Multi-Scale Context Aggregation by Dilated Convolutions

Yi-An Lai
2018.03.07

http://goo.gl/XLtqZB
Outline

● Situations and Results (4)
● Dilated, Explained (3)
● Context Module and the Front End (6)
● Show and Justify (8)
● Extensions (2)
S1: Dense Prediction?

- One pixel, one prediction: **Pixel-wise**
  - Segmentation, surface normal, depth estimation, style transfer

- Common ConvNets **NOT** tailored for this
  - LeNet, AlexNet, VGG: **image-wise** classifications

“—”: Redundant components?
“+”: Better networks?

S2: What we want

- Customized architecture, tailored module
  - Full-resolution output

- **Multiscale** contexts: receptive field (RF)
  - LeNet, et al. — Stacking + Pooling / Sub-sampling — RF↑
  - Conflicts: **multiscale** vs **dense prediction**

https://guillaumebrg.wordpress.com/2016/02/13/adopting-the-vgg-net-approach-more-layers-smaller-filters/
S3: What others did

- Noh et al. — Encoder-Decoder
  - Heavy downsampling necessary?

- Farabet et al. — Rescaled images
  - Separate handling necessary?
S4: What we’ve done

- Why not modify architecture
  - Replace Fully-connected with Conv (previous works)
  - Remove redundant components — last 2 pooling

- Why not invent new module
  - Multiscale-capable, resolution-preserving

D1: Dilated Conv

- Vanilla Conv:
  \[(F \ast k)(p) = \sum_{s+t=p} F(s) k(t).\]

- \(l\)-Dilated Conv: filter element spacing
  \[(F \ast_l k)(p) = \sum_{s+l|t=p} F(s) k(t).\]

Q1: What’s the RF here?
D2: Dilated C

- Red: receptive field
- F1: dilation = 1
- F2: dilation = 2
- F3: dilation = 4
- **Exponential RF** as stacking with dilations 1, 2, 4, 8, ...
  \[ F_{i+1} = F_i \ast_{2^i} k_i \quad \text{for} \quad i = 0, 1, \ldots, n - 2. \]
- (i+1)th layer RF:
  \[ (2^{i+2} - 1) \times (2^{i+2} - 1) \]
  (Whiteboard explain)
- **Multiscale + full resolution!**
D3: WaveNet - 1D Dilated Conv

- Causal/Online version

Receptive field

http://goo.gl/XLtqZB

https://deepmind.com/blog/wavenet-generative-model-raw-audio/
So Far, So Good?
C1: The Module

- **Goal:** performance↑ using multiscale contexts
- **All Dilated Conv = Multiscale + Full-resolution**
- **(W, H, C) in, (W, H, C) out → Plug into any layer!**
  - Equip feature maps with multiscale contextual info
- **Module:**
  - 1 - 7: + ReLU
  - Equal #channel

Q2: Why stop RF expansion at layer 6?

![Image](https://leonardoaraujosantos.gitbooks.io/artificial-intelligence/content/convolutional_neural_networks.html)
C2: Initializations - inspiration

- RNN initialized with **identity** matrix + ReLU (Red) → Beat LSTM

"A Simple Way to Initialize Recurrent Networks of Rectified Linear Units." QV Le, et al

Image: [http://speech.ee.ntu.edu.tw/~tlkagk/courses_ML17_2.html](http://speech.ee.ntu.edu.tw/~tlkagk/courses_ML17_2.html)
C3: Initializations

- Basic: #channel the same
  - Identity initializations → "copy & paste"

- Large: #channel double
  - Copy & paste but with scaling
  - All other zeros → break symmetry
    → Randomly initialized and small (<< 0.5)

