Paper review: Multi-Scale Context Aggregation By Dilated Convolution

Paper written by Fisher Yu (Princeton University), Vladlen Koltun(Intel Labs)

Reviewed by Yujin Cheon (22.12.02)

Abstract

Semantic segmentation을 하는 최신(2016년) 모델들은 대부분 image classification을 하는 CNN을 베이스로 만들어졌습니다. 하지만 semantic segmentation과 같은 픽셀 단위의 dense prediction과 이미지 내 객체의 유무만 판단하면 되는 image classfication은 구조적으로 다르기 때문에 이 논문에서는 dense prediction을 위해 특별히 디자인한 CNN model인 DilatedNet을 제안합니다.

DilatedNet은 **해상도를 낮추지 않은 채로 다양한 스케일(multi-scale)의 contextual information을 얻기 위해 dilated convolution을 사용**합니다. Dilated convolution은 공간 정보의 손실 없이 **기하급수적으로 receptive field를 확장하는 방법입니다.** 기존의 CNN은 pooling layer의 max pooling이나 conv layer의 stride=2 등을 통해서 feature map의 크기를 축소한 뒤에 convolution을 통해 receptive field를 확장했다면, dilated convolution은 이 과정을 거치지 않기 때문에 공간적인 정보의 손실 없이 RF를 확장할 수 있게 되는 것입니다.

DilatedNet은 **front end module**과 **context module**로 이루어져 있습니다.

1. Introduction-DilatedNet이 나오게 된 배경

컴퓨터 비전의 많은 문제들은 dense prediction 문제들입니다. 그 중 대표적인 것은 semantic segmentation으로 이미지의 pixel 하나하나를 주어진 카테고리 중의 하나로 분류합니다. semantic segmentation이 어려운 이유는 서로 상반되어 보이는 **pixel level의 정확도**와 **다양한 스케일의 contextual reasoning**이 동시에 요구되기 때문입니다.

이미지 분류에 쓰이는 Convolutional network를 dense prediction용으로 바꿔서 (repurposed) 쓰는 경우들이 있다. 하지만 이는 **이미지 분류와 dense prediction의 구조적 차이에서 기인한 문제가 생깁니다**.

기존의 CNN은 주로 feature map을 축소시키는 downsampling을 통해 receptive field를 확장시켜서 이미지의 더 넓은 부분을 커버할 수 있게 되지만, 그와 동시에 pooling등으로 해상도를 낮추는 것이기 때문에, multi-scale contextual reasoning은 가능해도, pixel level accuracy는 잃어버리게 됩니다.

이 논문 전에도 이 문제를 해결하기 위한 두 가지 approach가 제기되었습니다.

(1) striding과 pooling을 반복하여 넓은 영역을 포함하는 global perspective을 가진 후, 잃어버린 해상도를 회복하기 위하여 **반복적으로 upsampling**을 하는 것이다. 저자는 upsampling을 할 것이라면 그렇게 severe한 downsampling이 필수적이었는지 의문을 제기함.

(2) 다양한 크기로 rescale된 이미지들을 동시에 input으로 주어 이 input들의 예측결과를 통합해 output을 산출하는 것이다. 저자는 이 접근법에 대해 rescaled input image를 각각의 분석하는 것이 필수적인지 물음.

그래서 이 논문에서는 위의 두가지 접근법이 아닌, 순전히 dense prediction만을 위해 고안한 모델을 제안한다. 이는 해상도와 coverage를 잃지 않고 receptive field만을 기하급수적으로 증가시키는 **dilated convolution**을 써서 만들어졌습니다.

구체적인 아키텍처로는 기존의 이미지 분류에서 쓰인 것을 dense prediction에 맞게 dilated convolution을 적용하여 바꾼 **front end module**. 이 front end module에서 나온 feature map을 input으로 가지는 **context module**이 있습니다.

지금부터 차례대로 소개하도록 하겠습니다.

