Designing Convolutional Neural Networks for Urban Scene Understanding

Ye Yuan

CMU-RI-TR-17-06

May 2017

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Alexander G. Hauptmann, Chair
Kris M. Kitani
Xinlei Chen

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

Keywords: Semantic Segmentation, Urban Scene Understanding, Hybrid Dilated Convolution, Dense Upsampling Convolution

To my parents

Abstract

Semantic segmentation is one of the essential and fundamental problems in computer vision community. The task is particularly challenging when it comes to urban street scenes, where the object scales vary significantly. Recent advances in deep learning, especially deep convolutional neural networks (CNNs), have led to significant improvement over previous semantic segmentation systems. In this work, we show how to improve semantic understanding of urban street scenes by manipulating convolution-related operations that are better for practical use. First, we implement dense upsampling convolution (DUC) to generate pixel-level prediction, which is able to capture and decode more detailed information that is generally missing in bilinear upsampling. Second, we propose a hybrid dilated convolution (HDC) framework in the encoding phase. This framework 1) effectively enlarges the receptive fields of the network to aggregate global information; 2) alleviates what we call the gridding issue caused by the standard dilated convolution operation. We evaluate our approaches thoroughly on the Cityscapes dataset, and achieve a new state-of-art result of 80.1% mIOU in the test set. We also are state-of-the-art overall on the KITTI road estimation benchmark and the PASCAL VOC2012 segmentation task.

Acknowledgments

First of all, I would like to thank my advisor Alexander G. Hauptmann for his guidance and support. I am very grateful to have an opportunity working with Informedia Lab. I would like to thank my amazing group mates like Jia Chen, Poyao Huang, Zhenzhong Lan, Han Lu and Shoou-I Yu. I had many ups and downs in research during the last two years, the useful discussions with them really inspire me. I am also extremely grateful to my colleagues in TuSimple, especially Panqu Wang, Pengfei Chen, Ding Liu, Zehua Huang and Xiaodi Hou. I hope you guys all the best in your future endeavor building cutting-edge artificial intelligence applications. Also I would like to thank Haoqi Fan and Shanghang Zhang for their valuable suggestions in this thesis.

Finally, I would like to thank my parents for their unconditional love and trust.

Contents

1 Introduction **1**
1.1 Background . . . 1
1.2 Contributions . . . 1

2 Related Work **5**
2.1 Deeper FCN models . . . 5
2.2 Dilated Convolution . . . 5
2.3 Feature Decoding . . . 5
2.4 Prediction Refinement . . . 6
2.5 Mixed Feature Representation . . . 6

3 Methods **7**
3.1 Overall Architecture . . . 7
3.1.1 Decoding . . . 7
3.1.2 Encoding . . . 7
3.2 Dense Upsampling Convolution (DUC) . . . 8
3.3 Hybrid Dilated Convolution (HDC) . . . 10

4 Evaluation **13**
4.1 Cityscapes Dataset . . . 13
4.1.1 Baseline Model . . . 13
4.1.2 Dense Upsampling Convolution (DUC) . . . 14
4.1.3 Hybrid Dilated Convolution (HDC) . . . 16
4.1.4 Test Set Results . . . 16
4.2 KITTI Road Segmentation . . . 17
4.3 PASCAL VOC2012 . . . 18

5 Conclusion **21**
5.1 Summary . . . 21
5.2 Future Work . . . 21

Bibliography **23**

List of Figures

1.1 Sample images in Cityscapes dataset: the scale of objects varies significantly. . . 2

3.1 Illustration of our overall architecture with ResNet-101 network, Hybrid Dilated Convolution (HDC) and Dense Upsampling Convolution (DUC) layer. 8
3.2 Correspondence between feature map and final score map. Channels omitted for clarity. 9
3.3 Illustration of the gridding problem. Left to right: the pixels (marked in blue) contribute to the calculation of the center pixel (marked in red) through three convolution layers with kernel size 3×3. (a) all convolutional layers have a dilation rate $r = 2$. (b) subsequent convolutional layers have dilation rates of $r = 1, 2, 3$, respectively. 11

4.1 Effect of Dense Upsampling Convolution (DUC) on the Cityscapes validation set. From left to right: input image, ground truth (areas with black color are ignored in evaluation), baseline model, and our ResNet-DUC model. 14
4.2 Effect of Hybrid Dilated Convolution (HDC) on the Cityscapes validation set. From left to right: input image, ground truth, result of the ResNet-DUC model, result of the ResNet-DUC-HDC model. 17
4.3 Effectiveness of HDC in eliminating the gridding effect. First row: ground truth patch. Second row: prediction of the ResNet-DUC model. A strong gridding effect is observed. Third row: prediction of the ResNet-DUC-HDC model. 18
4.4 Examples of visualization on Kitti road segmentation test set. The road is marked in red. 20
4.5 Examples of visualization on the PASCAL VOC2012 segmentation validation set. Left to right: input image, ground truth, our result before CRF, and after CRF. 20

List of Tables

4.1 Ablation studies for applying ResNet-101 on the Cityscapes dataset. **DS**: Down-sampling rate of the network. **Cell**: neighborhood region that one predicted pixel represents. 15
4.2 Result of different variants of the HDC module. 16
4.3 Performance on Cityscapes test set. 19
4.4 Performance on different road scenes in KITTI test set. MaxF: Maximum F1-measure, AP: Average precision. 19
4.5 Performance on the Pascal VOC2012 test set. 19

Chapter 1

Introduction

1.1 Background

Semantic segmentation aims to assign a categorical label to every pixel in an image, which plays an important role in many real world applications, such as self-driving vehicles. The task is particularly challenging when it comes to urban street scenes, where the object scales vary significantly. Figure 1.1 shows the scale variation of objects in street.

The recent success of deep convolutional neural network (CNN) models [16, 20, 29] has enabled remarkable progress in pixel-wise semantic segmentation tasks due to rich hierarchical features and an end-to-end trainable framework [5, 18, 21, 24, 25, 35, 38]. Since the introduction of FCN in [25], improvements on fully-supervised semantic segmentation systems are generally focused on following perspectives: (1) a fully-convolutional network (FCN), first introduced in [25], replacing the last few fully connected layers by convolutional layers to make efficient end-to-end learning and inference that can take arbitrary input size; (2) Conditional Random Fields (CRFs), to capture both local and long-range dependencies within an image to refine the prediction map; (3) dilated convolution (or Atrous convolution), which is used to increase the resolution of intermediate feature maps in order to generate more accurate predictions while maintaining the same computational cost; (4) combining features from different stages of a CNN to get more precise dense prediction.

1.2 Contributions

We are pursuing further improvements on semantic segmentation especially for urban scene understanding from another perspective: the *convolutional* operations for both decoding (from intermediate feature map to output label map) and encoding (from input image to feature map) counterparts. We design DUC and HDC to make *convolution* operations better serve the need of pixel-level semantic segmentation. The technical details are described in Chapter 3. We show that our approaches achieve a new state-of-the-art result of 80.1% mIOU in the Cityscapes pixel-level semantic labeling task. We also are state-of-the-art overall on the KITTI road estimation benchmark and the PASCAL VOC2012 segmentation task.

