(DUC, HDC) Understanding Convolution for Semantic Segmentation
(2018)

P wang et al.
Resume

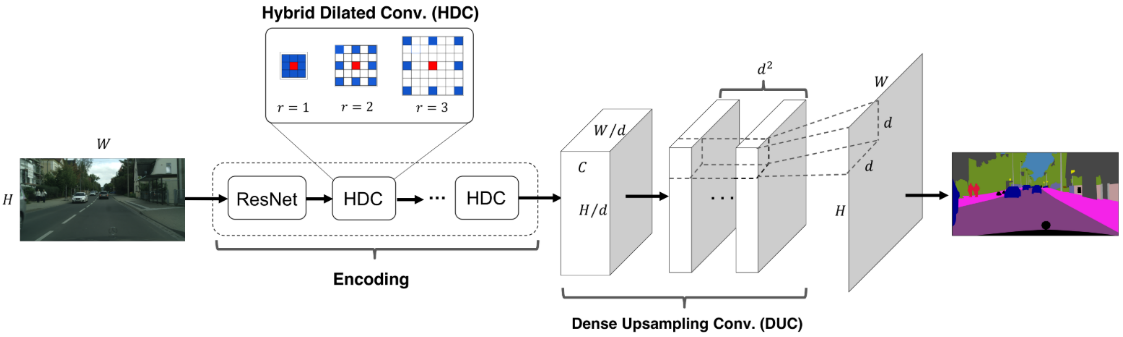
March 12, 2019

1 Introduction

The authors propose a dense upsampling convolution (DUC) to generate pixel-level predictions, which is able to capture and decode more detailed information that is generally missing in bilinear upsampling. and they also propose a hybrid dilated convolution (HDC) framework in the encoding phase. Thus effectively enlarging the receptive field (RF) of the network to aggregate global information and alleviating the gridding issue caused by the stanard dilated convolution operation.

2 Dense upsampling convolution (DUC)

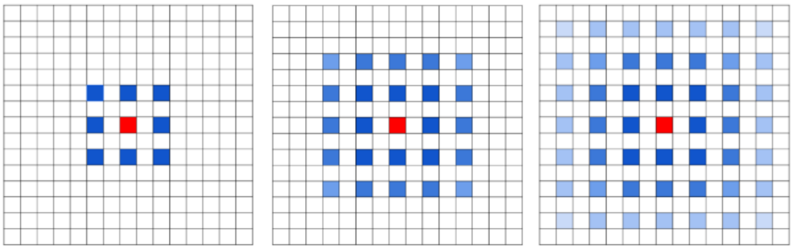
With an input image with height H, width W, and color channels C, and the goal of pixel-level semantic segmentation is to generate a label map with size H W where each pixel is labeled with a category label. After feeding the image into a deep FCN, a feature map with dimension h w c is obtained at the final layer before making predictions, where h = H/d, w = W/d, and d is the downsampling factor. Instead of performing bilinear upsampling, which is not learnable, or using deconvolution network in which zeros have to be padded in the unpooling step before the convolution operation, DUC applies convolutional operations directly on the feature maps to get the dense pixel-wise prediction map.



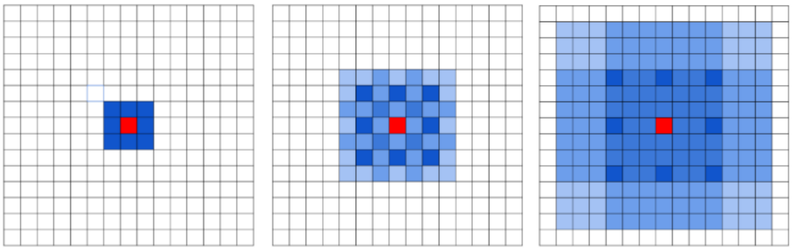
The DUC operation is all about convolution, which is performed on the feature map from ResNet of dimension h w c to get the output feature map of dimension h w (d L), where L is the total number of classes in the semantic segmentation task. Thus each layer of the dense convolution is learning the prediction for each pixel. The output feature map is then reshaped to H W L with a softmax layer, and an element wise argmax operator is applied to get the final label map.