C4: Front End - Modified VGG 16

1. Input 900 x 900 x 3 (reflection padding)
C4: Front End - Modified VGG 16

1. Input 900 x 900 x 3 (reflection padding)

2. No padding in the middle at all
C4: Front End - Modified VGG 16

1. Input 900 x 900 x 3 (reflection padding)

2. No padding in the middle at all

3. Last 2 pooling $\rightarrow$ dilated
1. Input 900 x 900 x 3 (reflection padding)

2. No padding in the middle at all

3. Last 2 pooling → dilated

4. Fully Connected → Conv

Dilation: 2
Kernal: 7 x 7

Dilation: 4

C=4096
C=4096
C=21
C=21
C4: Front End - Modified VGG 16

1. Input 900 x 900 x 3 (reflection padding)

2. No padding in the middle at all

3. Last 2 pooling → dilated

4. Fully Connected → Conv

5. Softmax: 21 classes → 64 x 64 x 21

Prediction: upsample to original size (500 x 500)
C4: Front End - VGG 16

1. Input 900 x 900 x 3 (reflection padding)

2. No padding in the middle at all

3. Last 2 pooling → dilated

4. Fully Connected → Conv

5. Softmax: 21 classes → 64 x 64 x 21

Prediction: upsample to original size (500 x 500)

Q3: Where did authors plug in the module?
   A. after blue 1
   B. after blue 2
   C. after blue 3
   D. after brown
C5: Front End Performance

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.

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

→ Modified architecture performs good!
C6: Front Line

More accurate

More details
S1: Training

- **VOC 2012 + COCO**
  - COCO: images with at least 1 class in VOC
  - 2-stage: (VOC + COCO) → VOC
  - Front End **only**: 71.3% test → Our modifications justified

- **Module: context aggre.**
  - Add padding for inputs (64 x 64 x C), why? → RF 67 x 67, to output 64 x 64 x C
  - Train Front End → Fixed → Train module (Joint ~ the same)
  - Goal: Testing the module + Pairing with **structured prediction** (Could be jointly learned)
Warning of New Stuff!
S2: Wait, structured prediction?

- Inputs and outputs: objects with **structures**
  - Object: sequence, boxes, images, list, tree…
  - Need more powerful $F$ to define “compatibility” of $(X, Y)$
S2: Wait, structured prediction?

- Inputs and outputs: objects with **structures**
  - *Object*: sequence, boxes, images, list, tree…
  - Need more powerful $F$ to define “compatibility” of $(X, Y)$

\[ f: X \rightarrow Y \]

- Retrieval

  "Machine learning" (keyword)

  ![A list of web pages (Search Result)](image)

[http://speech.ee.ntu.edu.tw/~tlkagk/courses_MLSD_15_2.html](http://speech.ee.ntu.edu.tw/~tlkagk/courses_MLSD_15_2.html)
S2: Wait, structured prediction?

- Inputs and outputs: objects with structures
  - Object: sequence, boxes, images, list, tree...
  - Need more powerful $F$ to define “compatibility” of $(X, Y)$

Retrieval

```
X:  "Machine learning" (keyword)  Y:  A list of web pages (Search Result)
```

Speech

```
X:  (One kind of sequence)  Y:  “大家好，歡迎大家來修
  機器學習及其結構化” (Another kind of sequence)
```

http://speech.ee.ntu.edu.tw/~tlkagk/courses_MLSD_15_2.html
S2: Wait, structured prediction?

- Inputs and outputs: objects with **structures**
  - **Object**: sequence, boxes, images, list, tree…
  - Need more powerful $F$ to define “compatibility” of $(X, Y)$

- **Retrieval**
  - $X$: "Machine learning" (keyword)
  - $Y$: A list of web pages (Search Result)

- **Detection**
  - $X$: Image
  - $Y$: Object Positions

- **Speech**
  - $X$: (One kind of sequence)
  - $Y$: “大家好，歡迎大家來修機器學習及其深層與結構化” (Another kind of sequence)

[http://speech.ee.ntu.edu.tw/~tlkagk/courses_MLSD_15_2.html](http://speech.ee.ntu.edu.tw/~tlkagk/courses_MLSD_15_2.html)
S2: Wait, structured prediction?

- Inputs and outputs: objects with **structures**
  - **Object**: sequence, boxes, images, list, tree…
  - Need more powerful $F$ to define “compatibility” of $(X, Y)$

- **Retrieval**
  - $X$: “Machine learning” (keyword)
  - $Y$: A list of web pages (Search Result)

- **Detection**
  - $X$: Image
  - $Y$: Object Positions

- **Speech**
  - $X$: (One kind of sequence)
  - $Y$: “大家好，歡迎大家來修機器學習及其深層與結構化” (Another kind of sequence)

- **Pose**
  - $X$: Image
  - $Y$: Pose

http://speech.ee.ntu.edu.tw/~tlkagk/courses_MLSD_15_2.html
S3: How?