In decoding (prediction), most state-of-the-art semantic segmentation systems simply use

(a) Zürich

(b) Stuttgart

Figure 1.1: Sample images in Cityscapes dataset: the scale of objects varies significantly.

bilinear upsampling (before the post-processing stage such as CRF) to get the output label map [5, 21, 24]. Bilinear upsampling is not learnable and may lose fine details. We propose a method called *dense upsampling convolution (DUC)*, which is easy to implement and can achieve pixel-level accuracy: instead of trying to recover the full-resolution label map at once, we learn an array of upscaling filters to upscale the downsized feature maps into the final dense feature map of the desired size. DUC naturally fits the FCN framework by enabling end-to-end training, and it increases the mIOU of pixel-level semantic segmentation on the Cityscapes dataset [7] significantly, especially on objects that are relatively small.

For the encoding part (feature learning), dilated convolution recently became popular [5, 33, 35, 39], as it maintains the resolution and receptive field of the network by in inserting "holes"in the convolution kernels, thus eliminating the need for downsampling (by max-pooling or strided convolution). However, an inherent problem exists in the current dilated convolution framework,

which we identify as "gridding": as zeros are padded between two pixels in a convolutional kernel, the receptive field of this kernel only covers an area with checkerboard patterns - only locations with non-zero values are sampled, losing some neighboring information. The problem gets worse when the rate of dilation increases, generally in higher layers when the receptive field is large: the convolution kernel is too sparse to capture any local information, since the non-zero values are too far apart. Information that contributes to a fixed pixel always comes from its predefined gridding pattern, thus losing a huge portion of information. Here we propose a simple *hybrid dilation convolution (HDC)* framework as a first attempt to address this problem: instead of using the same rate of dilation for the same spatial resolution, we use a range of dilation rates and concatenate them serially the same way as "blocks"in ResNet-101 [16]. We show that HDC helps the network to alleviate the gridding problem. Moreover, choosing proper rates can effectively increases the receptive field size and improves the accuracy for objects that are relatively big.

Chapter 2

Related Work

2.1 Deeper FCN models

Significant gains in mean Intersection-over-Union (mIoU) scores on PASCAL VOC2012 dataset [11] were reported when the 16-layer VGG-16 model [29] was replaced by a 101-layer ResNet-101 [16] model [5]; using 152 layer ResNet-152 model yields further improvements [33]. This trend is consistent with the performance of these models on ILSVRC [27] object classification tasks, as deeper networks generally can model more complex representations and learn more discriminative features that better distinguish among categories.

2.2 Dilated Convolution

Dilated Convolution (or Atrous convolution) was originally developed in *algorithme à trous* for wavelet decomposition [17]. The main idea of dilated convolution is to insert "holes"(zeros) between pixels in convolutional kernels to increase image resolution, thus enabling dense feature extraction in deep CNNs. In the semantic segmentation framework, dilated convolution is also used to enlarge the field of convolutional kernels. Yu & Koltun [35] use serialized layers with increasing rates of dilation to enable context aggregation, while [5] design an "atrous spatial pyramid pooling (ASPP)"scheme to capture multi-scale objects and context information by placing multiple dilated convolution layers in parallel. More recently, dilated convolution has been applied to a broader range of tasks, such as object detection [8], optical flow [28], and audio generation [32].

2.3 Feature Decoding

In the pixel-wise semantic segmentation task, the output label map has the same size as the input image. Because of the operation of max-pooling or strided convolution in CNNs, the size of feature maps of the last few layers of the network are inevitably downsampled. Multiple approaches have been proposed to decode accurate information from the downsampled feature map to label maps. Bilinear interpolation is commonly used [5, 21, 24], as it is fast and memory-efficient.

Another popular method is called deconvolution, in which the unpooling operation, using stored pooling switches from the pooling step, recovers the information necessary for feature visualization [36]. In [25], a single deconvolutional layer is added in the decoding stage to produce the prediction result using stacked feature maps from intermediate layers. In [10], multiple deconvolutional layers are applied to generate chairs, tables, or cars from several attributes. Noh et al. [26] employ deconvolutional layers as mirrored version of convolutional layers by using stored pooled location in unpooling step. [26] show that coarse-to-fine object structures, which are crucial to recover fine-detailed information, can be reconstructed along the propagation of the deconvolutional layers. Fischer at al. [12] use a similar mirrored structure, but combine information from multiple deconvolutional layers and perform upsampling to make the final prediction.

2.4 Prediction Refinement

People use Conditional Random Fields (CRFs) to capture both local and long-range dependencies within an image to refine the prediction map. This includes applying fully connected pairwise CRFs [19] as a post-processing step [5], integrating CRFs into the network by approximating its mean-field inference steps [21, 24, 38] to enable end-to-end training, and incorporating additional information into CRFs such as edges [18] and object detections [2].

2.5 Mixed Feature Representation

Some researchers also explored various architecture to make use of features from different stages of CNN. The original FCN [25] models make predictions based on the combination of features from both early layers and subsequent layers. [30] proposed a module, called mixed context network to make use of features from different layers. The recent PixelNet [3] also suggested to use multiscale convolutional features as an intermediate representation for general pixel level prediction problems.

Chapter 3

Methods

3.1 Overall Architecture

As we mentioned before, the main challenge for semantic segmentation in urban street scenes is that the scale of objects can vary tremendously. Solving this task requires both object-level information and pixel-level accuracy. Thus, we argue that an ideal convolutional neural network designed for dense pixel-labeling in street scenes should be able to: (1) capture high-level global context and (2) identify fine-grained local structure. A FCN style model could be divided into two parts: an encoder to extract semantic features from raw image, and a decoder to get category predictions for each pixel. We will discuss the design decisions for each part that consider both receptive field and feature resolution. The main idea is: (1) In encoding phase we try to balance receptive field and resolution by carefully using dilated convolution; (2) In decoding phase we use simple but effective way to make predictions.

3.1.1 Decoding

In decoding part, most state-of-the-art segmentation systems like DeepLab [5] usually produces a coarse score map (corresponding to log-probabilities) first, then use a simple bilinear interpolation to produce final full resolution score map. In this way, some fine details in the image will not be captured since the network is also trained with coarse ground truth label (downsampled by a factor of 8 in DeepLab). This is particularly a problem in urban street scenes, where some small objects like pole and pedestrians suffer from inaccurate coarse-grained predictions. To address this problem, we propose a method called *dense upsampling convolution* (DUC) to better decode feature map into dense pixel-level predictions.

3.1.2 Encoding

The encoder part is usually based on networks pretrained on ImageNet [27] classification task since pixel-level annotations for segmentation task is very expensive and usually not enough for training deep networks.

