DenseASPP for Semantic Segmentation in Street Scenes

Maoke Yang Kun Yu Chi Zhang Zhiwei Li Kuiyuan Yang
DeepMotion
{maokeyang, kunyu, chizhang, zhiweili, kuiyuanyang}@deepmotion.ai

Abstract

Semantic image segmentation is a basic street scene understanding task in autonomous driving, where each pixel in a high resolution image is categorized into a set of semantic labels. Unlike other scenarios, objects in autonomous driving scene exhibit very large scale changes, which poses great challenges for high-level feature representation in a sense that multi-scale information must be correctly encoded. To remedy this problem, atrous convolution[14] was introduced to generate features with larger receptive fields without sacrificing spatial resolution. Built upon atrous convolution, Atrous Spatial Pyramid Pooling (ASPP)[2] was proposed to concatenate multiple atrous-convolved features using different dilation rates into a final feature representation. Although ASPP is able to generate multi-scale features, we argue the feature resolution in the scale-axis is not dense enough for the autonomous driving scenario. To this end, we propose Densely connected Atrous Spatial Pyramid Pooling (DenseASPP), which connects a set of atrous convolutional layers in a dense way, such that it generates multi-scale features that not only cover a larger scale range, but also cover that scale range densely, without significantly increasing the model size. We evaluate DenseASPP on the street scene benchmark Cityscapes[4] and achieve state-of-the-art performance.

1. Introduction

With Fully Convolutional Network (FCN)[16], semantic image segmentation has achieved promising results with significantly improved feature representation. High level semantic information plays a crucial role in achieving high performance for a segmentation network. To extract high level information, FCN uses multiple pooling layers to increase the receptive field size of an output neuron. However, increased number of pooling layers leads to reduced feature map size, which poses serious challenges to up-sample the segmentation output back to full resolution. On the other hand, if we output the segmentation from an early layer with larger resolution, we were not able to make use of higher level semantics for better reasoning.

Figure 1. Illustration of challenging scale variations of street scenes from Cityscapes [4]. In the first exemplar image, the same category such as person varies largely in scale caused by distance to the camera. The second exemplar image illustrates an even challenging case where a large bus is close while several small traffic lights are far away.

Atrous convolution[14] is proposed to resolve the contradictory requirements between larger feature map resolution and larger receptive fields. An atrous kernel can be *dilated* in varied rates by inserting zeros into appropriate positions in the kernel mask. Compared to the traditional convolution operator, atrous convolution is able to achieve a larger receptive field size without increasing the numbers of kernel parameters. A feature map produced by an atrous convolution can be as the same size as the input, but with each output neuron possessing a larger receptive field, and therefore encoding higher level semantics.

Although atrous convolution solves the contradiction between feature map resolution and receptive field size, a method that simply generates a semantic mask from atrous-convolved feature map still suffers from a limitation. Specifically, all neurons in the atrous-convolved feature map share the same receptive field size, which means the process of semantic mask generation only made use of features from a single scale. However, experiences [24, 2, 3] show that multi-scale information would help resolve ambiguous cases and results in more robust classifi-

cation. To this end, ASPP[2, 3] proposed to concatenate feature maps generated by atrous convolution with different dilation rates, so that the neurons in the output feature map contain multiple receptive field sizes, which encode multi-scale info and eventually boost performance.

However, ASPP still suffers from another limitation. Specifically, input images under the autonomous driving scenarios are of high resolution, which requires neurons to have even larger receptive field. To achieve a large enough receptive field in ASPP, a large enough dilation ratio has to be employed. However, as the dilation rate increases (e.g. $d > 24$), the atrous convolution becomes more and more ineffective and gradually loses it modeling power[3]. Therefore, it is important to design a network structure that is able to encode multi-scale information, and simultaneously achieves a large enough receptive field size.

This motivates us to propose Dense Atrous Spatial Pyramid Pooling (DenseASPP) to solve challenging scale variations in street scenes as illustrated in Fig. 1. DenseASPP consists of a base network followed by a cascade of atrous convolution layers. It uses dense connections to feed the output of each atrous convolution layer to all unvisited atrous convolution layers ahead, see Fig. 2. In DenseASPP, each atrous convolution layer only makes use of atrous filters with reasonable dilation rate ($d \leq 24$). By the series of atrous convolutions, neurons at later layers obtain larger and larger receptive field without suffering from the kernel degradation issue in ASPP. And by the series of feature concatenations, neurons at each intermediate feature map encode semantic information from multiple scales, and different intermediate feature maps encode multi-scale information from different scale ranges. Therefore, the final output feature map in DenseASPP not only covers semantic information in a large scale range, but also covers that range in a very dense manner, see Fig. 3. We evaluates DenseASPP on Cityscapes datasets and achieves state-of-the-art performance with an mean Intersection-over-Union score of 80.6%.

To summarize, this paper makes two following contributions:

1. DenseASPP is able to generate features that covers a very large scale range (in terms of receptive field sizes).

2. The generated features of DenseASPP are able to cover the above scale range in a very dense manner.

It is worth to note that the above two properties cannot be simultaneously achieved by simply stacking atrous convolutional layers in cascade or in parallel.