- Structured prediction in only 3 steps
- As easy as putting an elephant into a refrigerator
S4: 3 Steps

1. **Evaluation**: how does “compatibility” $F(X, Y)$ look like?

   - **Object Detection**: $F(x=\text{人物图}, y=\text{框})$
   - **Summarization**: $F(x=\text{一篇长文}, y=\text{一篇短文})$

   [http://speech.ee.ntu.edu.tw/~tlkagk/courses_MLSD15_2.html](http://speech.ee.ntu.edu.tw/~tlkagk/courses_MLSD15_2.html)

   Energy-based model: [https://cs.nyu.edu/~yann/research/ebm/](https://cs.nyu.edu/~yann/research/ebm/)
S4: 3 Steps

1. **Evaluation**: how does “compatibility” $F(X, Y)$ look like?

   - **Object Detection**: $F(x=\text{[image]}, y=\text{[object]})$
   - **Summarization**: $F(x=\text{[long document]}, y=\text{[summary]})$  

2. **Inference**: Solve “arg max” problem
   - Detection $Y$: all possible bounding boxes
   - Summarization $Y$: all combinations of sentences in a doc

http://speech.ee.ntu.edu.tw/~tlkagk/courses_MLSD15_2.html  
**Energy-based model**: https://cs.nyu.edu/~yann/research/ebm/
S4: 3 Steps

1. **Evaluation**: how does “compatibility” $F(X, Y)$ look like?

   - **Object Detection**: $F(x=\text{}, y=\text{})$
   - **Summarization**: $F(x=\text{}, y=\text{})$

2. **Inference**: Solve “arg max” problem
   - Detection $Y$: all possible bounding boxes
   - Summarization $Y$: all combinations of sentences in a doc

3. **Training**: Given data $(x_i, y_i)$, learn $F$ such that

   - $F(x^1, y^1)$ for all $y \neq \hat{y}^1$
   - $F(x^2, y)$ for all $y \neq \hat{y}^2$
   - $F(x^r, y)$ for all $y \neq \hat{y}^r$

   ![Diagram](http://speech.ee.ntu.edu.tw/~tlkagk/courses_MLSD15_2.html)

   Energy-based model: [https://cs.nyu.edu/~yann/research/ebm/](https://cs.nyu.edu/~yann/research/ebm/)
S5: Structured predictions used

- **Goal**: refine details to improve performance

- **CRF**: [Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials, Krähenbühl, et al. (2011)]
  - Pairwise links between pixels
  - Capture fine edge details with long range dependencies

- **RNN-CRF**: [Conditional Random Fields as Recurrent Neural Networks, Zheng, et al. (2015)]
  - Interpret CRFs as RNN
  - Can be plug into CNNs and jointly learned

- **Q4**: What are our inputs and outputs for structured prediction here?

- More about Conditional Random Fields (CRFs)
  - Machine Learning: A Probabilistic Perspective
  - [http://dontloo.github.io/blog/CRF/](http://dontloo.github.io/blog/CRF/)
S6: Evaluate Module