We choose ResNet-101[16] as our default base network for three reasons: (1) although recently some new architectures[31, 34] claim to be better than ResNet, it's still the state-of-art

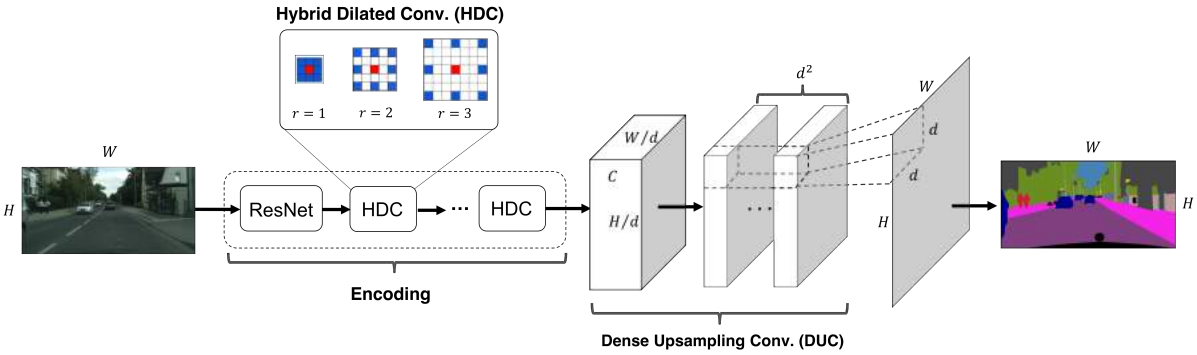

Figure 3.1: Illustration of our overall architecture with ResNet-101 network, Hybrid Dilated Convolution (HDC) and Dense Upsampling Convolution (DUC) layer.

CNN model for visual recognition. (2) deeper model has larger effective receptive field, which is one of the key factors in scene understanding; (3) the residual design in ResNet has similar effect of fusing multi-stage feature, which proves to be useful in dense pixel-level prediction task [3, 25].

We also use dilated convolution to enlarge receptive field. Due to the constraints of computing resource, traditional CNN models designed for classification or detection tasks usually adopt pooling or strided convolution operators to reduce feature map resolution. Theoretically, one can arbitrarily enlarge the receptive field by using very large sampling rate in dilated convolution. However, it's not clear how to choose an appropriate dilation rate for each layer. Previous works [5, 35] use dilation rate 2, 4, 8, 16 in multiple layers to achieve the same receptive field with original network. We argue that it's a suboptimal way and propose *hybrid dilation convolution* (HDC) framework as a first attempt to address this problem, details will be covered in following section. Figure 3.1 depicts the architecture of our overall architecture with ResNet-101 network, Hybrid Dilated Convolution (HDC) and Dense Upsampling Convolution (DUC) layer.

3.2 Dense Upsampling Convolution (DUC)

Given an input image X, the goal of pixel-level semantic segmentation is to generate outputs Y where each pixel is assigned a category label from $\{1, 2, ..., L\}$ (L is the total number of classes). Suppose the image has width W and height H, after feeding the image into a deep FCN, a feature map $f_{enc}(X)$ (f_{enc} is the encoder mentioned before) with dimension $h \times w \times c$ 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 deconvolution, 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 $f_{enc}(X)$ from ResNet of dimension $h \times w \times c$ to get the output $f_{dec}(f_{enc}(X))$ of dimension $h \times w \times (d^2 \times L)$. Each feature from $f_{dec}(f_{enc}(X))$ produces the pre-activation score map for a neighboring region of size $d \times d$. Thus the dense convolution is learning the prediction for each pixel. The output feature map

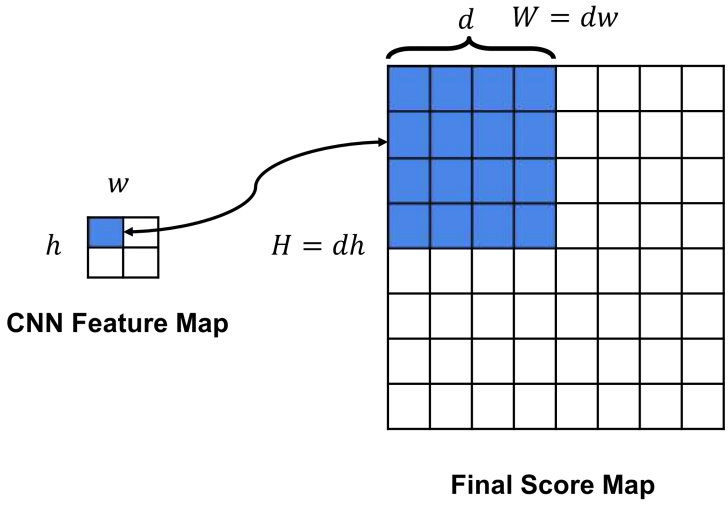

Figure 3.2: Correspondence between feature map and final score map. Channels omitted for clarity.

is then reshaped to $H \times W \times L$ with a softmax layer, and an element-wise *argmax* operator is applied to get the final label map. In practice, the "reshape" operation is not necessary, as the feature map can be flatten to a vector to be fed into the softmax layer. Figure 3.1 illustrates the correspondence between feature map and final score map. The final category label for a pixel in location (i, j) ($i = 1, ..., W$, $j = 1, ..., H$) will be (To be clear, we adopt a MATLAB style notation for matrix indexing) :

$$Y[i, j] = \underset{\text{channel}}{\operatorname{argmax}} f_\theta(X)\left[\left\lceil \frac{i}{d} \right\rceil, \left\lceil \frac{j}{d} \right\rceil, kL : (k+1)L\right] \tag{3.1}$$

where $f_\theta(X)$ is the score map we get after applying softmax activation on feature map $f_{dec}(f_{enc}(X))$. Index $k = d \cdot j \pmod d - d + i \pmod d$ finds the location of softmax scores in feature channel.

The key idea of DUC is to divide the raw image into equal $d \times d$ sub-regions, and the corresponding feature in $f_{enc}(X)$ is responsible to predict the label of each pixel in this region. We are able to achieve this because the feature we obtained from CNN has encoded information of a large neighboring region in raw image. The region is usually named "receptive field", refers to the part of the image that is visible to one filter at a time. The receptive field increases as we stack more convolutional layers. There are two major advantages of DUC:

1. It produces accurate dense pixel-level predictions. Since DUC is learnable, it's capable of capturing and recovering fine-detailed information that is generally missing in the bilinear interpolation operation. For example, if a network has a downsample rate of $1/16$, and an object has a length or width less than 16 pixels (such as a pole or a person far away), then it is more than likely that bilinear upsampling will not be able to recover this object. Meanwhile, the corresponding training labels have to be downsampled to correspond with the output dimension, which will already cause information loss for fine details. The prediction of DUC, on the other hand, is performed at the original resolution, thus enabling

pixel-level decoding. In addition, the DUC operation can be naturally integrated into the FCN framework, and makes the whole encoding and decoding process end-to-end trainable.