2. Related Work

Semantic image segmentation requires high-level features to represent each pixel for semantic prediction, and ConvNets become the backbone for high-level feature extraction [10, 24, 14, 2, 3].

Since ConvNets are designed to do prediction at the whole image level, multiple modifications are made for pixel-level prediction. Fully convolutional network transforms fully connected layers into convolution layers to enable a classification net to output a heat-map [16]. With several down-sampling layers, resolution of high-level feature maps is with quite low resolution (typically 1/32 of the input image), and bilinear up-sampling or deconvolution is used to recover the resolution.

To compensate the low resolution of high-level features, feature maps from middle or early layers are also used by skip-connections [6, 1, 20]. Due the low-resolution is caused by down-sampling layers, DeepLab [14] proposed to remove last few max-pooling layers and reconfigure the network use atrous convolution to reuse pre-trained weights. Instead of adding atrous convolution layers to remove pooling layers, more atrous convolution layers are stacked in cascade to further increase the receptive field size [3] to cover large objects and bring broad contexts.

In addition, multi-scale features is another important factor to segment objects are with various scales and ambiguous pixels requiring diverse range of contextual information. Following spatial pyramid matching [13], PSPNet [24] and ASPP [2] are proposed to concatenate features of multiple receptive field sizes together for final prediction, where PSPNet employs four spatial pyramid pooling (downsampling) layers in parallel to aggregate information from multiple receptive field sizes and assign to each pixel via upsampling. ASPP concatenates features from multiple atrous convolution layers with different dilation rates arranged in parallel. The proposed DenseASPP combines the advantages of using atrous convolution layers in parallel and in cascade, and generates features of more scales in a larger range.

DenseASPP is named after DenseNet [8] which can be viewed as a special case of DenseASPP by setting dilation rate as 1. Thus, DenseASPP also shares the advantages of DenseNet including alleviating gradient-vanishing problem and substantially fewer parameters. In dilated DenseNet, a concurrent work by Yee [23], the same idea is used for cardiac MRI segmentation.

3. Dense Atrous Spatial Pyramid Pooling

In this section, we start with preliminary knowledges of atrous convolution and atrous spatial pyramid pooling, then introduce the proposed approach.

3.1. Atrous convolution and pyramid pooling

Atrous convolution is first introduced in [14] to increasing receptive field while keeping the feature map resolution unchanged. In one dimensional case, let $y[i]$ denote output

