# Scene Parsing through Per-Pixel Labeling: *a better and faster way*

**Shu Kong**

CS, ICS, UCI

# Image Understanding --> Scene Parsing

UCIrvine
UNIVERSITY OF CALIFORNIA, IRVINE

# Scene Parsing

semantic segmentation

classifying each pixel into one of defined categories

UCIrvine
University of California, Irvine

# Scene Parsing

semantic segmentation (*what&where*)

localization (*where*)

support, surface normal (*relation*)

# Outline

1. Background
2. Attention to Perspective: Depth-aware Pooling Module
3. Recurrent Refining with Perspective Understanding in the Loop
4. Attention to Perspective Again
5. Pixel-wise Attentional Gating (PAG)
6. Pixel-Level Dynamic Routing
7. Conclusion

# Outline

1. Background
2. Attention to Perspective: Depth-aware Pooling Module
3. Recurrent Refining with Perspective Understanding in the Loop
4. Attention to Perspective Again
5. Pixel-wise Attentional Gating (PAG)
6. Pixel-Level Dynamic Routing
7. Conclusion

# Scene Parsing

semantic segmentation

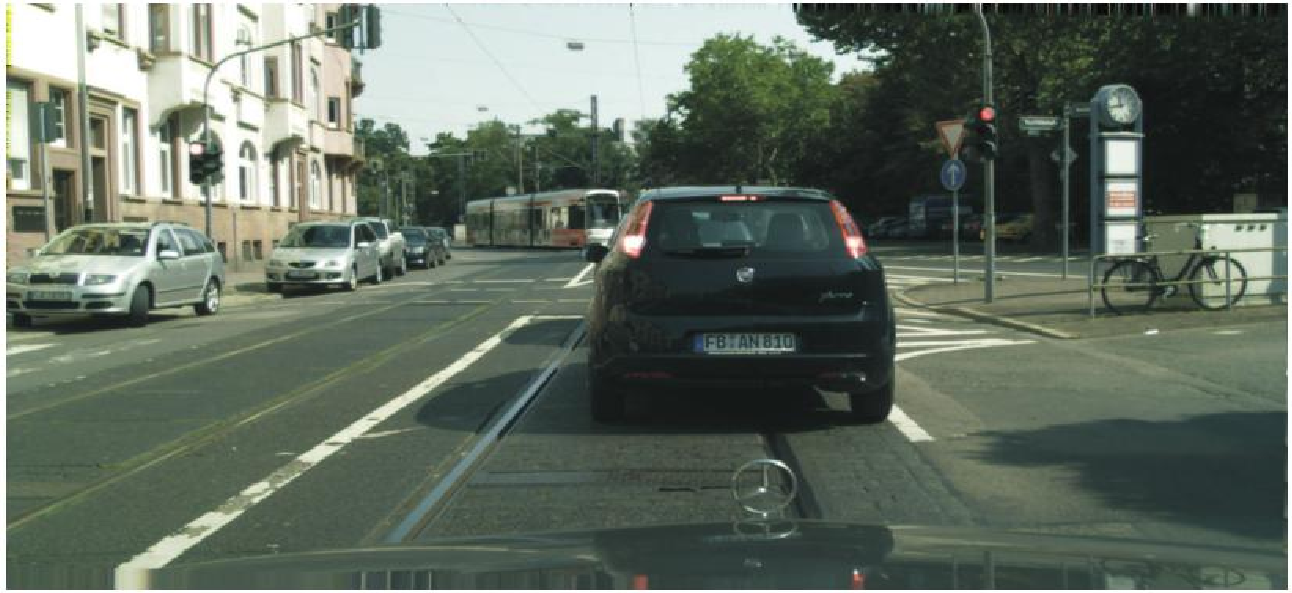
classifying each pixel into one of defined categories