2. It is more efficient in computation and storage. DUC allows us to apply the convolution operation directly between the input feature map and the output label maps without the need of inserting extra values in deconvolutional layers (the "unpooling"operation). Thus it avoids some overheads storing internal feature representations.

3.3 Hybrid Dilated Convolution (HDC)

In 1-D, dilated convolution is defined as:

$$g[i] = \sum_{l=1}^{L} f[i + r \cdot l]h[l], \tag{3.2}$$

where $f[i]$ is the input signal, $g[i]$ is the output signal , $h[l]$ denotes the filter of length L, and r corresponds to the dilation rate we use to sample $f[i]$. In standard convolution, $r = 1$.

In a semantic segmentation system, 2-D dilated convolution is constructed by inserting "holes" (zeros) between each pixel in the convolution kernel. For a convolution kernel with size $k \times k$, the size of resulting dilated filter is $k_d \times k_d$, where $k_d = k + (k-1) \cdot (r-1)$. Dilated convolution is used to maintain high resolution of feature maps in FCN through replacing the max-pooling operation or strided convolution layer while keeping the same receptive field (or "field of view"in [5]) of the corresponding layer. For example, if a convolution layer in ResNet-101 has a stride $s = 2$, then the stride is reset to 1 to remove downsampling, and the dilation rate r is set to 2 for all convolution kernels of subsequent layers. This process is applied iteratively through all layers that have a downsampling operation, thus the feature map in the output layer can maintain the same resolution as the input layer. In practice, however, dilated convolution is generally applied on feature maps that are already downsampled to achieve a reasonable efficiency/accuracy trade-off [5].

Although dilated convolution allows us to achieve a high resolution of feature map, we argue that a lot of fine details of image are still lost by adopting a same dilation rate in consecutive convolutional layers. For a feature pixel p in a dilated convolutional layer l, the information that contributes to it comes from a nearby $k_d \times k_d$ region in layer $l-1$ centered at p. Since dilated convolution introduces zeros in the convolutional kernel, the actual number of pixels that participate in the computation from the $k_d \times k_d$ region is just $k \times k$. If $k = 3, r = 2$, only 9 out of 25 pixels in the region are used for the computation (Figure 3.3 (a)). Since all layers have equal dilation rates r, then the receptive field for pixel p in the top dilated convolution layer l_{top} is:

$$\text{RF} = [2r(l_{top} - l_{bottom} + 1) + 1] \times [2r(l_{top} - l_{bottom} + 1) + 1] \tag{3.3}$$

where $l_{top} - l_{bottom} + 1$ is the number of layers. However, the number of pixels that p can see is only:

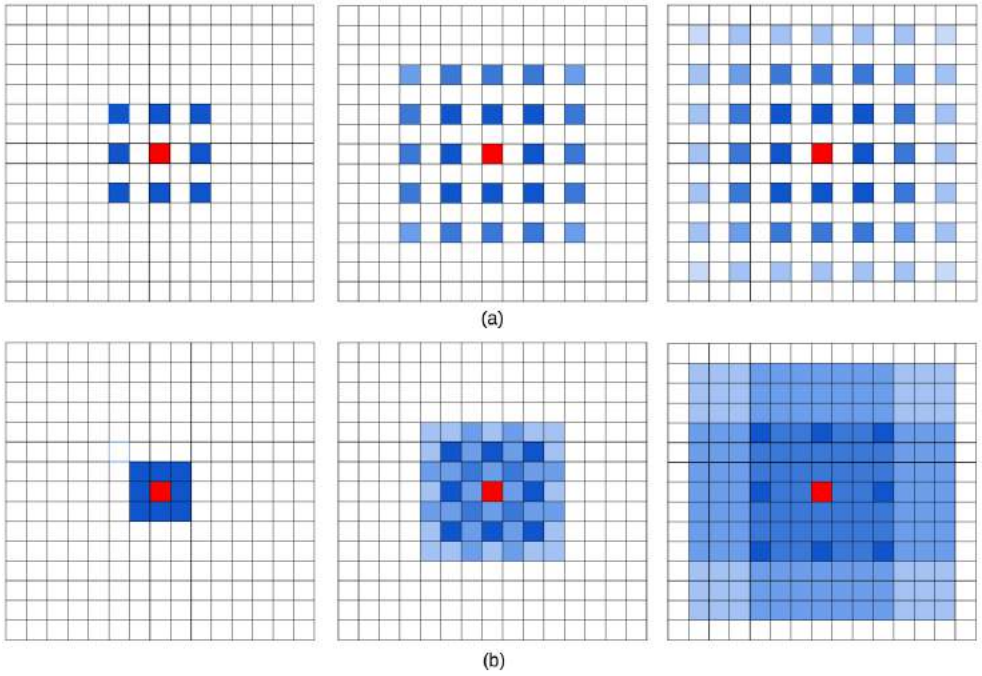

Figure 3.3: Illustration of the gridding problem. Left to right: the pixels (marked in blue) contribute to the calculation of the center pixel (marked in red) through three convolution layers with kernel size 3×3. (a) all convolutional layers have a dilation rate $r = 2$. (b) subsequent convolutional layers have dilation rates of $r = 1, 2, 3$, respectively.

$$\text{pixels} = RF \times \left[\frac{r(l_{top} - l_{bottom} + 1) + 1}{2r(l_{top} - l_{bottom} + 1) + 1} \right]^2 \tag{3.4}$$

As a result, pixel p can only view information in a checkerboard fashion, and lose a large portion (at least 75% when $r = 2$) of information, which we refer it as the "**gridding**"problem. When r becomes large in higher layers due to additional downsampling operations, the sample from the input can be very sparse, which may not be good for learning because (1) local information is completely missing; (2) the information can be irrelevant across large distances. Another outcome of the gridding effect is that pixels in nearby $r \times r$ regions at layer l receive information from completely different set of "grids" which may impair the consistency of local information.

Here we propose a simple solution- *hybrid dilated convolution (HDC)*, to address this theoretical issue. Instead of using the same dilation rate for all layers after the downsampling occurs, we use a different dilation rate for each layer. The assignment of dilation rate follows a sawtooth wave-like fashion: a number of layers are grouped together to form the "rising edge"of the wave that has an increasing dilation rate, and the next group repeats the same pattern. For example, for all layers that have dilation rate $r = 2$, we form 3 succeeding layers as a group, and change their dilation rates to be 1, 2, and 3, respectively. By doing this, the top layer can access information from a broader range of pixels, in the same region as the original configuration (Figure 3.3 (b)). This process is repeated through all layers, thus making the receptive field unchanged at the top layer.