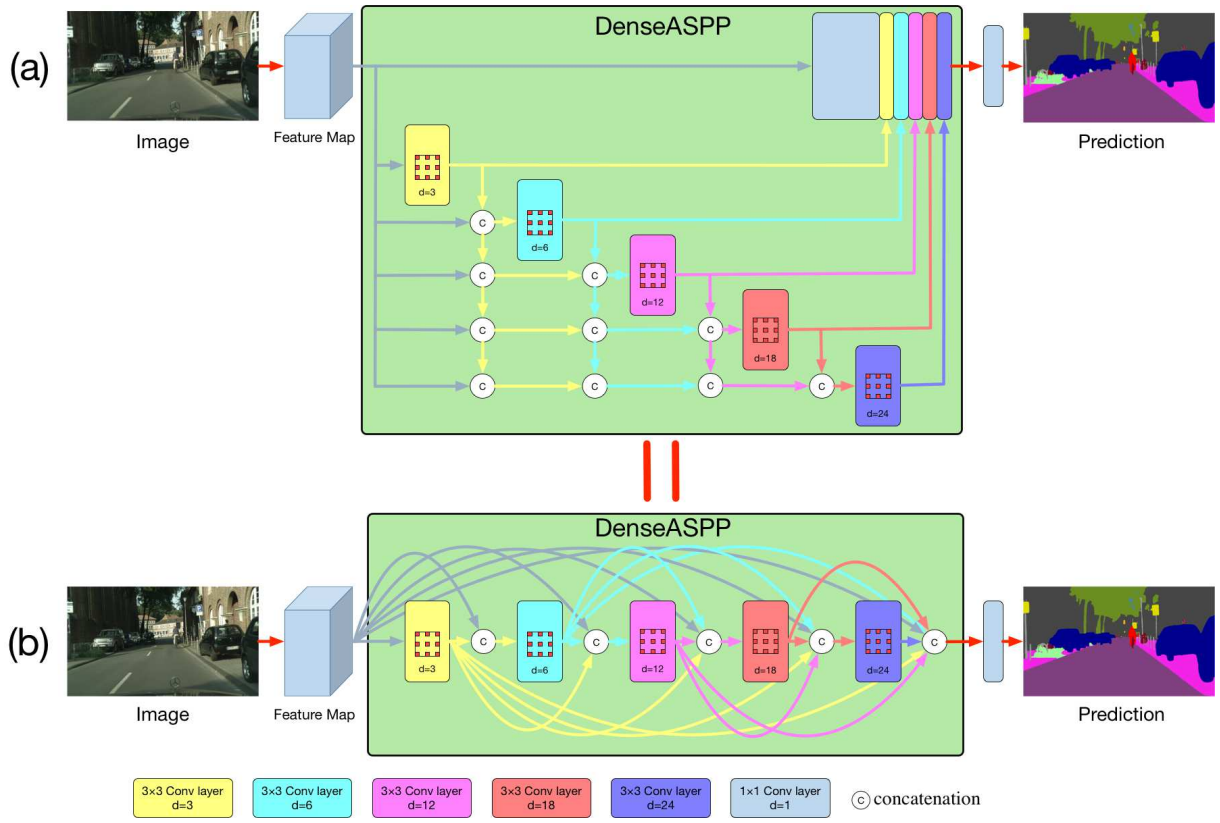

Figure 2. The structure of DenseASPP, (a) illustrate DenseASPP in detail, the output of each dilated convolutional layer is concatenated with input feature map, and then feed into the next dilated layer. Each path of DenseASPP compose a feature representation of correspond scale. (b) illustrate this structure in a more concrete version

signal and $x[i]$ denote input signal, atrous convolution can be formulated as follows:

$$y\left[i\right]=\sum_{k=1}^{K}x\left[i+d\cdot k\right]\cdot w\left[k\right] \tag{1}$$

where d is the dilation rate, $w[k]$ denotes the k-th parameter of filter, and K is the filter size. This equation reduces to a standard convolution when $d=1$. Atrous convolution is equivalent to convolving the input x with up-sampled filters produced by inserting $d-1$ zeros between two consecutive filter values. Thus, a large dilation rate means a large receptive field.

In street scene segmentation, objects usually have very different sizes. To handle this case, the feature maps must be able to cover different scales of receptive fields. For this goal, DeepLabV3 [3] proposed two strategies, i.e. *cascading* and *parallel* of several atrous convolutional layers with different dilation rates. In cascading mode, since a upper atrous layer accepts output of a lower atrous layer, it can efficiently produce large receptive fields. In parallel mode, since multiple atrous layers accept the same input and their outputs are concatenated together, the obtained output is indeed a sampling of the input with different scales of receptive fields. The parallel mode is formally termed as ASPP, which is an abbreviation of Atrous Spatial Pyramid Pooling in [2].

To simplify notations, we use $H_{K,d}(x)$ to term an atrous convolution, and consequently write ASPP as

$$y=H_{3,6}\left(x\right)+H_{3,12}\left(x\right)+H_{3,18}\left(x\right)+H_{3,24}\left(x\right) \tag{2}$$

In this work, motivated by DenseNets [8], we further push the boundaries of cascading and parallel strategies to a novel architecture which is able to generate much more densely scaled receptive fields than [3]. Experimental results on the Cityscapes[4] dataset demonstrated its effectiveness.

3.2. Denser feature pyramid and larger receptive field

The structure of DenseASPP is illustrated in Fig. 2(a). The atrous convolutional layers are organized in a cascade fashion, where the dilation rate of each layer increases layer by layer. Layers with small dilation rates are put in the lower part, while layers with large dilation rates are put in

the upper part. The output of each atrous layer is concatenated with the input feature map and all the outputs from lower layers, and the concatenated feature map is fed into the following layer. The final output of DenseASPP is a feature map generated by multi-rate, multi-scale atrous convolutions. The proposed structure can simultaneously compose a much denser and larger feature pyramid using only a few atrous convolutional layers. Following equation (2), each atrous layer in DenseASPP can be formulated as follows:

$$y_l = H_{K,d_l}\left([y_{l-1}, y_{l-2}, \cdots, y_0]\right) \tag{3}$$

where d_l represents the dilation rate of layer l, and $[\cdots]$ denotes the concatenation operation. $[y_{l-1}, \cdots, y_0]$ means the feature map formed by concatenating the outputs from all previous layers. Compared with the original ASPP [3], DenseASPP stacks all dilated layers together, and connects them with dense connections. This change brings us mainly two benefits: *denser feature pyramid*, and *larger receptive field*. We explain our network design in detail in terms of these two benefits in the following two subsections.

3.2.1 Denser feature pyramid

DenseASPP composes a denser feature pyramid than that in ASPP. The word 'denser' not only means better scale diversities of the feature pyramid, but also means there are more pixels involved in convolution than that in ASPP.

Denser scale sampling: DenseASPP is an effective architecture to sample inputs at different scales. A key design of DenseASPP is using dense connections to enable diverse ensembling of layers with different dilation rates. Each ensemble is equivalent to a kernel in different scale, i.e. different receptive field. Consequently, we get a feature map with many more scales than that in ASPP [2].

Dilation is able to increase receptive field of a convolution kernel. For an atrous convolutional layer with dilation rate d and kernel size K, the equivalent receptive field size is:

$$R = (d-1) \times (K-1) + K \tag{4}$$

For example, for a 3×3 convolutional layer with $d = 3$, the corresponding receptive field size is 7.

Stacking two convolutional layers together can give us a larger receptive field. Suppose we have two convolution layers with the filter size K_1 and K_2 respectively, the new receptive field is:

$$K = K_1 + K_2 - 1 \tag{5}$$

For example, a convolutional layer with kernel size 7 stacked with a convolutional layer with kernel size 13 will result in a receptive field of size 19.