# Scene Parsing from Perspective Image

## large scale variation

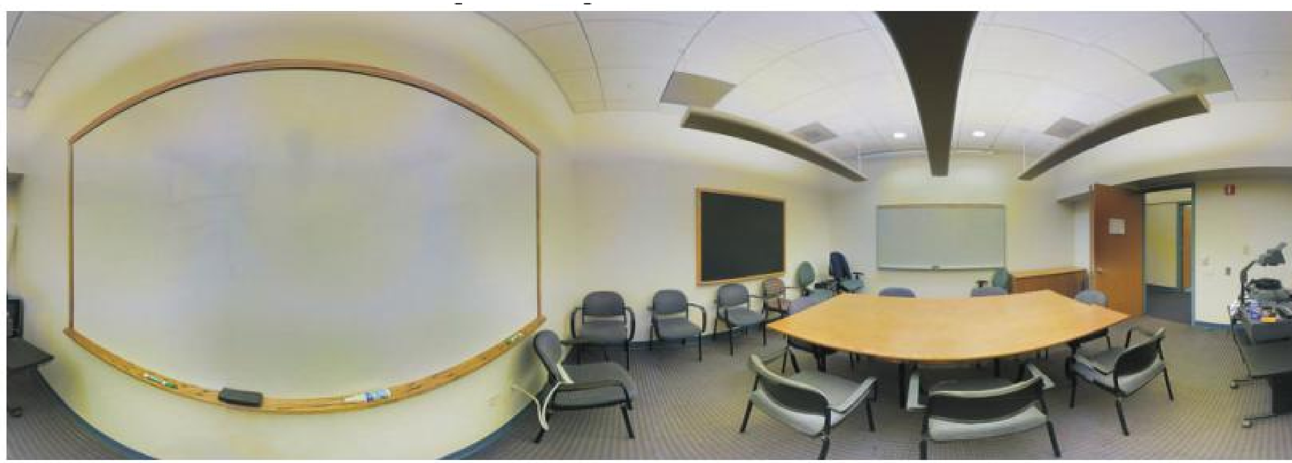
car, pole

car vs. train

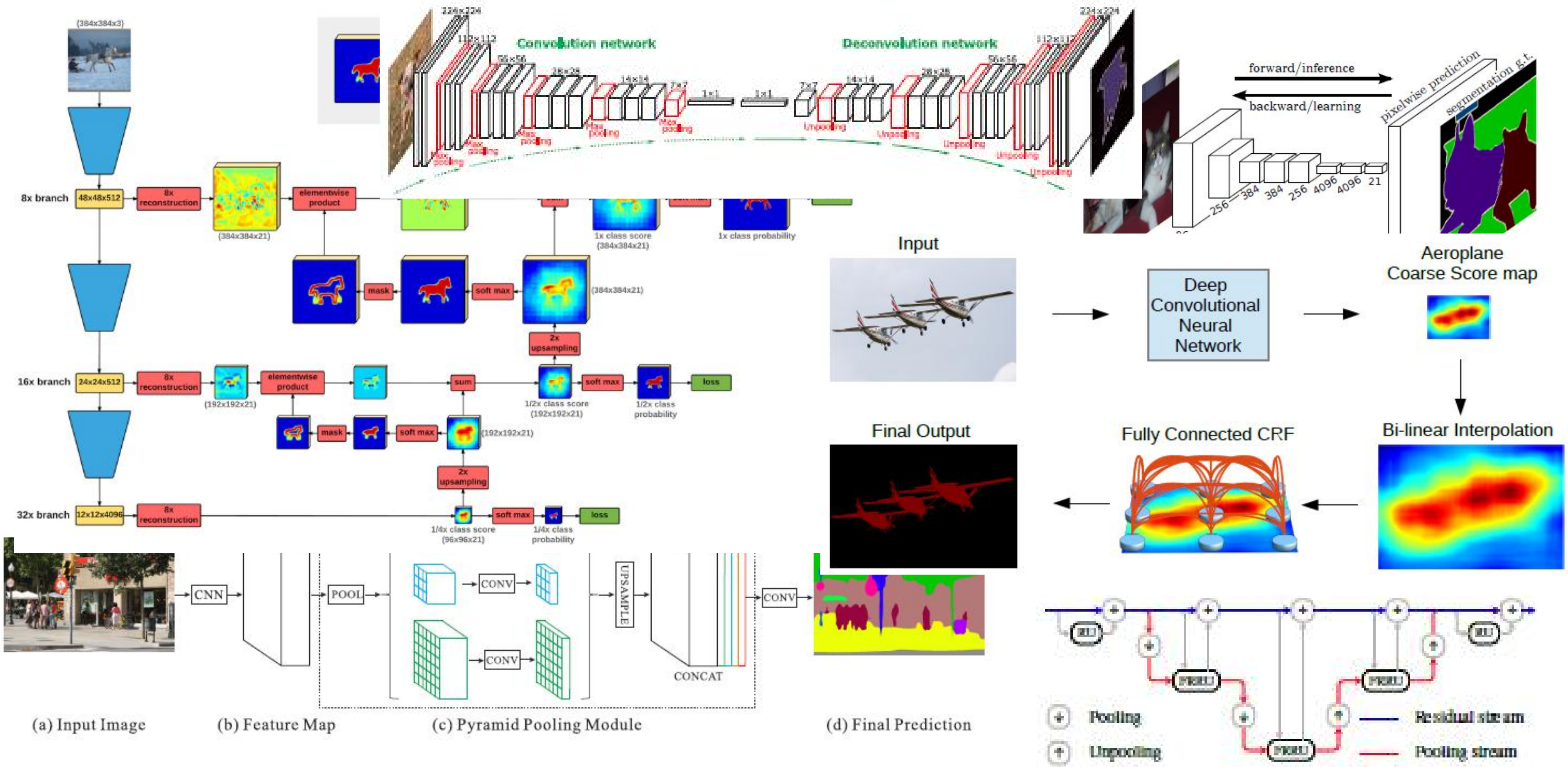


white board, chair

chair vs. white board



# Tons of (Deep) Scene Parser, but...




None of them consider “perspective” explicitly.