2. Dilated Convolutions

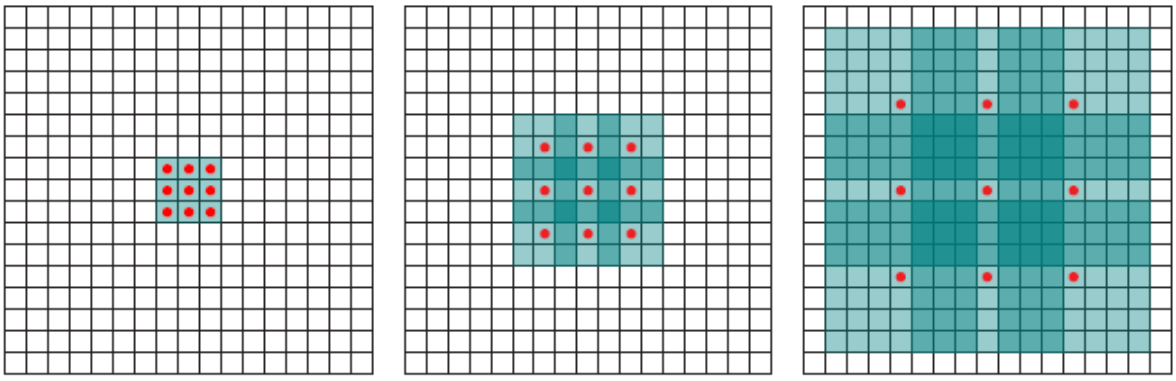


Figure 1: Systematic dilation supports exponential expansion of the receptive field without loss of resolution or coverage. (a) F_1 is produced from F_0 by a 1-dilated convolution; each element in F_1 has a receptive field of 3×3. (b) F_2 is produced from F_1 by a 2-dilated convolution; each element in F_2 has a receptive field of 7×7. (c) F_3 is produced from F_2 by a 4-dilated convolution; each element in F_3 has a receptive field of 15×15. The number of parameters associated with each layer is identical. The receptive field grows exponentially while the number of parameters grows linearly.

easy to see that the size of the receptive field of each element in F_{i+1} is $(2^{i+2} - 1) \times (2^{i+2} - 1)$. The receptive field is a square of exponentially increasing size. This is illustrated in Figure 1.

dilated convolution은 resolution 손실 없이 기하급수적으로 receptive field를 확장할 수 있으며, receptive field가 확장되어도 파라미터 수는 일정하게 유지된다.

(a)의 filter size는 3x3이며, 파라미터는 9개를 가지고 있다. dilated factor=1. (b)는 dilated factor=2인 경우. 빨간점을 제외한 filter의 가중치는 0으로 채워져 있어 9개의 파라미터를 가지지만 receptive field는 7x7입니다. 즉, maxpooling이나 stride=2 conv를 적용하지 않아도 receptive field를 확장할 수 있어 공간 정보 훼손을 감소할 수 있고, 연산량을 유지하며 receptive field를 확장할 수 있다는 장점이 있습니다. (c)는 dilated factor=4인 경우. parameter 수는 앞의 경우와 같이 9개이고 receptive field는 15x15입니다.

- Dilated Convolution Formula

Let $F : \mathbb{Z}^2 \to \mathbb{R}$ be a discrete function. Let $\Omega_r = [-r, r]^2 \cap \mathbb{Z}^2$ and let $k : \Omega_r \to \mathbb{R}$ be a discrete filter of size $(2r + 1)^2$. The discrete convolution operator $*$ can be defined as

$$(F * k)(\mathbf{p}) = \sum_{\mathbf{s}+\mathbf{t}=\mathbf{p}} F(\mathbf{s})\, k(\mathbf{t}). \tag{1}$$

We now generalize this operator. Let l be a dilation factor and let $*_l$ be defined as

$$(F *_l k)(\mathbf{p}) = \sum_{\mathbf{s}+l\mathbf{t}=\mathbf{p}} F(\mathbf{s})\, k(\mathbf{t}). \tag{2}$$