Fig3 illustrates a simplified feature pyramid of DenseASPP to help readers better understand its scale diversity. This DenseASPP is constructed with dilation rate of 3, 6, 12, 18. The number(s) in each strip represent the combination of different dilation rate, and the length of each strip represents the equivalent kernel size of each combination. It is obvious that dense connections between stacked atrous layers are able to compose feature pyramid with much denser scale diversity. The receptive fields that DenseASPP ensembles are a super set of that in ASPP.

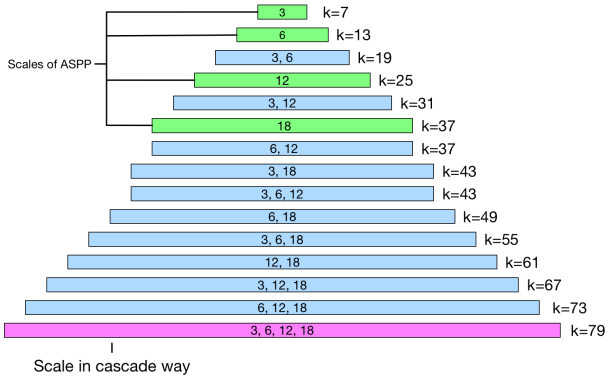

Figure 3. Illustration of DenseASPP's scale pyramid corresponding to the setting of densely stacking atrous convolutions with dilation rates (3, 6, 12, 18). DenseASPP produces feature pyramid with much larger scale diversity (i.e. high resolution in the scale-axis) and larger receptive field. k on the right side of each strip represents the receptive field size of the corresponding combination.

Denser pixel sampling: Compared to ASPP, DenseASPP gets more pixels involved in the computation of feature pyramid. ASPP composes feature pyramid using 4 atrous convolutional layers with dilation rate of 6, 12, 18, 24. The pixel sampling rate of an atrous convolutional layer with large dilation rate is quite sparse compared to a traditional convolution layer of the same receptive field.

Fig. 4(a) illustrates a traditional one-dimensional atrous convolution layer, which has a dilation rate of 6. This convolution have a receptive field size 13. However, in such a large kernel, only 3 pixels are sampled for calculation. This phenomenon gets worse in the two-dimensional case. Although large receptive fields are obtained, a lot of information is abandoned in the calculation process.

The situation is quite different in DenseASPP. Dilation rate increases layer by layer in DenseASPP, thus, convolutions in the upper layer can employ features from the lower layers, and make pixel sampling denser. Fig. 4(b) illustrates this process: an atrous layer with dilation rate 3 is put below the layer with dilation rate 6. For the original atrous layer with dilation rate of 6, information of 7 pixels will contribute for the final calculation, which is denser than the original 3 pixels. In the two-dimensional case, 49 pixels will contribute for the final prediction while in the standard

one-layer dilated convolution only 9 pixels will contribute. This phenomenon becomes more obvious when the dilation rate goes larger. Fig. 4(c) illustrates the phenomenon for the 2D version. The convolutional layer with larger dilation rate can draw help from the filter with smaller dilation rate, and samples pixels in a much denser way.

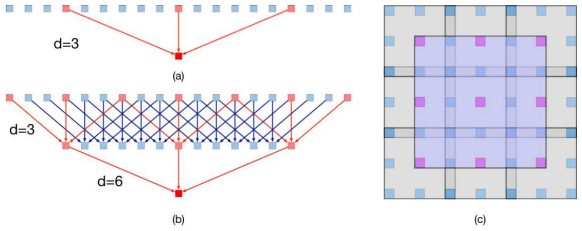

Figure 4. (a) Standard one-dimensional atrous convolution with dilation rate of 6. (b)Stacking an atrous layer with small dilation rate below an atrous layer with larger dilation rate makes a denser sampling rate. Red color denotes where the information come from. (c)Two-dimensional version of (b).

3.2.2 Larger receptive field

Another benefit brought by DenseASPP is the larger receptive field. Atrous convolutional layers works in parallel in traditional ASPP, and four sub-branches do not share any information in the feed forward process. To the opposite, atrous convolutional layers in DenseASPP share information through skip connections. Layers with small dilation rate and large dilation rate work interdependently, in which the feed forward process will not only compose a denser feature pyramid, but also come up with a larger filter to perceive larger context.

Following equation (5), let R_{max} denote the largest receptive filed of a feature pyramid, and function $R_{K,d}$ means that of a convolutional layer with kernel size K and dilation rate d. Thus, the largest receptive field of ASPP(6, 12, 18, 24) is:

$$\begin{aligned} R_{max} &= max\left[R_{3,6}, R_{3,12}, R_{3,18}, R_{3,24}\right] \\ &= R_{3,24} \\ &= 51 \end{aligned} \tag{6}$$

while in the case of DenseASPP(6, 12, 18, 24), the largest receptive field is:

$$\begin{aligned} R_{max} &= R_{3,6} + R_{3,12} + R_{3,18} + R_{3,24} - 3 \\ &= 122 \end{aligned} \tag{7}$$

Such a large receptive field can provide global information for large objects in high resolution images. For example, the resolution of Cityscapes [4] is 2048×1024, and the last feature map of our segmentation network is 256×128. DenseASPP(6, 12, 18, 24) covers a feature map size of 122, and DenseASPP(3, 6, 12, 18, 24) covers a larger feature map size of 128.