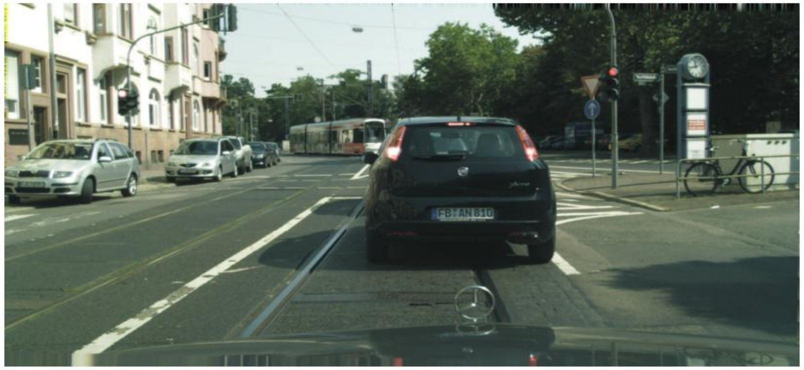
# Outline



UCIrvine
University of California, Irvine

# Attention to Perspective: Depth-aware Pooling

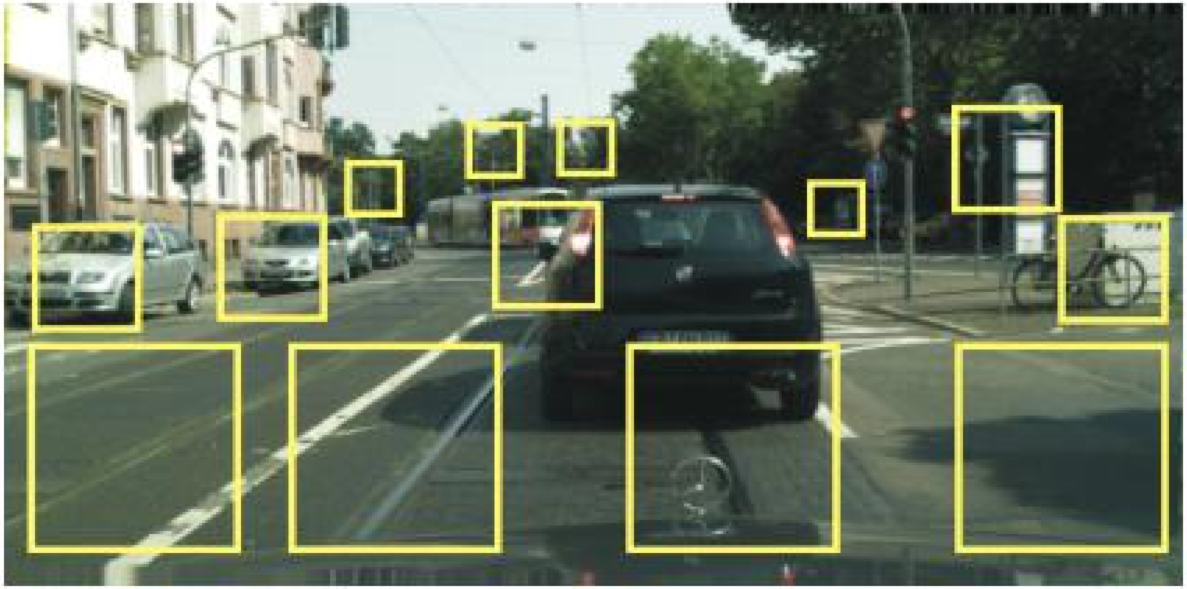
For each pixel, deciding the size of field of view (FoV) to aggregate information