Another benefit of HDC is that it can use arbitrary dilation rates through the process, thus naturally enlarging the receptive fields of the network without adding extra modules [35], which is important for recognizing objects that are relatively big. One important thing to note, however, is that the dilation rate within a group should not have a common factor relationship (like 2,4,8, etc.), otherwise the gridding problem will still hold for the top layer. This is the key difference between our HDC approach and the atrous spatial pyramid pooling (ASPP) module in [5], or the context aggregation module in [35], where dilation factors that have common factor relationships are used. In addition, HDC is naturally integrated with the original layers of the network, without any need to add extra modules as in [5, 35].

Chapter 4

Evaluation

We report our experiments and results on three challenging semantic segmentation datasets: Cityscapes [7], KITTI dataset [13] for road estimation, and PASCAL VOC2012 [11]. We use ResNet-101 or ResNet-152 networks that have been pretrained on the ImageNet dataset as a starting point for all of our models. The output layer contains the number of semantic categories to be classified depending on the dataset (including background, if applicable). We use the cross-entropy error at each pixel over the categories. This is then summed over all pixel locations of the output map, and we optimize this objective function using standard Stochastic Gradient Descent (SGD). We use MXNet [6] to train and evaluate all of our models on NVIDIA TITAN X GPUs.

4.1 Cityscapes Dataset

The Cityscapes Dataset is a large dataset that focuses on semantic understanding of urban street scenes. The dataset contains 5000 images with fine annotations across 50 cities, different seasons, varying scene layout and background. The dataset is annotated with 30 categories, of which 19 categories are included for training and evaluation (others are ignored). The training, validation, and test set contains 2975, 500, and 1525 fine images, respectively. An additional 20000 images with coarse (polygonal) annotations are also provided, but are only used for training.

4.1.1 Baseline Model

We use the DeepLab-V2 [5] ResNet-101 framework to train our baseline model. Specifically, the network has a downsampling rate of 8, and dilated convolution with rate of 2 and 4 are applied to *res4b* and *res5b* blocks, respectively. An ASPP module with dilation rate of 6, 12, 18, and 24 is added on top of the network to extract multiscale context information. The prediction maps and training labels are downsampled by a factor of 8 compared to the size of original images, and bilinear upsampling is used to get the final prediction. Since the image size in the Cityscapes dataset is 1024×2048, which is too big to fit in the GPU memory, we partition each image into twelve 800×800 patches with partial overlapping, thus augmenting the training set to have 35700 images. This data augmentation strategy is to make sure all regions in an image can be visited. This is an improvement over random cropping, in which nearby regions may be visited repeatedly.

We train the network using mini-batch SGD with patch size 544×544 (randomly cropped from the 800×800 patch) and batch size 12, using multiple GPUs. The initial learning rate is set to 2.5×10^{-4}, and a "*poly*"learning rate (as in [5]) with $power = 0.9$ is applied. Weight decay is set to 5×10^{-4}, and momentum is 0.9. The network is trained for 20 epochs and achieves mIoU of 72.3% on the validation set.

4.1.2 Dense Upsampling Convolution (DUC)

We will examine the effect of DUC on the baseline network first since it can be applied to any semantic segmentation framework. In DUC the only thing we change is the shape of the top convolutional layer. For example, if the dimension of the top convolutional layer is $68 \times 68 \times 19$ in the baseline model (19 is the number of classes), then the dimension of the same layer for a network with DUC will be $68 \times 68 \times (r^2 \times 19)$ where r is the total downsampling rate of the network ($r = 8$ in this case). The prediction map is then reshaped to size $544 \times 544 \times 19$. DUC will introduce extra parameters compared to the baseline model, but only at the top convolutional layer. We train the ResNet-DUC network the same way as the baseline model for 20 epochs, and achieve a mean IOU of 74.3% on the validation set, a 2% increase compared to the baseline model. Visualization of the result of ResNet-DUC and comparison with the baseline model is shown in Figure 4.1

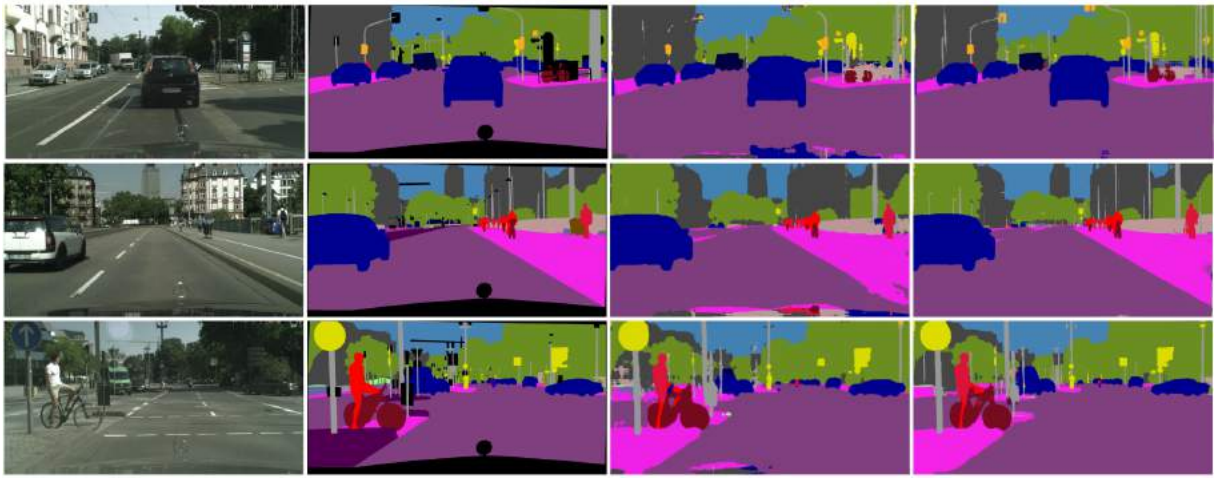

Figure 4.1: Effect of Dense Upsampling Convolution (DUC) on the Cityscapes validation set. From left to right: input image, ground truth (areas with black color are ignored in evaluation), baseline model, and our ResNet-DUC model.

From Figure 4.1, we can clearly see that DUC is very helpful for identifying small objects, such as poles, traffic lights, and traffic signs. Consistent with our intuition, pixel-level dense upsampling can recover detailed information that is generally missed by bilinear interpolation.

Ablation Studies We examine the effect of different settings of the network on the performance. Specifically, we examine: 1) the downsampling rate of the network, which controls the resolution of the intermediate feature map; 2) whether to apply the ASPP module, and the number of parallel paths in the module; 3) whether to perform 12-fold data augmentation; and 4) cell size, which determines the size of neighborhood region ($cell \times cell$) that one predicted pixel

projects to. Pixel-level DUC should use $cell = 1$; however, since the ground-truth label generally cannot reach pixel-level precision, we also try $cell = 2$ in the experiments. From Table 4.1 we can see that making the downsampling rate smaller decreases the accuracy. Also it significantly raises the computational cost due to the increasing resolution of the feature maps. ASPP generally helps to improve the performance, and increasing ASPP channels from 4 to 6 (dilation rate 6 to 36 with interval 6) yields a 0.2% boost. Data augmentation helps to achieve another 1.5% improvement. Using $cell = 2$ yields slightly better performance when compared with $cell = 1$, and it helps to reduce computational cost by decreasing the channels of the last convolutional layer by a factor of 4.