3.3. Model size control

To control model size and to prevent the network from growing too wide, we followed [8] to add a 1×1 convolutional layer before every dilated layer in DenseASPP to reduce feature map's depth into half of its original size. Besides, thin filters are used to further control the output size.

Suppose every atrous layer outputs n feature maps, DenseASPP have c_0 feature maps as input, and the l-th 1×1 convolutional layer before l-th dilated layer have c_l input feature maps, we have:

$$c_l = c_0 + n \times (l-1) \tag{8}$$

In our setting, every 1×1 convolutional layer before the dilated layer reduces the dimension into $c_0/2$ channels. And we set $n = c_0/8$ for all atrous layers in DenseASPP. Thus, the number of parameters in DenseASPP can be calculated as follows:

$$\begin{aligned} S &= \sum_{l=1}^{L} \left[c_l \times 1^2 \times \frac{c_0}{2} + \frac{c_0}{2} \times K^2 \times n \right] \\ &= \sum_{l=1}^{L} \left[\frac{c_0}{2} \left(c_0 + (l-1) \times \frac{c_0}{8} \right) + \frac{c_0}{2} \times K^2 \times \frac{c_0}{8} \right] \\ &= \frac{c_0^2}{32} \left(15 + L + 2K^2\right) L \end{aligned} \tag{9}$$

where L is the number of atrous layers in DenseASPP, and K is the kernel size. For example, the feature map of DenseNet121 have 512 channels, thus n is set to 64 for DenseNet121-based model. Besides, before every atrous layer, the number of channels in a feature map is reduced into 256 by a 1×1 convolutional layer. Consequently, DenseASPP (3, 6, 12, 18, 24) outputs a feature map with 832 channels, totaling 1,556,480 parameters, which is much smaller than the model size of DenseNet121 (nearly 1×10^7 parameters).

4. Experiments

DenseASPP is proposed to tackle the scale variations and contextual information demanding in street scenes, we empirically verify it on Cityscapes dataset in this section. Cityscapes [4] is comprised of a large, diverse set of high-resolution (2048×1024) images recorded in streets, where 5000 of these images have high quality pixel-level labels of 19 classes and results 9.43×10^9 labeled pixels in total. Following the standard setting of Cityscapes, the 5000 images are split into 2975 training and 500 validation images with publicly available annotation, as well as 1525 test images with annotations withheld and comparison to other methods is performed via a dedicated evaluation server. For quantitative evaluation, mean of class-wise Intersection over Union (mIoU) are used.

Table 1. Category-wise comparison on the Cityscapes test set.

Methods	mIoU	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle
FCN-8s [16]	65.3	97.4	78.4	89.2	34.9	44.2	47.4	60.1	65	91.4	69.3	93.9	77.1	51.4	92.6	35.3	48.6	46.5	51.6	66.8
DeepLabv2-CRF[2]	70.4	97.9	81.3	90.3	48.8	47.4	49.6	57.9	67.3	91.9	69.4	94.2	79.8	59.8	93.7	56.5	67.5	57.5	57.7	68.8
FRRN[19]	71.8	98.2	83.3	91.6	45.8	51.1	62.2	69.4	72.4	92.6	70	94.9	81.6	62.7	94.6	49.1	67.1	55.3	53.5	69.5
RefineNet[15]	73.6	98.2	83.3	91.3	47.8	50.4	56.1	66.9	71.3	92.3	70.3	94.8	80.9	63.3	94.5	64.6	76.1	64.3	62.2	70
PEARL[11]	75.4	98.4	84.5	92.1	54.1	56.6	60.4	69	74	92.9	70.9	95.2	83.5	65.7	95	61.8	72.2	69.6	64.8	72.8
GCN[18]	76.9	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
DUC[21]	77.6	98.5	85.5	92.8	58.6	55.5	65	73.5	77.9	93.3	72	95.2	84.8	68.5	95.4	70.9	78.8	68.7	65.9	73.8
PSPNet[24]	78.4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
ResNet-38[22]	78.4	98.5	85.7	93.1	55.5	59.1	67.1	74.8	78.7	93.7	72.6	95.5	86.6	69.2	95.7	64.5	78.8	74.1	69	76.7
DenseASPP(Ours)	80.6	98.7	87.1	93.4	60.7	62.7	65.6	74.6	78.5	93.6	72.5	95.4	86.2	71.9	96.0	78.0	90.3	80.7	69.7	76.8

4.1. Implementation Details

We implement our methods on Pytorch [17]. For a ConvNet pre-trained on ImageNet [5], we first remove the last two pooling layers and the last classification layer, and set the dilation rates of the convolution layers after the two removed pooling layers to be 2 and 4 respectively to make the pre-trained weights reusable. The modified ConvNets outputs the basic feature map of $\frac{1}{8}$ input image resolution. ASPP, PSPNet and the proposed DenseASPP are all added on the basic feature map, and all output feature maps of $\frac{1}{8}$ input image resolution. Followed the feature map, a convolution layer with nineteen 1×1 filters are used to predict the $\frac{1}{8}$ label map, which is further up-sampled by a factor of 8 to define the cross entropy loss using the ground-truth label map.