# Attention to Perspective: Depth-aware Pooling

For each pixel, deciding the size of field of view (FoV) to aggregate information

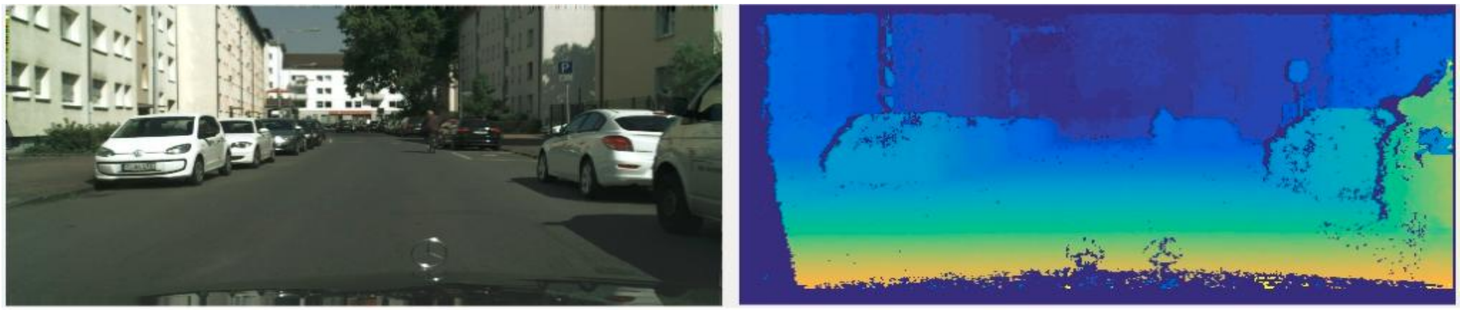
The closer the object is to the camera, the larger size it appears in the image, the larger FoV the network should "pool".



# Depth-aware Pooling Module

Depth conveys the scale information.

The closer the object is to the camera, the larger size it appears in the image, the larger FoV the network should “pool”.



# Depth-aware Pooling Module

How to use depth to choose the FoV size?

# Depth-aware Pooling Module

How to use depth to choose the FoV size?

How about making the pooling size adaptive w.r.t depth?

# Depth-aware Pooling Module

How to use depth to choose the FoV size?

How about making the pooling size adaptive w.r.t depth?

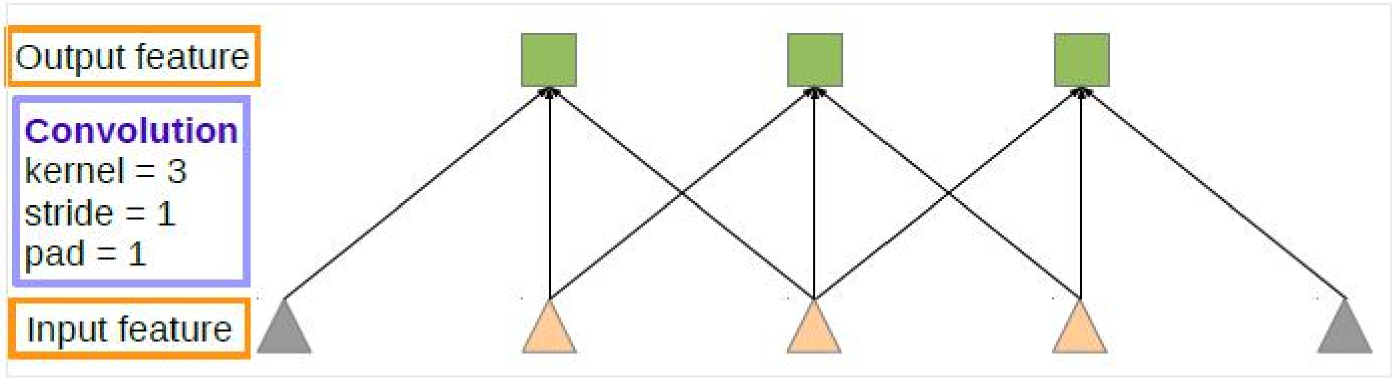
We turn to dilated convolution (Atrous Convolution).