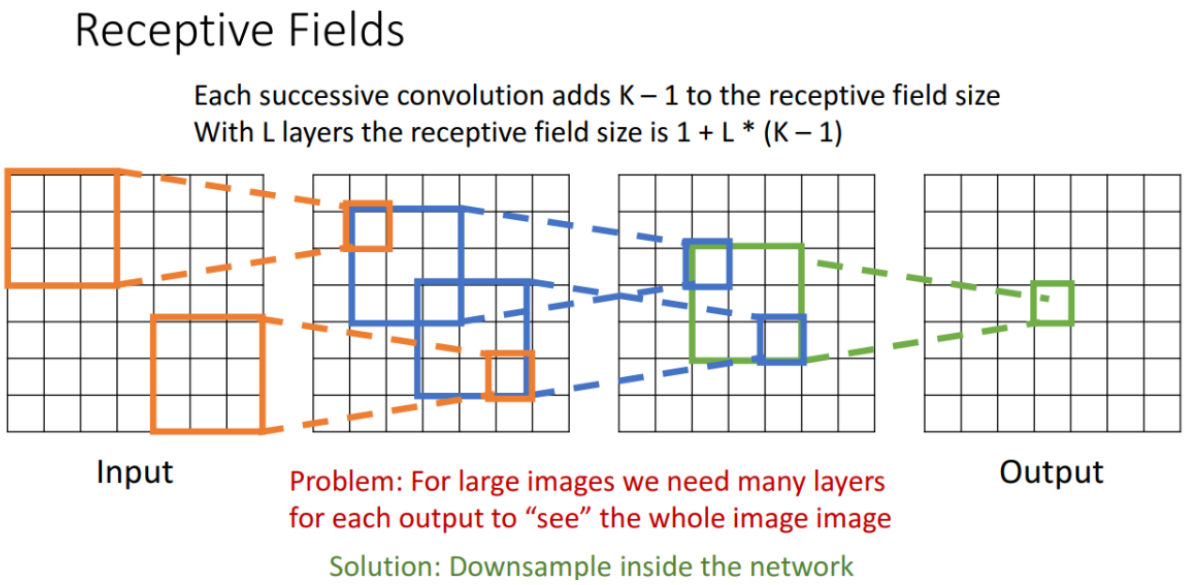
We will refer to $*_l$ as a dilated convolution or an l-dilated convolution. The familiar discrete convolution $*$ is simply the 1-dilated convolution.

F(s): feature map, function. s: 기존의 receptive field 크기..?

k(t): filter, kernel. t: filter size-1 ..?

p: receptive field. 보통은 s+t로 계산됨. 하지만 dilated conv의 경우 s+l*t로 계산됨. l=dilated factor.

- Recap on Receptive Field calculation>



위 예시는 3-conv layers일때의 receptive field 예시입니다. Input layer부터 Output layer까지 3x3 convolution을 세 번 거치는데, output에서 input으로 거꾸로 거슬러 올라가면 output의 cell 하나가 그 전 layer의 3x3 region에서 왔고, 3x3 region은 또 전 layer의 5x5 region에서 왔으며, 결국 거슬러 올라가 input layer의 7x7 region에서 온 것입니다. 따라서 output layer에서 receptive field 사이즈는 7x7입니다. 이러한 receptive field size를 구하는 방법은 1+L*(K-1)로, kernel size-1을 layer의 개수 L만큼 더해줍니다.

Dilated conv의 경우는 dilated factor를 K-1앞에 곱해줌으로 기존보다 훨씬 receptive field를 확장합니다.

- Convolution with a Dilated Filter vs Dilated Convolution

이 논문이 쓰이기 전 dilated convolution은 “convolution with a dilated filter”이라는 이름으로 불렸습니다. 이 모델에서는 dilated filter 자체가 만들어지는 것이 아니라, 기존의 평범한 filter를 가지고 convolution operator가 dilated conv 연산을 할 수 있도록 수정되어 작동하기 때문에 dilated convolution이라고 따로 정의하였다고 합니다.

3. Multi-Scale Context Aggregation (Context Module)

DilatedNet은 front end module과 context module로 구성되어 있습니다.

context module은 multi-scale contextual information을 종합하여 dense prediction의 성능을 향상시킵니다. 이 모듈의 특징은 C개의 feature map을 입력 받으면, C개의 feature map을 출력합니다. input과 output의 크기와 채널 수가 동일하므로 기존의 다른 dense prediction 구조에 plug in 해서 사용할 수 있습니다.