Effective & Synergistic

<table>
<thead>
<tr>
<th></th>
<th>acro</th>
<th>bike</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>table</th>
<th>dog</th>
<th>horse</th>
<th>mbike</th>
<th>person</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>mean IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Front end</strong></td>
<td>86.3</td>
<td>38.2</td>
<td>76.8</td>
<td><strong>66.8</strong></td>
<td>63.2</td>
<td>87.3</td>
<td>78.7</td>
<td>82</td>
<td>33.7</td>
<td>76.7</td>
<td>53.5</td>
<td>73.7</td>
<td>76</td>
<td>76.6</td>
<td>83</td>
<td>77.8</td>
<td>44</td>
<td>79.9</td>
<td>66.3</td>
<td>69.8</td>
<td></td>
</tr>
<tr>
<td><strong>Front + Basic</strong></td>
<td>86.4</td>
<td>37.6</td>
<td>78.5</td>
<td>66.3</td>
<td>64.1</td>
<td>89.9</td>
<td>79.9</td>
<td>84.9</td>
<td><strong>36.1</strong></td>
<td>79.4</td>
<td>55.8</td>
<td>77.6</td>
<td>81.6</td>
<td>79</td>
<td>84.1</td>
<td>80.9</td>
<td>83.2</td>
<td>44.7</td>
<td>83.4</td>
<td>65.6</td>
<td>71.3</td>
</tr>
<tr>
<td><strong>Front + Large</strong></td>
<td><strong>87.3</strong></td>
<td><strong>39.2</strong></td>
<td><strong>80.3</strong></td>
<td>65.6</td>
<td><strong>66.4</strong></td>
<td><strong>90.2</strong></td>
<td><strong>82.6</strong></td>
<td><strong>85.8</strong></td>
<td>34.8</td>
<td>81.9</td>
<td>51.7</td>
<td>79</td>
<td>84.1</td>
<td>80.9</td>
<td>83.2</td>
<td>51.2</td>
<td>83.2</td>
<td>44.7</td>
<td>83.4</td>
<td>65.6</td>
<td><strong>72.1</strong></td>
</tr>
<tr>
<td><strong>Front end + CRF</strong></td>
<td>89.2</td>
<td>38.8</td>
<td>80</td>
<td><strong>69.8</strong></td>
<td>63.2</td>
<td>88.8</td>
<td>80</td>
<td>85.2</td>
<td>33.8</td>
<td>80.6</td>
<td>55.5</td>
<td>77.1</td>
<td>80.8</td>
<td>77.3</td>
<td>84.3</td>
<td>53.1</td>
<td>80.4</td>
<td>45</td>
<td>80.7</td>
<td><strong>67.9</strong></td>
<td>71.6</td>
</tr>
<tr>
<td><strong>Front + Basic + CRF</strong></td>
<td>89.1</td>
<td>38.7</td>
<td>81.4</td>
<td>67.4</td>
<td>65</td>
<td>91</td>
<td>81</td>
<td>86.7</td>
<td>37.5</td>
<td>81</td>
<td>57</td>
<td>79.6</td>
<td>83.6</td>
<td>79.9</td>
<td>84.6</td>
<td>52.1</td>
<td>83.3</td>
<td>43.3</td>
<td>82.6</td>
<td>67.2</td>
<td><strong>72.7</strong></td>
</tr>
<tr>
<td><strong>Front + Large + CRF</strong></td>
<td><strong>89.6</strong></td>
<td><strong>39.9</strong></td>
<td><strong>82.7</strong></td>
<td>66.7</td>
<td><strong>67.5</strong></td>
<td><strong>91.1</strong></td>
<td><strong>83.3</strong></td>
<td><strong>87.4</strong></td>
<td>36</td>
<td><strong>83.3</strong></td>
<td><strong>52.5</strong></td>
<td><strong>80.7</strong></td>
<td><strong>85.7</strong></td>
<td><strong>81.8</strong></td>
<td><strong>84.4</strong></td>
<td><strong>52.6</strong></td>
<td><strong>84.4</strong></td>
<td><strong>45.3</strong></td>
<td><strong>83.7</strong></td>
<td><strong>66.7</strong></td>
<td><strong>73.3</strong></td>
</tr>
<tr>
<td><strong>Front end + RNN</strong></td>
<td>88.8</td>
<td>38.1</td>
<td>80.8</td>
<td><strong>69.1</strong></td>
<td>65.6</td>
<td>89.9</td>
<td>79.6</td>
<td>85.7</td>
<td><strong>36.3</strong></td>
<td>83.6</td>
<td>57.3</td>
<td>77.9</td>
<td>83.2</td>
<td>77</td>
<td><strong>84.6</strong></td>
<td><strong>54.7</strong></td>
<td><strong>82.1</strong></td>
<td><strong>46.9</strong></td>
<td><strong>80.9</strong></td>
<td><strong>66.7</strong></td>
<td>72.5</td>
</tr>
<tr>
<td><strong>Front + Basic + RNN</strong></td>
<td>89</td>
<td>38.4</td>
<td>82.3</td>
<td>67.9</td>
<td>65.2</td>
<td>91.5</td>
<td>80.4</td>
<td>87.2</td>
<td><strong>38.4</strong></td>
<td><strong>82.1</strong></td>
<td><strong>57.7</strong></td>
<td><strong>79.9</strong></td>
<td>85</td>
<td>79.6</td>
<td>84.5</td>
<td>53.5</td>
<td>84</td>
<td>45</td>
<td>82.8</td>
<td>66.2</td>
<td><strong>73.1</strong></td>
</tr>
<tr>
<td><strong>Front + Large + RNN</strong></td>
<td><strong>89.3</strong></td>
<td><strong>39.2</strong></td>
<td><strong>83.6</strong></td>
<td>67.2</td>
<td><strong>69</strong></td>
<td><strong>92.1</strong></td>
<td><strong>83.1</strong></td>
<td><strong>88</strong></td>
<td><strong>38.4</strong></td>
<td><strong>84.8</strong></td>
<td><strong>55.3</strong></td>
<td><strong>81.2</strong></td>
<td><strong>86.7</strong></td>
<td><strong>81.3</strong></td>
<td><strong>84.3</strong></td>
<td><strong>53.6</strong></td>
<td><strong>84.4</strong></td>
<td><strong>45.8</strong></td>
<td><strong>83.8</strong></td>
<td><strong>67</strong></td>
<td><strong>73.9</strong></td>
</tr>
</tbody>
</table>