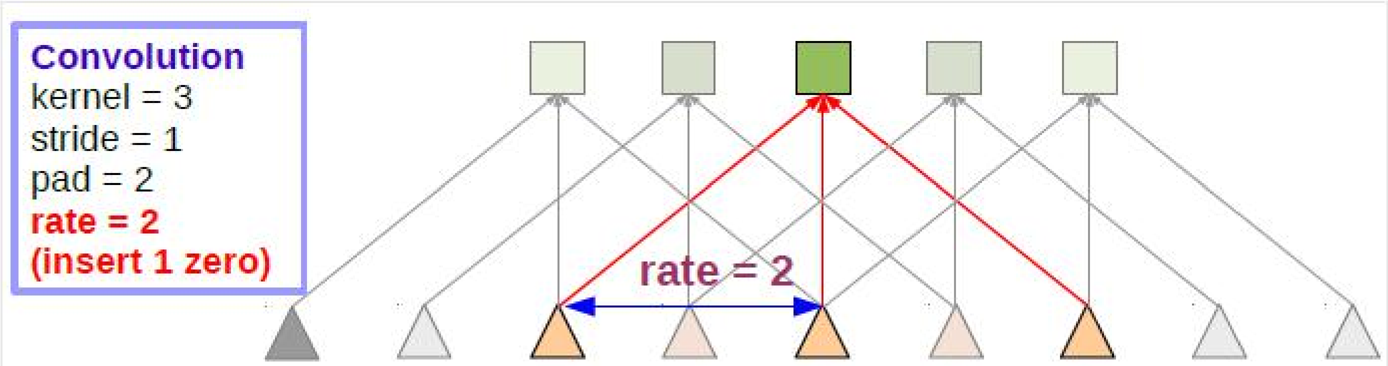
# Depth-aware Pooling Module

Atrous convolution (skipping/inserting zero)

a trous (French) -- holes (English)

