason, therefore, that depth estimation

and semantic segmentation may can benefit from each other when it comes to designing a suitable network architecture.

Inspired by the recently proposed semantic segmentation network Discriminative Feature Network (DFN) [38], we describe our proposed a novel model for estimating depth specialized for encoding local depth structure. In DFN architecture, the Channel Attention Block module (CAB) is proposed to incorporate features at multiple scales, with the idea originate from channel attention [16].

We describe the architecture of our D-DFN. The model structure is illustrated in Figure 5, comprising an encoder and a decoder architecture. The encoder is a ResNet-101 backbone, followed by a global pooling layer for capturing global information. The Residual Refinement Block upconvolves the high-level features to the output stage by stage. The Channel Attention Block incorporates features from different stages with channel-attention. See supplementary material for a more detailed comparison of our D-DFN and the original DFN architecture.

Training Loss. We adopt the L_1 loss for depth error

$$L_{depth} = \frac{1}{N} \sum_i |d_i - \hat{d}_i|. \tag{12}$$

We also consider an auxiliary loss for incorporating depth gradients and thus learning fine local depth details. The auxiliary loss takes the form

$$L_{grad} = |\nabla_x d_i - \nabla_x \hat{d}_i| + |\nabla_y d_i - \nabla_y \hat{d}_i|, \tag{13}$$

	mIoU(%)	mAcc(%)
Baseline (RGB only)	29.6	41.2
Depth-aware conv [34]	36.9	49.0
$\mathbf{W}^{\mathcal{L}}$	37.2	49.4
$\mathbf{W}^{\mathcal{L}} + \mathcal{N}^{\mathcal{S}}$ (discretized dilation rate)	38.5	51.2
$\mathbf{W}^{\mathcal{L}} + \mathcal{N}^{\mathcal{S}}$ (bilinear sampling)	39.1	51.7
3DN-Conv (final)	**39.3**	**52.4**

Table 1. Semantic segmentation performance on NYUDv2, with components in 3DN-Conv, trained from scratch.

where N denotes the total number of pixels, d_i the predicted depth value at the i-th pixel and $\hat{d}_i$ is the ground truth. This auxiliary loss enforces smoothness, by penalizing inconsistent depth gradients [19, 15]. The final loss is a weighted combination: $L_{depth} + \lambda_{grad} L_{grad}$, with $\lambda_{grad} = 1$.

5. Experimental Results

We provide an extensive evaluation of our 3DN-Conv. We adopt Deeplab V3 [5] and DFN [38] as our baseline semantic segmentation models. The popular Deeplab V3 is used for most of the experiments and ablation studies, while DFN is for compare with state-of-the-art methods as it is empirically more powerful for complex scene segmentation. Following [21], we adopt the following common metrics to evaluate semantic segmentation: mean intersection-over-union (mIoU), mean accuracy (mAcc) and pixel accuracy (Acc). For more implementation details we kindly ask readers to refer to the supplementary material.

Datasets Evaluation is performed on two RGB-D segmentation datasets: NYUDv2 [31] and KITTI [37]. NYUDv2 contains a total of 1,449 RGB-D image pairs from 464 different scenes. The dataset is divided into 795 images from 249 scenes for training and 654 images from 215 scenes for testing. We use the 40-class segmentation setting [12]. KITTI provides parallel camera and LIDAR data for outdoor driving scenes. We use the semantic segmentation annotation proposed in [37], which contains 70 training and 37 testing images and annotations in 11 classes.

5.1. Depth-aware convolutions: ablation study

We examine the importance of each of the proposed components of 3DN-Conv. We use our implementation of depth-aware conv in [34] and applied on the same ResNet50-DeeplabV3 model. We evaluate the performance of semantic segmentation on the NYUDv2 dataset.