Batch normalization [9] is used before each weight layer in our implementation to ease the training and make it is comparable to concatenate feature maps from different layers. To avoid over-fitting, common data augmentations are used as preprocessing, including random flipping horizontally, random scaling in the range of [0.5, 2], random brightness jittering within the range of [-10, 10], and random crop of 512×512 image patches.

For training, we use the Adam optimizer [12] with an initial learning rate of 0.0003 and weight decay of 0.00001. The learning rate is scheduled by multiplying the initial learning rate with $\left(1 - \frac{epoch}{maxEpoches}\right)^{0.9}$. All models are trained for 80 epochs with mini-batch size of 8. The statistics of batch normalization is updated on the whole mini-batch.

4.1.1 DenseASPP

We followed [2] to build our baseline model, the only difference is that DenseNet121 is used to replace ResNet101[7]. We compare the proposed DenseASPP with the original ASPP. For both ASPP and DenseASPP, we use four atrous convolutional layers with dilation rates 6, 12, 18, 24 respectively. All other settings are kept the same. The results are evaluated on the validation set of Cityscapes, and listed in Table 2. Results shows that DenseASPP significantly improve the segmentation performance over the baseline model by 4.2%, and some examples are shown in Fig. 5. Deeper pre-trained models are helpful to further improve performance.

Table 2. DenseASPP improve the performance of segmentation by a huge level.

Base model	Structure	mIoU
DenseNet121	ASPP(6, 12, 18, 24)	72.0%
DenseNet121	DenseASPP(6, 12, 18, 24)	76.2%
DenseNet169	DenseASPP(6, 12, 18, 24)	77.7%
DenseNet201	DenseASPP(6, 12, 18, 24)	78.9%

4.1.2 Detailed study on DenseASPP components

DenseASPP is composed of multiple atrous convolutional layers with different dilation rates. In this part, experiments are designed to study how different settings of DenseASPP influence the performance quantitatively. The results are evaluated on Cityscapes' validation set, and illustrated in Table 3. From these experiments, we can get two conclusions. First, the segmentation performance improves with increasing of max feature scales(largest receptive of DenseASPP) of DenseASPP. In fact, both adding more layers and use large dilation rates can increase max scale of DenseASPP. After the max scale goes larger than 128, which is the width of feature map of Cityscapes' image, the performance stopped increasing. Second, even with a relatively weak base model, i.e. DenseNet121, DenseASPP can obtain reasonably high performances.

4.1.3 Comparing with state-of-the-art

We train DenseASPP based on DenseNet161(wider) [8] with only fine annotated data, and submit the results on test set to the evaluation system of Cityscapes[4]. With multi-scale $\{0.5, 0.8, 1.0, 1.2, 1.5, 2.0\}$ testing, this model achieves mIoU of 80.6% on the test set. We compare our

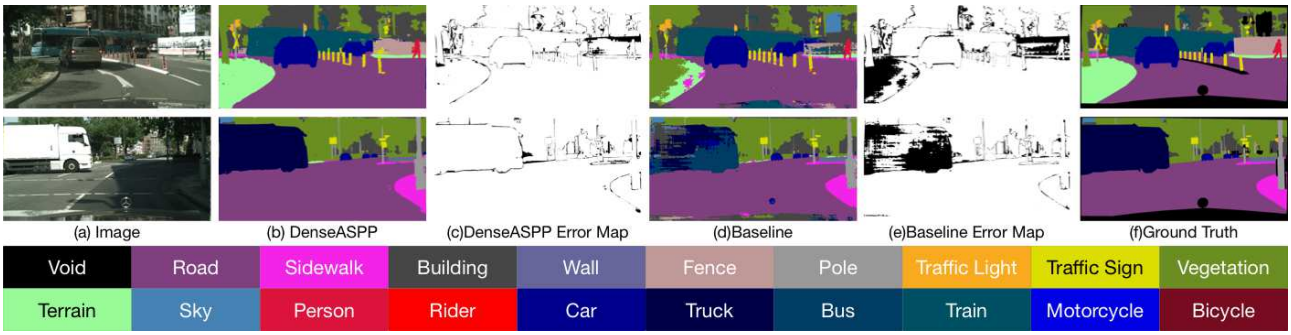

Figure 5. Due to the the ability to capture larger context, DenseASPP can correctly classify confusing categories 'vegetation' and 'terrain', and can distinguish 'fence' from the background.

Table 3. Performance of DenseASPP with different network settings

Structure	Max Scale	mIoU
DenseNet121 + DenseASPP(6, 12)	37	74.8%
DenseNet121 + DenseASPP(6, 12, 18)	73	75.6%
DenseNet121 + DenseASPP(3, 6, 12, 18)	79	75.7%
DenseNet121 + DenseASPP(6, 12, 18, 24)	122	76.2%
DenseNet121 + DenseASPP(3, 6, 12, 18, 24)	128	**76.6%**
DenseNet121 + DenseASPP(3, 6, 12, 18, 24, 30)	179	76.5%

results with state-of-the-art methods on Cityscapes, and the results is illustrated in Table 1 and Table 4. It is noted that we use the results reported in the original paper instead of the Cityscapes leader board.