$$y[i] = \sum_{k=1}^{K} x[i + r \cdot k] w[k]$$



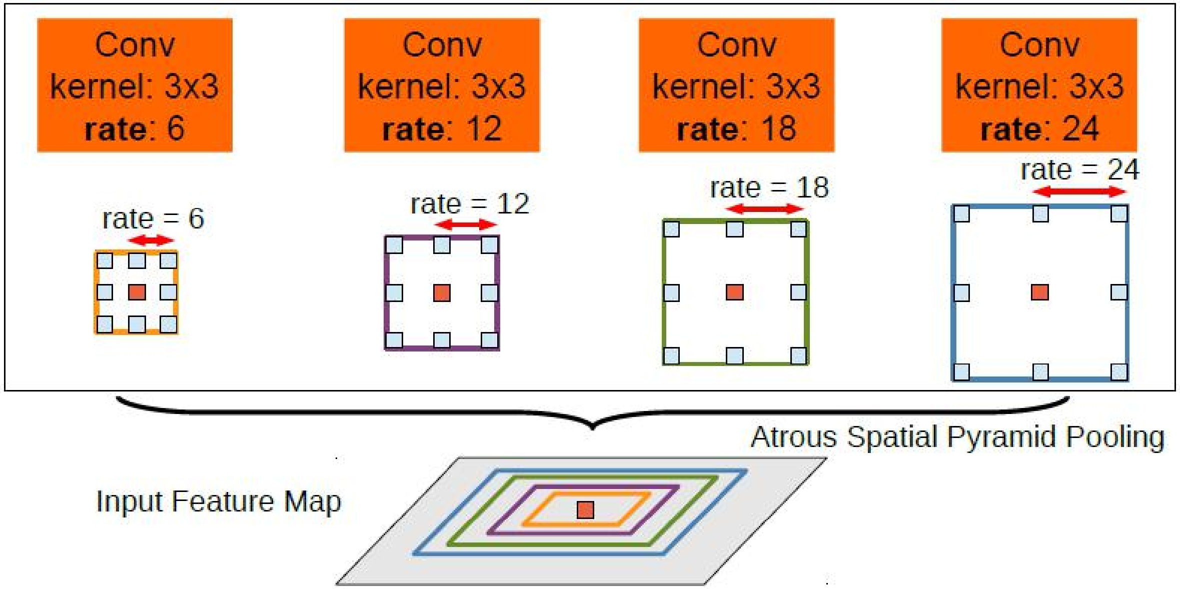






DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs

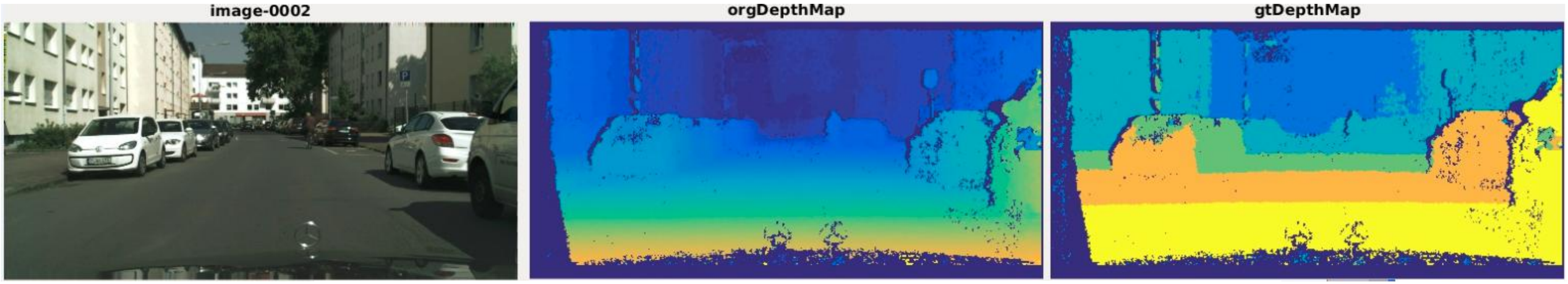
# Depth-aware Pooling Module

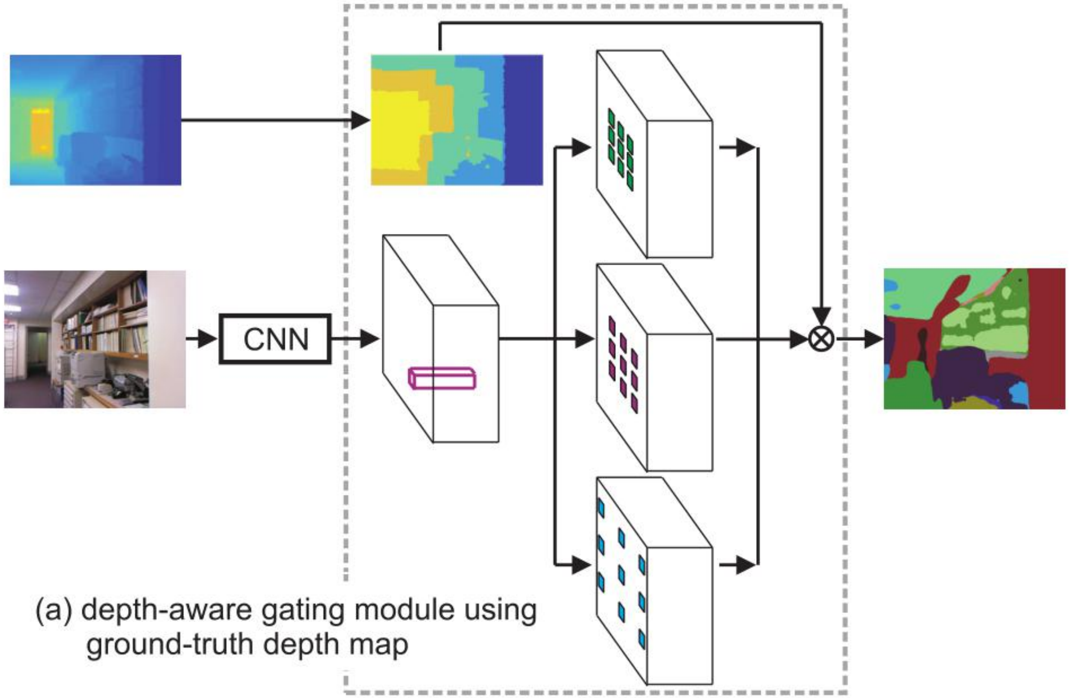
2D atrous convolution of different dilate rates.