The results are shown in Table 1. We observe the following: First both depth-aware convolution methods improve over the RGB segmentation baseline by a large margin, as was also noted in [34]. Our improved 3DN-Conv, using $\mathbf{W}^{\mathcal{L}}$ with a Gaussian window brings an additional improvement over the exponential decay in [34]. Further-

	mIoU(%)	mAcc(%)
Baseline (RGB only)	35.1	44.1
Depth-aware conv [34]	38.1	48.0
3DN-Conv (this paper)	**39.2**	**50.8**

Table 2. Semantic segmentation on KITTI, trained from scratch

more, we examine integrating scale adaptation to the proposed depth-aware convolution in two ways. First, the integration of scale can be done in a discretized manner. The depth intervals are first binned. Then, different dilation rate kernels are considered, similar to [6]. The second way of integrating scale is by considering bilinear rescaling of the convolutional receptive field, as described in Sec. 3.2. With the bilinear rescaling a continuous value for the scale is returned, thus allowing for more fine-tuned convolutions. In any case, per location a single scale is selected. We observe that adding scale adaptation improves the depth-aware convolutions considerably, especially when considering continuous scale values. This is not surprising, as the distribution of depth values is highly non-uniform, thus it is not straightforward how to bin fairly. Last, when considering also the RGB-only component for the convolution, $\mathbf{W}^{\mathcal{E}}$, the performance improves further in all three metrics.

We further evaluate on KITTI dataset [11, 37] to examine our method in various types of scenes. As in Table 2, we see that our 3DN-Conv outperforms both RGB baseline and the depth-aware convolution in [34].

We conclude that the proposed depth-aware convolutions improve semantic segmentation, especially when considering adaptive scaling of the receptive field, as well as additional RGB-only filters.

5.2. Fusing depth & RGB for segmentation

Next, we are explore the optimal way of integrating depth information for semantic segmentation. Specifically, we consider the following choices: (i) **early-fusion** of HHA features (ii) **late-fusion** of HHA features, (iii) **feature reweighting (modulation)**: feature map rescaled by depth, with a simple linear model, and (iv) **feature reweighting (non-local)**: feature map rescaled by depth, with non-local attention [35]. See supplementary material for the details.

For all the aforementioned methods we use the same backbone architecture and training pipeline. We compare the four methods of fusion above with the depth-aware convolutions from [34], as well as our 3D-Neighborhood convolution. We report results in Table 3.

We observe that the methods which perform convolution-level incorporation of depth, including the depth-aware convolution [34] and our 3DN-Conv, outperforms other methods by a large margin. The other methods help to improve the performance compared to the baseline by introduction of depth information. However,

	mIoU(%)	mAcc(%)
Baseline (RGB only)	29.6	41.2
HHA (early-fusion)	32.0	43.5
HHA (late-fusion)	33.7	44.4
Feature reweighting (modulation)	31.5	43.4
Feature reweighting (non-local)	32.6	47.2
Depth-aware conv [34]	36.9	49.0
3DN-Conv (this paper)	**39.3**	**52.4**

Table 3. Semantic segmentation performance on NYUDv2, with different method to incorporate depth into RGB network, on NYUDv2, trained from scratch. The spatial-variant kernel methods ([34] and ours) outperform other methods by a large margin.

the fusion methods are simply processing the HHA encoding of depth as extra channels in the network, and the assumptions of the feature reweighting method (see supplementary material) are too simple to capture the influence from the depth map. Overall, these methods to incorporate depth into deep network are regardless of the essence of geometry. We conclude that our 3DN-Conv is able to capture better depth-aware features by convolution-level incorporation and an explicit modeling of 3D geometry.

5.3. Segmentation with estimated depth

Next, we evaluate whether estimated depth can be used to improve semantic segmentation, in a similar way like depth returned by specialized sensory equipment. First, we examine the quality of the depth estimations. Then, we examine the benefits of using depth to semantic segmentation.