3 Hybrid Dilated Convolution (HDC)

The problem with dilation, is that for a pixel p in a dilated convolutional layer l, the information that contributes to pixel p comes from a nearby $k_d \times k_d$ region in layer $l1$ centered at p. The computation from the $k_d \times k_d$ region are just $k \times k$, with a gap of $r1$ between them. If k = 3, r = 2, only 9 out of 25 pixels in the region are used for the computation.



(a)



(b)

The authors propose to find the correct dilation rates so that each pixel in the top layer, contains information from all the pixels in the receptive field of the bottom layer, like in the second row of the figure above, and this is done by instead of having a fixed dilation for all the layers in a resnet block, they propose to vary the dilations $(1- > 2- > 3)$, this is based on this equation:

$$M_i = \max\left[M_{i+1} - 2r_i, M_{i+1} - 2\left(M_{i+1} - r_i\right), r_i\right]$$

and the objective is to select the correct dilation so that $M_2 \leq K$, where K is the size of the filter, so that in the first convolutions, all the pixels in the receptive field contribute to the output of the convolution.

4 Model

DUC For the DUC part, they only change is the size of the last convolution, if the dimension of the last convolutional layer is 68 68 19 in the baseline model (19 is the number of classes), then the dimension of the same layer for a network with DUC will be 68 68 (r 19) where r is the total downsampling rate of the network (r = 8 in this case). The prediction map is then reshaped to size of the input, 544 544 19. with different abalation studies (different levels of ASSP of deeplab, augmentation and downsampling rates of the resnet, also a cell size, a cell of 1 means Pixel-level DUC, cell 2 means the predictions will be at 1/2 of the size of the original image).

Network	DS	ASPP	Augmentation	Cell	mIoU
Baseline	8	4	*yes*	*n/a*	72.3
Baseline	4	4	*yes*	*n/a*	70.9
DUC	8	*no*	*no*	1	71.9
DUC	8	4	*no*	1	72.8
DUC	8	4	*yes*	1	74.3
DUC	4	4	*yes*	1	73.7
DUC	8	6	*yes*	1	74.5
DUC	8	6	*yes*	2	74.7

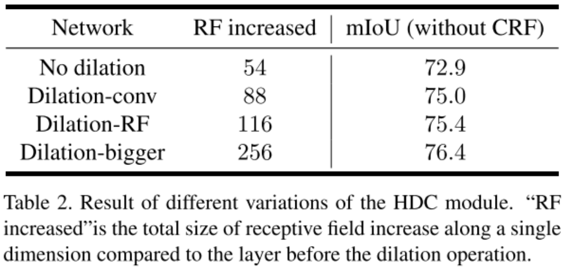
HDC With a 101 ResNet-DUC model as a starting point of applying HDC. they authros experiment with several variants of the HDC module:

1. No dilation: For all ResNet blocks containing dilation, make their dilation rate r = 1 (no dilation).

2. Dilation-conv: For all blocks contain dilation, group every 2 blocks together and make r = 2 for the first block, and r = 1 for the second block.

3. Dilation-RF: For the *res*4*b* module that contains 23 blocks with dilation rate r = 2, group every 3 blocks together and change their dilation rates to be 1, 2, and 3, respectively. For the last two blocks, keep r = 2. For the *res*5*b* module which contains 3 blocks with dilation rate r = 4, we change them to 3, 4, and 5, respectively.

4. Dilation-bigger: For *res*4*b* module, group every 4 blocks together and change their dilation rates to be 1, 2, 5, and 9, respectively. The rates for the last 3 blocks are 1, 2, and 5. For res5b module, set the dilation rates to be 5, 9, and 17.



Network	RF increased	mIoU (without CRF)
No dilation	54	72.9
Dilation-conv	88	75.0
Dilation-RF	116	75.4
Dilation-bigger	256	76.4

Table 2. Result of different variations of the HDC module. "RF increased"is the total size of receptive field increase along a single dimension compared to the layer before the dilation operation.