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

Table 4.1: Ablation studies for applying ResNet-101 on the Cityscapes dataset. **DS**: Downsampling rate of the network. **Cell**: neighborhood region that one predicted pixel represents.

Bigger Patch Size Since setting $cell = 2$ reduces GPU memory cost for network training, we explore the effect of patch size on the performance. Our assumption is that, since the original images are all 1024×2048, the network should be trained using patches as big as possible in order to aggregate both local detail and global context information that may help learning. As such, we make the patch size to be 880×880, and set the batch size to be 1 on each of the 4 GPUs used in training. Since the patch size exceeds the maximum dimension (800×800) in the previous 12-fold data augmentation framework, we adopt a new 7-fold data augmentation strategy: seven center locations with $x = 512$, $y = \{256, 512, ..., 1792\}$ are set in the original image; for each center location, a 880×880 patch is obtained by randomly setting its center within a 160×160 rectangle area centered at each center. This strategy makes sure that we can sample all areas in the image, including edges. Training with a bigger patch size boosts the performance to **75.7%**, a 1% improvement over the previous best result.

Compared with Deconvolution We compare our DUC model with deconvolution, which also involves learning for upsampling. Particularly, we compare with 1) direct deconvolution from the prediction map (dowsampled by 8) to the original resolution; 2) deconvolution with an upsampling factor of 2 first, followed by deconvolution with an upsampling factor of 4. We use the ResNet-DUC bigger patch model to train the networks. The above two models achieve mIOU of 75.1% and 75.0%, respectively, lower than the ResNet-DUC model (75.7% mIoU).

Conditional Random Fields (CRFs) Fully-connected CRFs [19] are widely used for improving semantic segmentation quality as a post-processing step of an FCN [5].

We follow the formation of CRFs as shown in [5]. We perform a grid search on parameters on the validation set, and use $\sigma_\alpha = 15$, $\sigma_\beta = 3$, $\sigma_\gamma = 1$, $w_1 = 3$, and $w_2 = 3$ for all of our models. Applying CRFs to our best ResNet-DUC model yields an mIoU of **76.7%**, a 1% improvement over the model does not use CRFs.

4.1.3 Hybrid Dilated Convolution (HDC)

We use the best 101 layer ResNet-DUC model as a starting point of applying HDC. Specifically, we experiment with several configurations of the HDC module:

1. No dilation: For all ResNet blocks containing dilation, we make their dilation rate $r = 1$ (no dilation).
2. Dilation-DeepLab: For $res4b$, we keep dilation rate $r = 2$. For $res5b$, we use $r = 5$. These follows the settings in DeepLab[16].
3. Dilation-HDC-var1: For all blocks contain dilation, we group every 2 blocks together and make $r = 2$ for the first block, and $r = 1$ for the second block.
4. Dilation-HDC-var2: For the $res4b$ module that contains 23 blocks with dilation rate $r = 2$, we group every 3 blocks together and change their dilation rates to be 1, 2, and 3, respectively. For the last two blocks, we keep $r = 2$. For the $res5b$ module which contains 3 blocks with dilation rate $r = 4$, we change them to 3, 4, and 5, respectively.
5. Dilation-HDC-var3: For $res4b$ module, we 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, we set the dilation rates to be 5, 9, and 17.

The result is summarized in Table 4.2. We can see that increasing receptive field size generally yields higher accuracy. Figure 4.3 illustrates the effectiveness of the ResNet-DUC-HDC model in eliminating the gridding effect. A visualization result is shown in Figure 4.2. We can see our best ResNet-DUC-HDC model performs particularly well on objects that are relatively big.

Network	Dilation Assignments	mIoU
No dilation	r=1	72.9
Dilation-DeepLab	res4b, r=2; res5b, r=4	74.7
Dilation-HDC-var1	r=2,1	75.0
Dilation-HDC-var2	res4b, r=1,2,3; res5b, r=3,4,5	75.4
Dilation-HDC-var3	res4b, r=1,2,5,9; res5b, r=5,9,17	76.4

Table 4.2: Result of different variants of the HDC module.

4.1.4 Test Set Results

To make the most of the models we have trained so far, we create an ensemble model to further improve the expressiveness of our framework. Specifically, we use CRF-processed ResNet-152-Deconv model, ResNet-101 HDC+DUC model, and ResNet-152 HDC+DUC model. For a given image, we first stack the label map of all models as a multi-channel representation. We

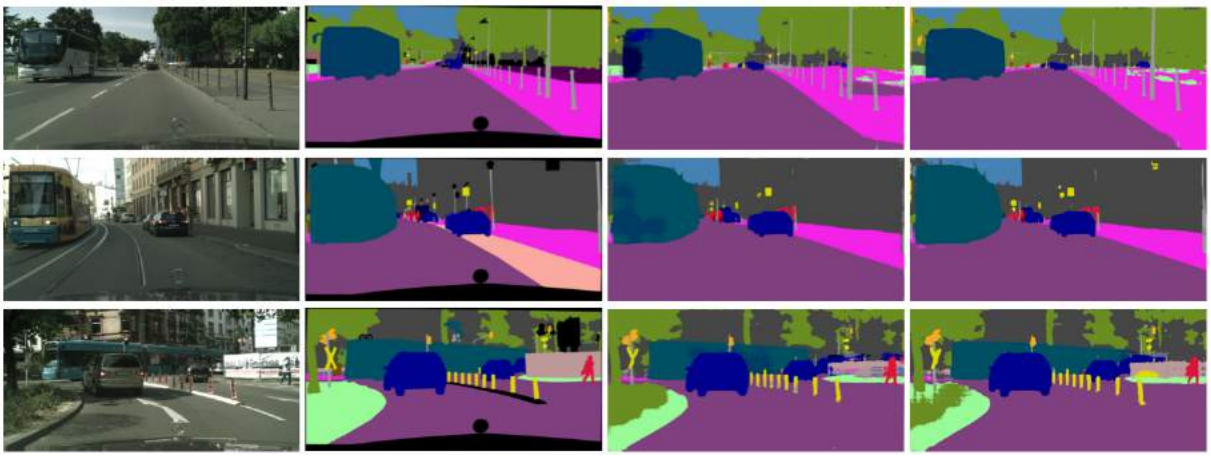

Figure 4.2: Effect of Hybrid Dilated Convolution (HDC) on the Cityscapes validation set. From left to right: input image, ground truth, result of the ResNet-DUC model, result of the ResNet-DUC-HDC model.

then use the SLIC algorithm (50000 superpixels per example, color ratio $M = 0$)[1] to gather the superpixels of the example, and merge nearby superpixels that have the same label. Using the merged superpixels, we traverse all possible label combinations ("rules") of all channels on selected categories that lead to performance improvement on a subset of images from validation set, and only adopt the rules that generalize well on the other images in validation set to avoid overfitting. We then apply these rules on the test set to get the final prediction result.