# Depth-aware Pooling Module

quantize the depth into five scales with dilate rates {1, 2, 4, 8, 16}



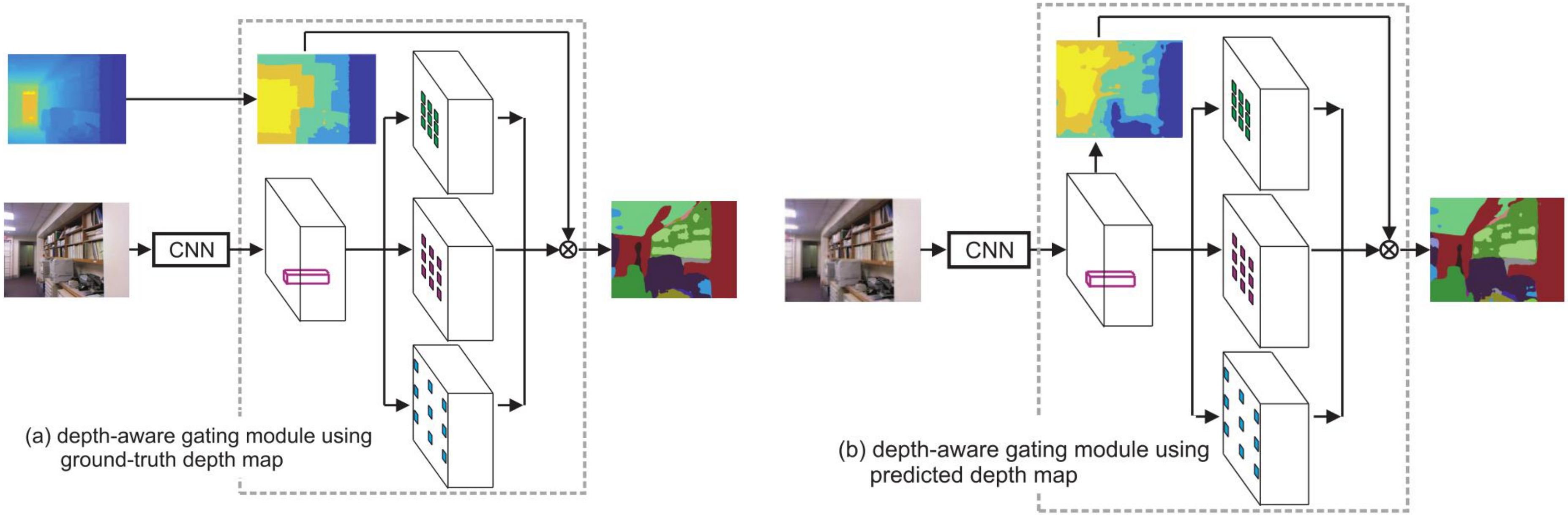

(a) depth-aware gating module using ground-truth depth map

# Depth-aware Pooling Module

Alternatively, learning depth estimator, and testing without depth

quantized depth scale classification

softmax weight for multiplicative gating




(a) depth-aware gating module using ground-truth depth map

(b) depth-aware gating module using predicted depth map



S. Kong, C. Fowlkes, Recurrent Scene Parsing with Perspective Understanding in the Loop, CVPR, 2018

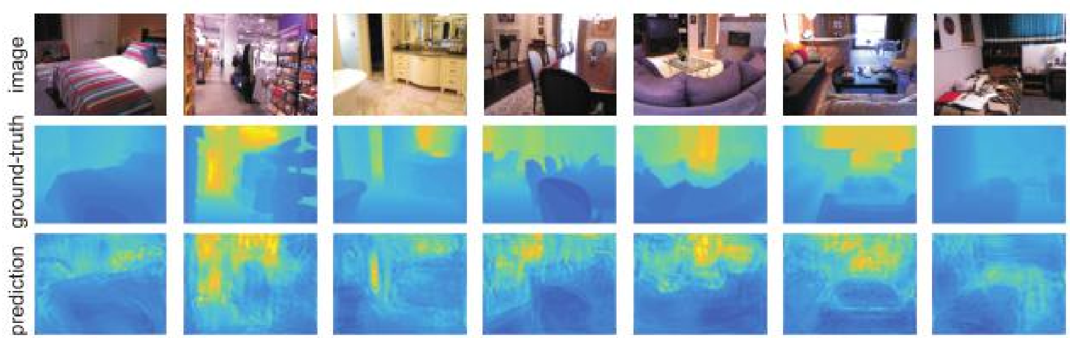
# Depth-aware pooling module

Alternatively, learning depth estimator, and testing without depth

reliable monocular depth estimation

Table 1: Depth prediction on NYU-depth-v2 dataset.