Depth estimation Following [9], we evaluate the depth estimation performance by taking the 304×228 center crop out of the downsampled image. As the goal is to use depth as a cue for semantic segmentation, it is important to have accurate local depth estimation. To this end, we evaluate depth estimation with global and local metrics, that is (i) Root mean squared error (rms): $\sqrt{\frac{1}{N}\sum_i (d_i - \hat{d}_i)^2}$, (ii) Mean log10 error (log): $\frac{1}{N}\sum_i ||\log_{10} d_i - \log_{10} \hat{d}_i||$. (iii) Gradient error (grad) $|\nabla_x d_i - \nabla_x \hat{d}_i| + |\nabla_y d_i - \nabla_y \hat{d}_i|$. Whereas the root mean squared error and the mean log10 error focus on the global evaluation of depth estimation, the local gradient error measures how well the local depth details are predicted. We report results in Table 4.

We observe that the proposed D-DFN improves in terms of both the global mean log10 error, however, it performs slightly worse in terms of root mean squared error. We attribute this to the fact that the logarithm scale normalizes the possible output values in a more reasonable range, in which regression is easier. Further, the proposed method improves the local gradient error by a noticeable 30% compared to [10]. This is important, as for semantic segmentation the local depth structure indicates the presence of se-

Figure 6. Depth estimation examples. Left top: input image. Right top: ground truth. Left bottom: predicted depth by DORN [10]. Right bottom: prediction by our D-DFN. Prediction by DORN shows artifacts and the local depth edges or gradient consistency are not well recovered, while our D-DFN output much sharper local depth structure.

	Global		Local
	RMS	log10	Local grad
Laina *et al.* [18]	0.573	0.055	0.137
DORN [10]	**0.509**	0.051	0.140
D-DFN (this paper)	0.528	**0.049**	**0.092**

Table 4. Depth estimation performance. Our D-DFN model achieves global results that is close-to the state-of-the-art model DORN, with significantly overwhelming local gradient error.

	Depth	mIoU(%)	mAcc(%)
Baseline	/	29.6	41.2
3DN-Conv	Ground Truth	39.3	52.4
3DN-Conv	DORN [10]	33.2	44.3
3DN-Conv	D-DFN	**36.4**	**48.1**

Table 5. RGB-D segmentation results with different source of depth. Segmentation models are trained from scratch.

mantic boundaries and can help with ambiguities. We corroborate this by showing some qualitative results in Figure 6, where the proposed D-DFN returns smoother outputs and finer local depth details.

Depth-aware semantic segmentation Next, we evaluate the performance of using estimated depth for RGB-D segmentation. We report results in Table 5.

We make several observations. First, we confirm the findings that ground truth depth improves semantic segmentation. When using the proposed D-DFN to estimate depth and help semantic segmentation, we improve considerably on top of the standard RGB baseline, coming close to
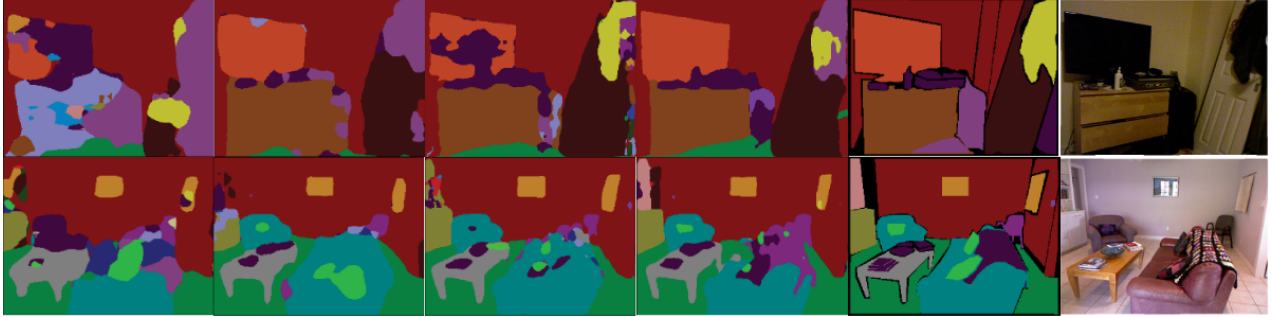
Figure 7. Qualitative results. Columns from left to right: (i) Deeplab V3; (ii) Deeplab V3 + 3DN-Conv; (iii) DFN; (iv) DFN + 3DN-Conv; (v) GT; (vi) image. Deeplab V3 trained from scratch and DFN pretrained on ImageNet.