Table 3: Controlled evaluation of the effect of the context module on the accuracy of three different architectures for semantic segmentation. Experiments performed on the VOC-2012 validation set.
S7: Now the real test

- Yellow: **off** by a margin
- Green: **over** by a margin

Table 4: Evaluation on the VOC-2012 test set.

<table>
<thead>
<tr>
<th>Method</th>
<th>aero</th>
<th>bike</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>table</th>
<th>dog</th>
<th>horse</th>
<th>mbike</th>
<th>person</th>
<th>plant</th>
<th>sheep</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>mean IoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepLab++</td>
<td>89.1</td>
<td>38.3</td>
<td>88.1</td>
<td>63.3</td>
<td>69.7</td>
<td>87.1</td>
<td>85</td>
<td>29.3</td>
<td>76.5</td>
<td>56.5</td>
<td>79.8</td>
<td>77.9</td>
<td>85.8</td>
<td>82.4</td>
<td>57.4</td>
<td>84.3</td>
<td>54.9</td>
<td>80.5</td>
<td>64.1</td>
<td>72.7</td>
<td>73.9</td>
</tr>
<tr>
<td>DeepLab-MSc++</td>
<td>89.2</td>
<td>46.7</td>
<td>88.5</td>
<td>63.5</td>
<td>68.4</td>
<td>87.0</td>
<td>81.2</td>
<td>86.3</td>
<td>32.6</td>
<td>80.7</td>
<td>62.4</td>
<td>81.0</td>
<td>81.3</td>
<td>84.3</td>
<td>82.1</td>
<td>56.2</td>
<td>84.6</td>
<td>58.3</td>
<td>76.2</td>
<td>67.2</td>
<td>73.9</td>
</tr>
<tr>
<td>CRF-RNN</td>
<td>90.4</td>
<td><strong>55.3</strong></td>
<td>88.7</td>
<td><strong>68.4</strong></td>
<td>69.8</td>
<td>88.3</td>
<td>82.4</td>
<td>85.1</td>
<td>32.6</td>
<td>78.5</td>
<td>64.4</td>
<td>79.6</td>
<td>81.9</td>
<td>86.4</td>
<td>81.8</td>
<td><strong>58.6</strong></td>
<td>82.4</td>
<td>53.5</td>
<td>77.4</td>
<td>70.1</td>
<td><strong>74.7</strong></td>
</tr>
<tr>
<td>Front end</td>
<td>86.6</td>
<td>37.3</td>
<td>84.9</td>
<td>62.4</td>
<td>67.3</td>
<td>86.2</td>
<td>82.1</td>
<td>32.6</td>
<td>77.4</td>
<td>58.3</td>
<td>75.9</td>
<td><strong>81</strong></td>
<td>83.6</td>
<td>82.3</td>
<td>54.2</td>
<td>81.5</td>
<td>50.1</td>
<td>77.5</td>
<td>63</td>
<td>71.3</td>
<td>71.3</td>
</tr>
<tr>
<td>Context</td>
<td>89.1</td>
<td>39.1</td>
<td>86.8</td>
<td>62.6</td>
<td>68.9</td>
<td>88.2</td>
<td>82.6</td>
<td>87.7</td>
<td>33.8</td>
<td>81.2</td>
<td>59.2</td>
<td>81.8</td>
<td>87.2</td>
<td>83.3</td>
<td>53.6</td>
<td>84.9</td>
<td>53.7</td>
<td>80.5</td>
<td>62.9</td>
<td>71.3</td>
<td>73.5</td>
</tr>
<tr>
<td>Context + CRF</td>
<td>91.3</td>
<td>39.9</td>
<td><strong>88.9</strong></td>
<td>64.3</td>
<td>69.8</td>
<td>88.9</td>
<td>82.6</td>
<td>89.7</td>
<td>34.7</td>
<td>82.7</td>