Context Module은 8 layer의 dilated conv로 이루어져 있으며, basic form과 large form이 있습니다.

Layer	1	2	3	4	5	6	7	8
Convolution	3×3	3×3	3×3	3×3	3×3	3×3	3×3	1×1
Dilation	1	1	2	4	8	16	1	1
Truncation	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No
Receptive field	3×3	5×5	9×9	17×17	33×33	65×65	67×67	67×67
Output channels								
Basic	C	C	C	C	C	C	C	C
Large	$2C$	$2C$	$4C$	$8C$	$16C$	$32C$	$32C$	C

Table 1: Context network architecture. The network processes C feature maps by aggregating contextual information at progressively increasing scales without losing resolution.

- Basic form
  - front end module에서 나온 feature map을 입력으로 받아, 여러 layer를 거치며 contextual information을 얻어서 accuracy를 증가시킬 수 있음.

- 모든 layer가 C개의 채널을 가짐.
- 3x3 convolution 7개의 layer, 마지막 1x1 convolution layer: 처음 7개의 layer 각각 dilated factor [1,1,2,4,8,16,1] 을 가지는 conv 연산을 하고, 마지막 layer에서는 1x1xC의 필터로 conv 연산을 해서 output의 크기를 조정함.
- front end module에서 나온 feature map이 64x64 사이즈이기 때문에 receptive field의 확장은 layer6까지로 함.
- filter 가중치의 random initialization은 accuracy를 증가시키지 못함. 따라서 아래 같은 identity ideninitialization을 사용:

$$k^b(\mathbf{t}, a) = 1_{[\mathbf{t}=0]} 1_{[a=b]},$$

a는 input feature map의 index b는 output map의 index이다. 위의 initialization은 전 layer가 그대로 다음 layer로 통과할 수 있도록 한다.

- Large Form
  - 각 layer마다 다양한 크기의 채널 수를 가짐
  - initialization formula

layers. Let c_i and c_{i+1} be the number of feature maps in two consecutive layers. Assume that C divides both c_i and c_{i+1}. The initialization is

$$k^b(\mathbf{t}, a) = \begin{cases} \frac{C}{c_{i+1}} & \mathbf{t} = 0 \text{ and } \left\lfloor \frac{aC}{c_i} \right\rfloor = \left\lfloor \frac{bC}{c_{i+1}} \right\rfloor \\ \varepsilon & \text{otherwise} \end{cases} \tag{5}$$