	Depth at inference time	mIoU(%)	mAcc(%)	Acc(%)
RefineNet [20]	/	46.5	58.9	73.6
FCN [21] (RGB-HHA)	Ground Truth	34.0	46.1	65.4
Qi *et al.* [26]	Ground Truth	43.1	55.7	/
Wang and Neumann [34]	Ground Truth	**48.4**	61.1	/
DFN [38]	/	45.5	58.8	72.7
DFN + 3DN-Conv	Ground Truth	48.2	**61.2**	**74.8**
DFN + 3DN-Conv	Estimated	47.0	59.5	74.3

Table 6. State-of-the-art in RGB-D segmentation with different source of depth. Our 3D neighborhood convolution successfully leverages the estimated depth by the proposed D-DFN depth estimation network to improve RGB-D semantic segmentation, without requiring sensory depth at inference time.

the benefits from using ground truth depth. Last, whereas DORN[10] has a lower global RMS error, it leads to notably lower semantic segmentation accuracies. This confirm our hypothesis that for semantic segmentation it is the local depth structures that are important. We conclude that estimated depth with D-DFN leads to improved semantic segmentation accuracy.

5.4. State-of-the-art comparison

Last, we compare our final model with the state-of-the-art. For the state-of-the-art comparisons we rely on DFN [38] semantic segmentation networks instead of Deeplab V3 [5], as DFN yields empirically the best results. The backbone network is ResNet-101 pre-trained by ImageNet. We report results in Table 6.

We make the following observations. For one, the baseline DFN model is close to the top performing RefineNet [20], when considering only RGB channels as the input. What is more, when relying on estimated depth for helping semantic segmentation, the proposed 3D Neighborhood convolution manages to get very close to models that must rely on ground truth depth, like [34] and ours. This is quite substantial, as the estimated depth is obtained for free with no extra sensory equipment at test time. Note that [34] rely on a RefineNet [20] with an additional pretraining on ADE20K dataset [41] that contains similar indoor scenes,

whereas our models do not require extra pretraining.

We also include qualitative results in Figure 7 to show the effectiveness of our 3DN-Conv. We show that a model with depth incorporated by 3DN-Convs outperforms its baseline model in terms of local edge quality, intra-object consistency and high-level semantic classification.

We conclude that 3D Neighborhood convolution can successfully leverage the estimated depth by the proposed D-DFN depth estimation network to improve RGB semantic segmentation, without sensory depth at inference time.

6. Conclusion

In this work we introduce depth-aware convolutions around 3D neighborhoods, which adapt the receptive field of convolutions according to the local depth. Further, we introduce the D-DFN model for estimating depth maps that are locally accurate around semantic boundaries. As a result, we can now use estimated depth to improve RGB-D based semantic segmentation. Results on the two datasets validate that indeed using estimated depth we improve semantic segmentation considerably. We conclude that convolutions that are aware of depth locality and scale improve RGB-D semantic segmentation, even when estimated depth.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017. 1, 2

[2] B. D. Brabandere, X. Jia, T. Tuytelaars, and L. V. Gool. Dynamic filter networks. In *NIPS*, 2016. 3

[3] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1

[4] 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 transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 1, 2

[5] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 1, 2, 6, 8

[6] H. Chu, W.-C. Ma, K. Kundu, R. Urtasun, and S. Fidler. Surfconv: Bridging 3d and 2d convolution for rgbd images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3002–3011, 2018. 2, 6

[7] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 3

[8] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2650–2658, 2015. 1, 2

[9] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014. 1, 2, 7

[10] H. Fu, M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2002–2011, 2018. 1, 2, 7, 8