Table 4. Performance comparison on Cityscapes test set.

Method	mIoU cla	iIoU cla	mIoU cat	iIoU cat
FCN-8s[16]	65.3	41.7	85.7	70.1
DeepLabv2-CRF[2]	70.4	42.6	86.4	67.7
FRRN[19]	71.8	45.5	88.9	75.1
RefineNet[15]	73.6	47.2	87.9	70.6
PEARL[11]	75.4	51.6	89.2	75.1
GCN[18]	76.9	-	-	-
DUC[21]	77.6	53.6	90.1	75.2
PSPNet[24]	78.4	56.7	90.6	78.6
ResNet-38[22]	78.4	59.1	90.9	81.1
DenseASPP(Ours)	80.6	57.9	90.7	78.1

4.2. Ablation Studies

The quality of the last feature map, which is the input of the decision making layer, i.e. softmax layer, is critical for an accurate segmentation. In this section, we first evaluate the feature map quality from both feature level and results level. Then, we study the two major reasons which affect feature maps significantly, i.e. size of receptive field and scale/pixel sampling rates.

4.2.1 Feature similarities

Context information is of great significance for distinguishing confusing categories and classifying large objects. In the Cityscapes dataset, some categories are easily to be misclassified each other due to similar appearances, e.g. 'vegetation' and 'terrain', 'bus','car', 'truck' and 'train'. For these categories, sufficient context information is critical.

Fig. 5 illustrates two examples where some confusing categories exist. Without enough context information, our baseline model cannot correctly classify 'terrain' and 'fence' in the first example, and misclassified the category of 'bus' and 'truck' and 'wall' in the second example. Our proposed model can correctly classify these images. These examples shows the ability of DenseASPP on modeling large context information.

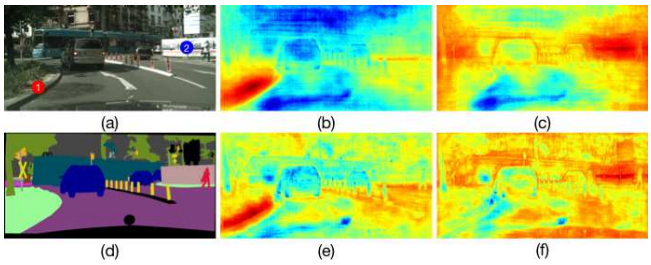

Figure 6. Illustration of feature similarities of all pixels to a pointed pixel. Hotter color means more similar in feature level. (a) and (d) are input image and corresponding ground-truth, (b) and (c) are results of DenseASPP at point 1 and 2 respectively, which are computed by output of DenseASPP. (e) and (f) are correspond results after DenseASPP is removed.

Feature level analysis are further performed to see how our proposed model classifies each pixel. The output of DenseASPP is a multi-channel feature map, based on which a softmax classification is employed to classify each pixel into one class. Thus, each position of the feature map correspond to a pixel, and pixels with the same label are likely have similar features. In order to study how DenseASPP

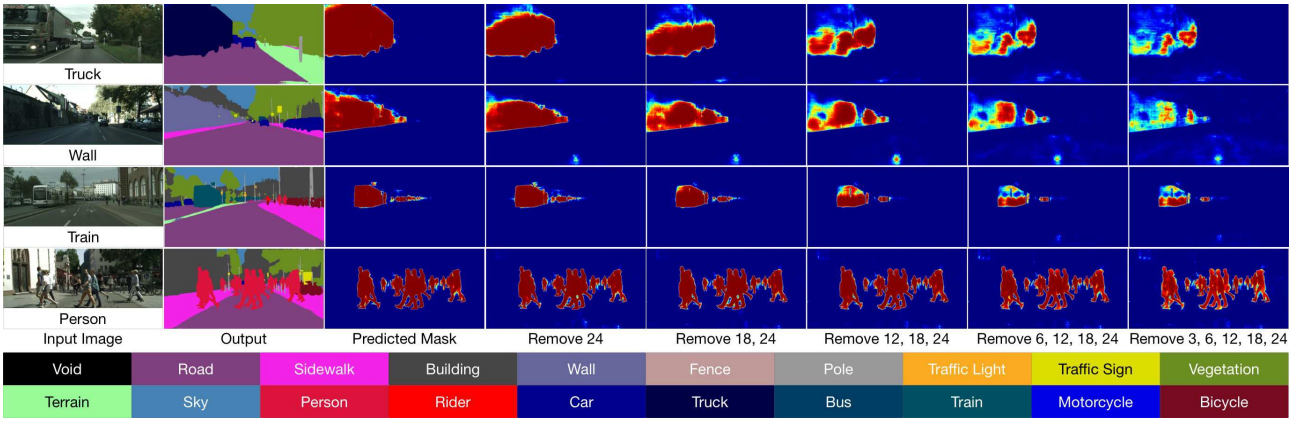

Figure 7. Effects of different spatial and pixel sampling rates. We gradually remove top atrous convolutional layers to visualize the results.

classifies a pixel, feature similarities between pixels are calculated in the whole feature map, and the results are illustrated by a heat-map as shown in Fig. 6. Cosine similarity is used to measure the similarities between features of pixels.