Our results are summarized in Table 4.3. There are separate entries for models trained using fine-labels only, and using a combination of fine and coarse labels. Our single ResNet-DUC-HDC model achieves 76.1% mIoU using fine data only, and the ensemble method boosts the performance to 77.6%. Adding coarse data help us achieve **78.5%** mIoU.

In addition, inspired by the design of the VGG network [29], in that a single 5×5 convolutional layer can be decomposed into two adjacent 3×3 convolutional layers to increase the expressiveness of the network while maintaining the receptive field size, we replaced the 7×7 convolutional layer in the original ResNet-101 network by three 3×3 convolutional layers. By retraining the updated network, we achieve a mIoU of **80.1%** on the test set using a single model with multiscale testing, without CRF post-processing, or model ensemble procedure described above. Our result achieves the state-of-the-art performance on the Cityscapes dataset. Compared with the strong baseline of Chen et al. [5], we improve the mIoU by a significant margin (9.7%), which demonstrates the effectiveness of our approach.

4.2 KITTI Road Segmentation

The KITTI road segmentation task contains images of three various categories of road scenes, including 289 training images and 290 test images. The goal is to decide if each pixel in images is road or not. It is challenging to use neural network based methods due to the limited number of training images. In order to avoid overfitting, we crop patches of 320×320 pixels with a stride of 100 pixels from the training images, and use the ResNet-101-DUC model pretrained

Figure 4.3: Effectiveness of HDC in eliminating the gridding effect. First row: ground truth patch. Second row: prediction of the ResNet-DUC model. A strong gridding effect is observed. Third row: prediction of the ResNet-DUC-HDC model.

from ImageNet during training. Other training settings are the same as Cityscapes experiment. We did not apply CRFs for post-processing.

Results We achieve the state-of-the-art results without using any additional information of stereo, laser points and GPS. Specifically, our model attains the highest maximum F1-measure in the sub-categories of urban unmarked (UU_ROAD), urban multiple marked (UMM_ROAD) and the overall category URBAN_ROAD of all sub-categories, the highest average precision across all three sub-categories and the overall category by the time of submission of this paper. Examples of visualization results are shown in Figure 4.4. The detailed results are displayed in Table 4.4 [1].

4.3 PASCAL VOC2012

The PASCAL VOC2012 segmentation benchmark contains 1464 training images, 1449 validation images, and 1456 test images. Using the extra annotations provided by [15], the training set is augmented to have 10582 images. The dataset has 20 foreground object categories and 1 background class with pixel-level annotation.

Results We first pretrain our 152 layer ResNet-DUC model using a combination of augmented VOC2012 training set and MS-COCO dataset [23], and then finetune the pretrained

[1]For thorough comparison with other methods, please check http://www.cvlibs.net/datasets/kitti/eval_road.php.

Method	mIoU
fine	
FCN 8s [25]	65.3
Dilation10 [35]	67.1
DeepLabv2-CRF [5]	70.4
Adelaide_context [21]	71.6
RefineNet [22]	73.6
ResNet-DUC-HDC-single (ours)	**76.1**
fine + *coarse*	
LRR-4x [14]	71.8
PSPNet [37]	80.2
ResNet-DUC-HDC-msc (ours)	**80.1**

Table 4.3: Performance on Cityscapes test set.

	MaxF	AP
UM_ROAD	95.64%	93.50%
UMM_ROAD	97.62%	95.53%
UM_ROAD	95.17%	92.73%
URBAN_ROAD	96.41%	93.88%

Table 4.4: Performance on different road scenes in KITTI test set. MaxF: Maximum F1-measure, AP: Average precision.

network using augmented VOC2012 trainval set. We use patch size 512×512 (zero-padded) throughout training. All other training strategies are the same as Cityscapes experiment. We apply CRF as a postprocessing step. We achieve mIOU of **83.1%** on the test set using a single model without any model ensemble or multiscale testing, which achieves state-of-the-art result. [2]. The detailed results are displayed in Table 4.5, and visualizations of the results are shown in Figure 4.5.

Method	mIoU
DeepLabv2-CRF[5]	79.7
CentraleSupelec Deep G-CRF[4]	80.2
FCN_MCN [30]	80.6
ResNet-DUC-CRF (ours)	**83.1**

Table 4.5: Performance on the Pascal VOC2012 test set.

[2]Result link: http://host.robots.ox.ac.uk:8080/anonymous/LQ2ACW.html

Figure 4.4: Examples of visualization on Kitti road segmentation test set. The road is marked in red.