[11] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013. 6

[12] S. Gupta, P. Arbelaez, and J. Malik. Perceptual organization and recognition of indoor scenes from rgb-d images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 564–571, 2013. 6

[13] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *European Conference on Computer Vision*, pages 345–360. Springer, 2014. 2

[14] C. Hazirbas, L. Ma, C. Domokos, and D. Cremers. Fusenet: Incorporating depth into semantic segmentation via fusion-based cnn architecture. In *Asian conference on computer vision*, pages 213–228. Springer, 2016. 1, 2

[15] J. Hu, M. Ozay, Y. Zhang, and T. Okatani. Revisiting single image depth estimation: Toward higher resolution maps with accurate object boundaries. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1043–1051. IEEE, 2019. 6

[16] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7132–7141, 2018. 5

[17] J. Jacobsen, J. van Gemert, Z. Lou, and A. Smeulders. Structured receptive fields in cnns. In *CVPR*, 2016. 3

[18] I. Laina, C. Rupprecht, V. Belagiannis, F. Tombari, and N. Navab. Deeper depth prediction with fully convolutional residual networks. In *3D Vision (3DV), 2016 Fourth International Conference on*, pages 239–248. IEEE, 2016. 1, 2, 7

[19] J. Li, R. Klein, and A. Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3372–3380, 2017. 6

[20] G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1925–1934, 2017. 1, 8

[21] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 1, 2, 6, 8

[22] W. Luo, Y. Li, R. Urtasun, and R. Zemel. Understanding the effective receptive field in deep convolutional neural networks. In *Advances in neural information processing systems*, pages 4898–4906, 2016. 2, 3, 4

[23] L. Porzi, A. Penate-Sanchez, E. Ricci, and F. Moreno-Noguer. Depth-aware convolutional neural networks for accurate 3d pose estimation in rgb-d images. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5777–5783. IEEE, 2017. 2

[24] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 1

[25] C. R. Qi, L. Yi, H. Su, and L. J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017. 1

[26] X. Qi, R. Liao, J. Jia, S. Fidler, and R. Urtasun. 3d graph neural networks for rgbd semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5199–5208, 2017. 2, 8

[27] A. Recasens, P. Kellnhofer, S. Stent, W. Matusik, and A. Torralba. Learning to zoom: a saliency-based sampling layer for neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 51–66, 2018. 3

[28] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *Advances in neural information processing systems*, pages 1161–1168, 2006. 2

[29] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions*

on pattern analysis and machine intelligence, 31(5):824–840, 2008. 2

[30] I. Shevlev and S. Avidan. Co-occurrence neural network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[31] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *European Conference on Computer Vision*, pages 746–760. Springer, 2012. 6

[32] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1746–1754, 2017. 1

[33] N. Verma, E. Boyer, and J. Verbeek. FeaStNet: Feature-steered graph convolutions for 3d shape analysis. In *CVPR*, 2018. 3

[34] W. Wang and U. Neumann. Depth-aware cnn for rgb-d segmentation. In *ECCV*, 2018. 2, 3, 4, 6, 7, 8

[35] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018. 3, 6

[36] D. Xu, E. Ricci, W. Ouyang, X. Wang, and N. Sebe. Multi-scale continuous crfs as sequential deep networks for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5354–5362, 2017. 1, 2

[37] P. Xu, F. Davoine, J.-B. Bordes, H. Zhao, and T. Denœux. Multimodal information fusion for urban scene understanding. *Machine Vision and Applications*, 27(3):331–349, 2016. 6

[38] C. Yu, J. Wang, C. Peng, C. Gao, G. Yu, and N. Sang. Learning a discriminative feature network for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1857–1866, 2018. 2, 5, 6, 8

[39] R. Zhang, S. Tang, Y. Zhang, J. Li, and S. Yan. Scale-adaptive convolutions for scene parsing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2031–2039, 2017. 3

[40] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2881–2890, 2017. 1, 2

[41] B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127(3):302–321, 2019. 8

[42] X. Zhu, H. Hu, S. Lin, and J. Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9308–9316, 2019. 3