<td>59.5</td>
<td>83</td>
<td>88.4</td>
<td>84.2</td>
<td>85</td>
<td>55.3</td>
<td>86.7</td>
<td>54.4</td>
<td><strong>81.9</strong></td>
<td>63.6</td>
<td><strong>74.7</strong></td>
</tr>
<tr>
<td>Context + CRF-RNN</td>
<td><strong>91.7</strong></td>
<td>39.6</td>
<td>87.8</td>
<td>63.1</td>
<td><strong>71.8</strong></td>
<td><strong>89.7</strong></td>
<td><strong>82.9</strong></td>
<td><strong>89.8</strong></td>
<td>37.2</td>
<td><strong>84</strong></td>
<td><strong>63</strong></td>
<td><strong>83.3</strong></td>
<td><strong>89</strong></td>
<td><strong>83.8</strong></td>
<td><strong>85.1</strong></td>
<td><strong>56.8</strong></td>
<td><strong>87.6</strong></td>
<td><strong>56</strong></td>
<td><strong>80.2</strong></td>
<td><strong>64.7</strong></td>
<td><strong>75.3</strong></td>
</tr>
</tbody>
</table>

→ Context module boost performance
S8: Other datasets

- Urban Scenes: CamVid, KITTI, Cityscapes
- Adjust #layer in the module w.r.t. input size
- Outperform all prior works
- Code: https://github.com/fyu/dilation
E1: Strength & Weakness

- **Strength**
  - *Multiscale* contextual information
  - Full-resolution context module can be **plugged** into any layer
  - Architecture from classification tasks is modified and justified for segmentation
E1: Strength & Weakness

- **Strength**
  - Multiscale contextual information
  - Full-resolution context module can be plugged into any layer
  - Architecture from classification tasks is modified and justified for segmentation

- **Weakness**
  - Only 64 * 64 output size: lose details → Helped by CRF
  - Full-resolution requires more memory usage
  - Performance $\uparrow$ $\phi \pi$ more parameters rather than multiscale info?
    → No experiment to compare with another module with all vanilla conv
E2: Got ideas?

1. Multi-head **self-attention** rather than dilated conv for multiscale info?

https://pythonmachinelearning.pro/introduction-to-convolutional-neural-networks-for-vision-tasks/
E2: Got ideas?

1. Multi-head **self-attention** rather than dilated conv for multiscale info?

2. Striding as “**learnable**” pooling
   → replace all pooling layers

https://pythonmachinelearning.pro/introduction-to-convolutional-neural-networks-for-vision-tasks/
E2: Got ideas?

1. Multi-head **self-attention** rather than dilated conv for multiscale info?

2. Striding as “**learnable**” pooling
   → replace all pooling layers

3. Other variants:

https://pythonmachinelearning.pro/introduction-to-convolutional-neural-networks-for-vision-tasks/
Than