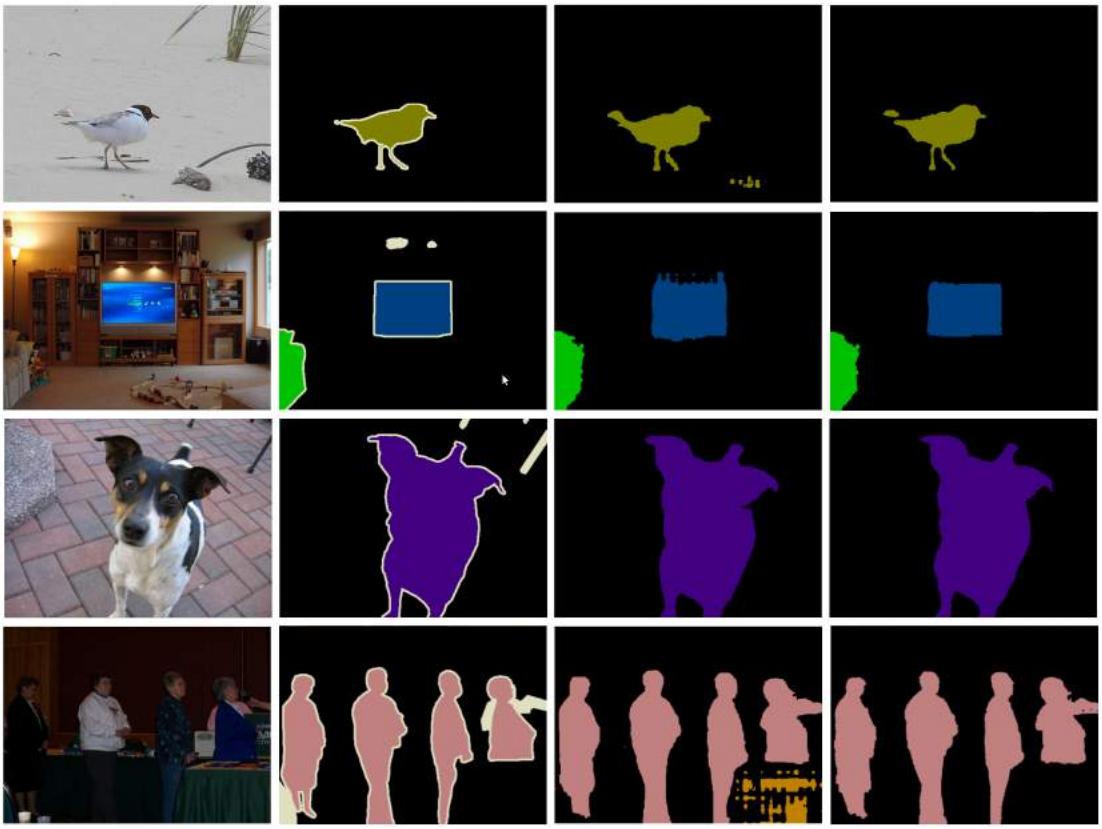

Figure 4.5: Examples of visualization on the PASCAL VOC2012 segmentation validation set. Left to right: input image, ground truth, our result before CRF, and after CRF.

Chapter 5

Conclusion

5.1 Summary

We propose simple yet effective convolutional operations for improving semantic segmentation systems. We designed a new dense upsampling convolution (DUC) operation to enable pixel-level prediction on feature maps, and hybrid dilated convolution (HDC) to deal with the gridding problem, effectively enlarging the receptive fields of the network. Experimental results demonstrate the effectiveness of our framework on various semantic segmentation tasks.

5.2 Future Work

Although hybrid dilated convolution (HDC) is a simple solution to of "**gridding**"problem, we still can't determine the best dilated rate of each layer. Recently, some researchers propose a learning-based method to dynamically determine the shape of convolution kernel and achieve promising results [9]. It would be interesting to work on this direction and avoid more hand-crafted designs.

Bibliography

[1] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels. Technical report, 2010. 4.1.4

[2] Anurag Arnab, Sadeep Jayasumana, Shuai Zheng, and Philip Torr. Higher order potentials in end-to-end trainable conditional random fields. *arXiv preprint arXiv:1511.08119*, 2015. 2.4

[3] Aayush Bansal, Xinlei Chen, Bryan Russell, Abhinav Gupta, and Deva Ramanan. Pixelnet: Towards a general pixel-level architecture. *arXiv preprint arXiv:1609.06694*, 2016. 2.5, 3.1.2

[4] Siddhartha Chandra and Iasonas Kokkinos. Fast, exact and multi-scale inference for semantic image segmentation with deep gaussian crfs. *arXiv preprint arXiv:1603.08358*, 2016. 4.3

[5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016. 1.1, 1.2, 2.1, 2.2, 2.3, 2.4, 3.1.1, 3.1.2, 3.3, 3.3, 4.1.1, 4.1.2, 4.1.4, 4.3

[6] Tianqi Chen, Mu Li, Yutian Li, Min Lin, Naiyan Wang, Minjie Wang, Tianjun Xiao, Bing Xu, Chiyuan Zhang, and Zheng Zhang. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. *arXiv preprint arXiv:1512.01274*, 2015. 4

[7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 1.2, 4

[8] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. *arXiv preprint arXiv:1605.06409*, 2016. 2.2

[9] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. *arXiv preprint arXiv:1703.06211*, 2017. 5.2

[10] Alexey Dosovitskiy, Jost Tobias Springenberg, and Thomas Brox. Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1538–1546, 2015. 2.3

[11] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer*

vision, 88(2):303–338, 2010. 2.1, 4

[12] Philipp Fischer, Alexey Dosovitskiy, Eddy Ilg, Philip Häusser, Caner Hazırbaş, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. *arXiv preprint arXiv:1504.06852*, 2015. 2.3

[13] Jannik Fritsch, Tobias Kuehnl, and Andreas Geiger. A new performance measure and evaluation benchmark for road detection algorithms. In *International Conference on Intelligent Transportation Systems (ITSC)*, 2013. 4

[14] Golnaz Ghiasi and Charless Fowlkes. Laplacian reconstruction and refinement for semantic segmentation. *arXiv preprint arXiv:1605.02264*, 2016. 4.1.4

[15] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *2011 International Conference on Computer Vision*, pages 991–998. IEEE, 2011. 4.3

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1.1, 1.2, 2.1, 3.1.2, 2

[17] Matthias Holschneider, Richard Kronland-Martinet, Jean Morlet, and Ph Tchamitchian. A real-time algorithm for signal analysis with the help of the wavelet transform. In *Wavelets*, pages 286–297. Springer, 1990. 2.2

[18] Iasonas Kokkinos. Pushing the boundaries of boundary detection using deep learning. *arXiv preprint arXiv:1511.07386*, 2015. 1.1, 2.4

[19] P Krähenbühl and V Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *Advances in neural information processing systems*, 2011. 2.4, 4.1.2

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1.1

[21] Guosheng Lin, Chunhua Shen, Ian Reid, et al. Efficient piecewise training of deep structured models for semantic segmentation. *arXiv preprint arXiv:1504.01013*, 2015. 1.1, 1.2, 2.3, 2.4, 4.1.4

[22] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation. *arXiv preprint arXiv:1611.06612*, 2016. 4.1.4

[23] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014. 4.3

[24] Ziwei Liu, Xiaoxiao Li, Ping Luo, Chen-Change Loy, and Xiaoou Tang. Semantic image segmentation via deep parsing network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1377–1385, 2015. 1.1, 1.2, 2.3, 2.4

[25] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 1.1, 2.3, 2.5, 3.1.2, 4.1.4

[26] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1520–1528, 2015. 2.3

[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 2.1, 3.1.2

[28] Laura Sevilla-Lara, Deqing Sun, Varun Jampani, and Michael J Black. Optical flow with semantic segmentation and localized layers. *arXiv preprint arXiv:1603.03911*, 2016. 2.2

[29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1.1, 2.1, 4.1.4

[30] Haiming Sun, Di Xie, and Shiliang Pu. Mixed context networks for semantic segmentation. *arXiv preprint arXiv:1610.05854*, 2016. 2.5, 4.3

[31] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *arXiv preprint arXiv:1602.07261*, 2016. 3.1.2

[32] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016. 2.2

[33] Zifeng Wu, Chunhua Shen, and Anton van den Hengel. High-performance semantic segmentation using very deep fully convolutional networks. *arXiv preprint arXiv:1604.04339*, 2016. 1.2, 2.1

[34] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *arXiv preprint arXiv:1611.05431*, 2016. 3.1.2

[35] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 1.1, 1.2, 2.2, 3.1.2, 3.3, 4.1.4

[36] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014. 2.3

[37] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. *arXiv preprint arXiv:1612.01105*, 2016. 4.1.4

[38] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1529–1537, 2015. 1.1, 2.4

[39] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *arXiv preprint arXiv:1608.05442*, 2016. 1.2