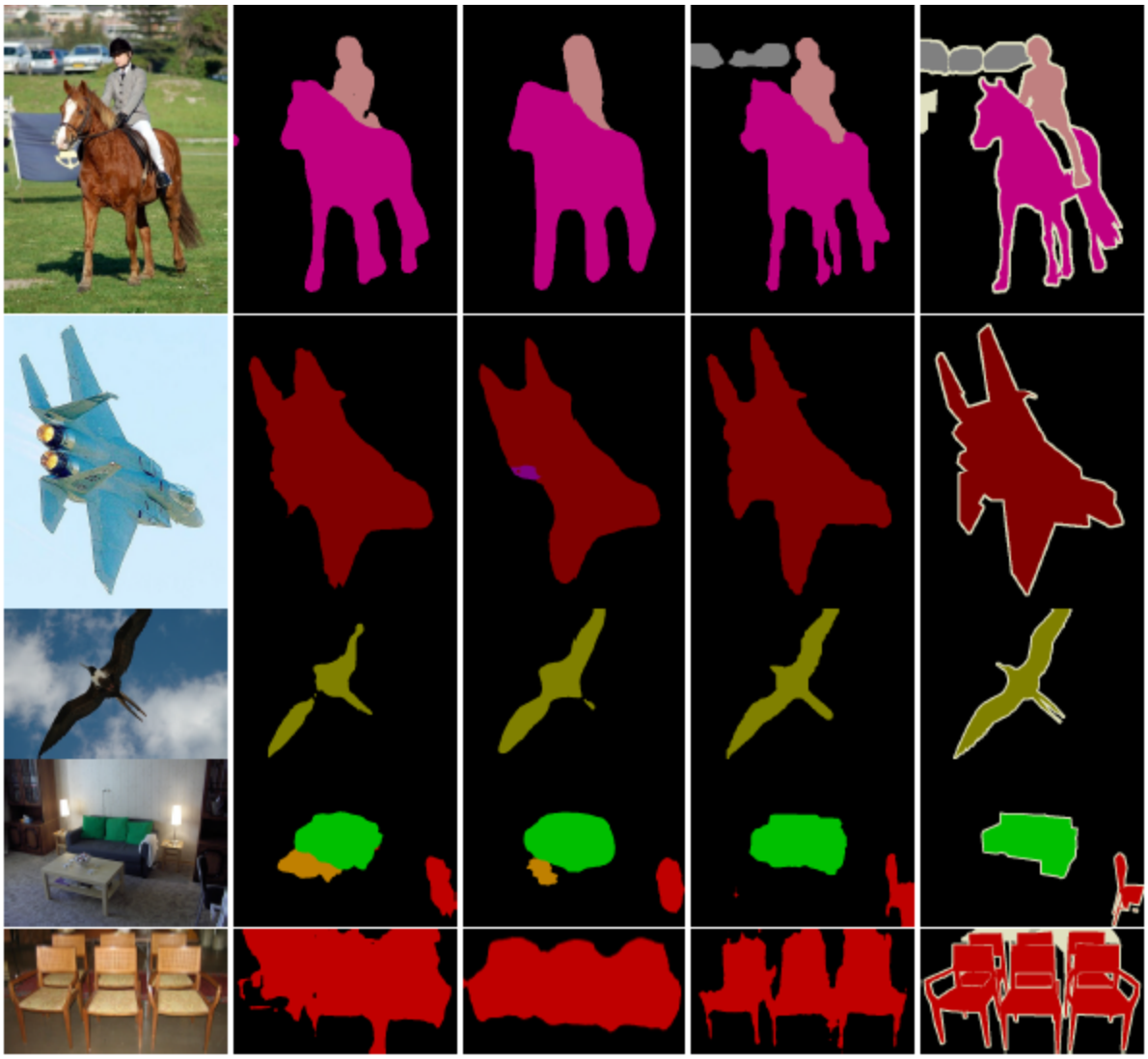
Here $\varepsilon \sim \mathcal{N}(0, \sigma^2)$ and $\sigma \ll C/c_{i+1}$. The use of random noise breaks ties among feature maps with a common predecessor.

4. Front End Module

front end 모듈은 context module의 앞에 연결되어 있는 모듈로, 3 채널의 컬러 이미지를 입력받아 C=21의 feature map을 생성하고 생성된 64x64x21의 feature map을 context module로 전달합니다.

Front end module은 VGG-16 net을 사용하여 만들어졌습니다. VGGNet의 마지막 두 conv-pool-conv-pool layer들을 2dilated conv-2dilated conv의 구조가 되도록 변경합니다. 즉 pooling layer를 두 개 다 지우고 convolution은 1-dilated conv에서 2-dilated conv로 각각 바꾸어 마지막 layer가 dilated factor=4만큼 dilation이 되도록 합니다.

기존의 classfication모델에서 마지막 두 개의 pooling layer를 제거하여 만들어진 이 front end module은 Pascal VOC 2012 test set에서 뛰어난 성능을 보여주었다.