The similarity map of DenseASPP shows that most of pixels in a continuous region with the same label have similar features. This means we get similar features for pixels in this region. However, the hot regions of results are much smaller and spotted when the DenseASPP is removed. Consequently, pixels in this region are unlikely to be classified to the same category. Large context information brought by DenseASPP is critical for correct segmentation in this case.

4.2.2 Visualization of receptive field

Since the empirical receptive field sizes are often much smaller than the theoretical ones, we use the method proposed by [25] to visualize the empirical receptive field sizes. Specifically, for a feature vector representing a image patch with theoretical receptive field size, we use a 8×8 mean image to cover the image patch in a sliding-window way and record the changes of the feature vector measured by the Euclidean distances as a heat-map. The heat-map indicates which pixels actually affect the feature vector.

The convolution layer with largest dilation rate of ASPP(6, 12, 18, 24) and DenseASPP(6, 12, 18, 24) are visualized respectively, and illustrated in Fig. 8. It is obviously to see that dilated layer of DenseASPP sampling denser, and captures larger receptive field.

4.2.3 Illustration of scale diversity

The more densely connected atrous convolution layers we use in DenseASPP, the more densely sampled feature map we can get. Thus, gradually removing top atrous convolutional layers in DenseASPP will reduce the sampling rates in scale space, and consequent segmentation performances.

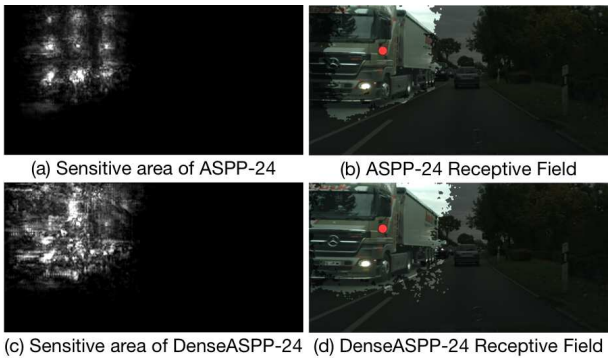

Figure 8. Comparing the receptive fields of ASPP and DenseASPP. Red dot means the reference pixel. Lighter pixels indicate strong correlations.

In this part, we did experiments to visualize such kinds of effects. Fig. 7 shows the results.

It is obvious that, with removing of top layers, the sampling rates in scale space decrease quickly. Large objects such as 'truck' and 'wall' is severely impacted. 'Train' in the middle scale is less affected. Results of small and easy objects, like 'person', are fine. This results further demonstrate the necessity of using densely connected cascaded atrous convolution layers in street scene segmentation.

5. Conclusion

In this paper, we propose DenseASPP to tackle the challenging problem of street scene segmentation where objects vary largely in scale. DenseASPP connects a set of atrous convolution layers in a dense way, which effectively generates densely spatial-sampled and scale-sampled features in a very large range. Theoretical analysis, visualization and quantitative experimental results on Cityscapes dataset are presented to demonstrate the effectiveness of DenseASPP.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *arXiv preprint arXiv:1606.00915*, 2016.

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

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

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

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

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arxiv 2015. *arXiv preprint arXiv:1512.03385*.

[8] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

[9] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

[10] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *CVPRW*, 2017.

[11] X. Jin, X. Li, H. Xiao, X. Shen, Z. Lin, J. Yang, Y. Chen, J. Dong, L. Liu, Z. Jie, et al. Video scene parsing with predictive feature learning. *arXiv preprint arXiv:1612.00119*, 2016.

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *Computer Science*, 2014.

[13] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006.

[14] C. Liang-Chieh, G. Papandreou, I. Kokkinos, k. murphy, and A. Yuille. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. In *ICLR*, 2015.

[15] G. Lin, A. Milan, C. Shen, and I. Reid. Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. *arXiv preprint arXiv:1611.06612*, 2016.

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

[17] A. Paszke, S. Gross, S. Chintala, et al. Pytorch, 2017.

[18] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun. Large kernel matters–improve semantic segmentation by global convolutional network. *arXiv preprint arXiv:1703.02719*, 2017.

[19] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe. Full-resolution residual networks for semantic segmentation in street scenes. *arXiv preprint arXiv:1611.08323*, 2016.

[20] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2015.

[21] P. Wang, P. Chen, Y. Yuan, D. Liu, Z. Huang, X. Hou, and G. Cottrell. Understanding convolution for semantic segmentation. *arXiv preprint arXiv:1702.08502*, 2017.

[22] Z. Wu, C. Shen, and A. v. d. Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *arXiv preprint arXiv:1611.10080*, 2016.

[23] C.-H. Yee. https://chuckyee.github.io/cardiac-segmentation/.

[24] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. *arXiv preprint arXiv:1612.01105*, 2016.

[25] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*, 2014.