(a) Image (b) FCN-8s (c) DeepLab (d) Our front end (e) Ground truth

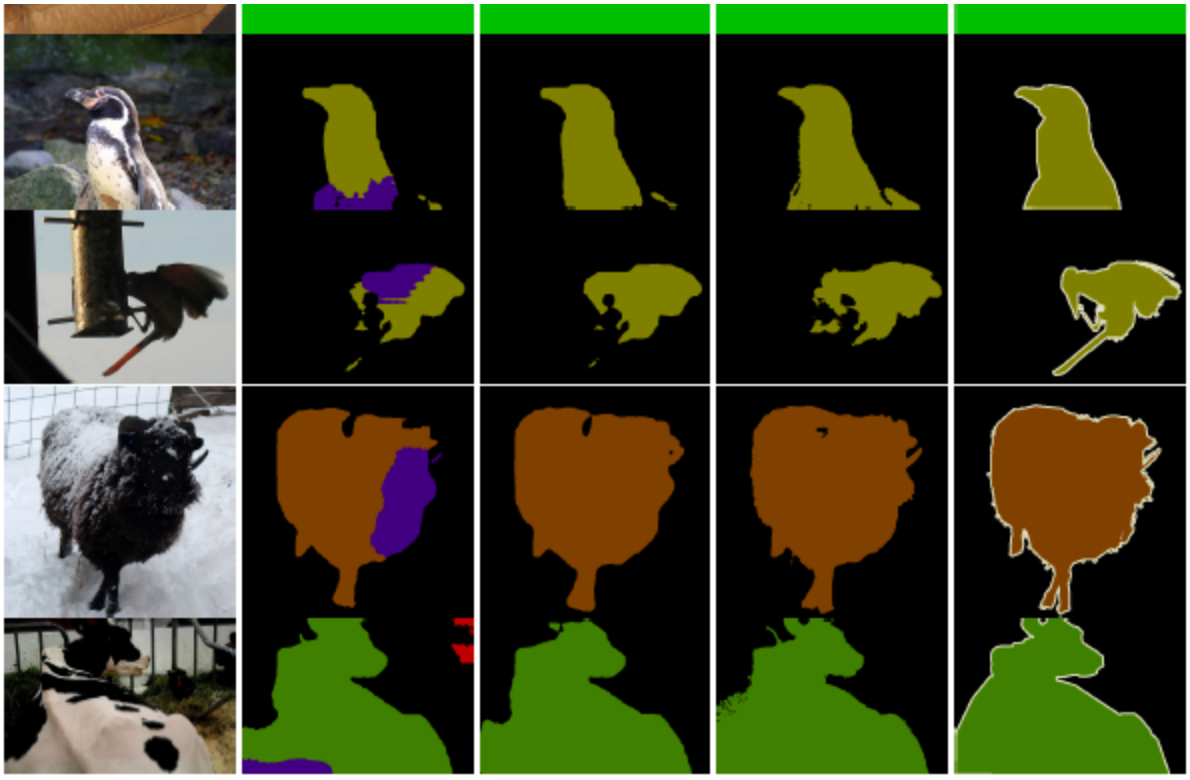
Figure 2: Semantic segmentations produced by different adaptations of the VGG-16 classification network. From left to right: (a) input image, (b) prediction by FCN-8s (Long et al., 2015), (c) prediction by DeepLab (Chen et al., 2015a), (d) prediction by our simplified front-end module, (e) ground truth.

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean IoU
FCN-8s	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
DeepLab	72	31	71.2	53.7	60.5	77	71.9	73.1	25.2	62.6	49.1	68.7	63.3	73.9	73.6	50.8	72.3	42.1	67.9	52.6	62.1
DeepLab-Msc	74.9	34.1	72.6	52.9	61.0	77.9	73.0	73.7	26.4	62.2	49.3	68.4	64.1	74.0	75.0	51.7	72.7	42.5	67.2	55.7	62.9
Our front end	**82.2**	**37.4**	**72.7**	**57.1**	**62.7**	**82.8**	**77.8**	**78.9**	**28**	**70**	**51.6**	**73.1**	**72.8**	**81.5**	**79.1**	**56.6**	**77.1**	**49.9**	**75.3**	**60.9**	**67.6**

Table 2: Our front-end prediction module is simpler and more accurate than prior models. This table reports accuracy on the VOC-2012 test set.

5. Experiment

- Pascal VOC 2012 image와 Microsoft COCO dataset으로 훈련된 front end module은 VOC 2012 validation set에서 69.8% mean IoU와 test set에서 71.3% mean IoU를 보였습니다. 논문에서는 이 모듈의 높은 정확도를 image classication용으로 만들어진 모델의 필요없는 부분들(중간 layer의 padding, 마지막 두 pooling layer)을 제거함으로 얻을 수 있었다고 합니다.
- Context Module의 성능 평가



(a) Image (b) Front end (c) +Context (d) +CRF-RNN (e) Ground truth

Figure 3: Semantic segmentations produced by different models. From left to right: (a) input image, (b) prediction by the front-end module, (c) prediction by the large context network plugged into the front end, (d) prediction by the front end + context module + CRF-RNN, (e) ground truth.

	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean IoU
Front end	86.3	38.2	76.8	**66.8**	63.2	87.3	78.7	82	33.7	76.7	53.5	73.7	76	76.6	83	**51.9**	77.8	44	79.9	**66.3**	69.8
Front + Basic	86.4	37.6	78.5	66.3	64.1	89.9	79.9	84.9	**36.1**	79.4	**55.8**	77.6	81.6	79	83.1	51.2	81.3	43.7	82.3	65.7	71.3
Front + Large	**87.3**	**39.2**	**80.3**	65.6	**66.4**	**90.2**	**82.6**	**85.8**	34.8	**81.9**	51.7	**79**	**84.1**	**80.9**	**83.2**	51.2	**83.2**	**44.7**	**83.4**	65.6	**72.1**
Front end + CRF	89.2	38.8	80	**69.8**	63.2	88.8	80	85.2	33.8	80.6	55.5	77.1	80.8	77.3	84.3	**53.1**	80.4	45	80.7	**67.9**	71.6
Front + Basic + CRF	89.1	38.7	81.4	67.4	65	91	81	86.7	**37.5**	81	**57**	79.6	83.6	79.9	**84.6**	52.7	83.3	44.3	82.6	67.2	72.7
Front + Large + CRF	**89.6**	**39.9**	**82.7**	66.7	**67.5**	**91.1**	**83.3**	**87.4**	36	**83.3**	52.5	**80.7**	**85.7**	**81.8**	84.4	52.6	**84.4**	**45.3**	**83.7**	66.7	**73.3**
Front end + RNN	88.8	38.1	80.8	**69.1**	65.6	89.9	79.6	85.7	36.3	83.6	57.3	77.9	83.2	77	**84.6**	**54.7**	82.1	**46.9**	80.9	66.7	72.5
Front + Basic + RNN	89	38.4	82.3	67.9	65.2	91.5	80.4	87.2	**38.4**	82.1	**57.7**	79.9	85	79.6	84.5	53.5	84	45	82.8	66.2	73.1
Front + Large + RNN	**89.3**	**39.2**	**83.6**	67.2	**69**	**92.1**	**83.1**	**88**	**38.4**	**84.8**	55.3	**81.2**	**86.7**	**81.3**	84.3	53.6	**84.4**	45.8	**83.8**	**67**	**73.9**

(b) front end module 만을 사용한 segmentation, (c)는 front end에 context module을 더한 것, (d) front end + context module + CRF-RNN(structured prediction)

위 실험은 context module과 structured prediction은 시너지 효과가 있다는 것을 시사하며, 동시에 context module은 structured prediction의 유무와 관계없이 accuracy를 증가시킨다.

6. Conclusion

Dense prediction을 위해서는 모델이 고해상도 output을 산출할 필요가 있는데, 이를 위해서는 중간 과정에서도 어느 정도의 해상도를 유지할 필요가 있습니다. 이 논문은 그것이 가능할 뿐만 아니라 그래야만 한다고 주장합니다. DilatedNet의 중심내용인 dilated convolution은 해상도와 coverage를 잃지 않고 receptive field를 확장시킴으로 모델을 dense prediction에 더욱 적합하게 만들었습니다. 이 dilated convolution으로 구성된 context module은 여러 기존의 segmentation task에 plug in해서 사용했을 때 accuracy를 증가시켜주는 효과를 보았습니다. 또한 마지막으로 기존의 classfication task에서 사용된 모델을 adaptation해서 segmentation을 하고 있는 모델들은 필요없는 부분을 버리고 구조를 단순화 함으로써 accuracy가 증가하였습니다.

References

https://arxiv.org/abs/1511.07122

https://meissa.tistory.com/39

https://deep-learning-study.tistory.com/664