| Metric $\delta <$ | Ladicky [23] | Liu [30] | Eigen [11] | Eigen [10] | Laina [24] | Ours | Ours -blur |
|---|---|---|---|---|---|---|---|
| 1.25 | 0.542 | 0.614 | 0.614 | 0.769 | 0.811 | 0.809 | 0.816 |
| $1.25^2$ | 0.829 | 0.883 | 0.888 | 0.950 | 0.953 | 0.945 | 0.950 |
| $1.25^3$ | 0.940 | 0.971 | 0.972 | 0.988 | 0.988 | 0.986 | 0.989 |

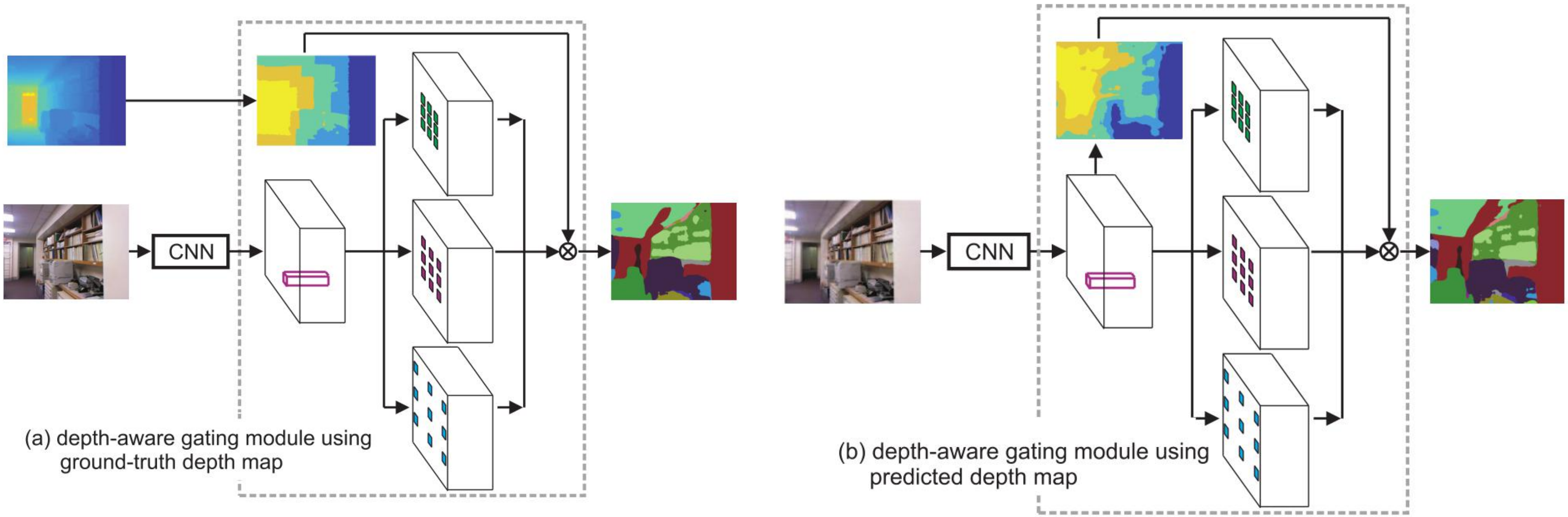



$$\ell_{depthReg}(\mathbf{D}, \mathbf{D}^*) = \frac{1}{|M|} \sum_{(i,j)\in M} \| \log(\mathbf{D}_{ij}) - \log(\mathbf{D}_{ij})^* \|_2^2$$

# Depth-aware pooling module

many possibilities to explore --

1. sharing the parameters in this pooling module (multiPool)
2. averaging the feature vs. attention vs. depth-aware gating
3. MultiPool vs. MultiScale (input)




(a) depth-aware gating module using ground-truth depth map

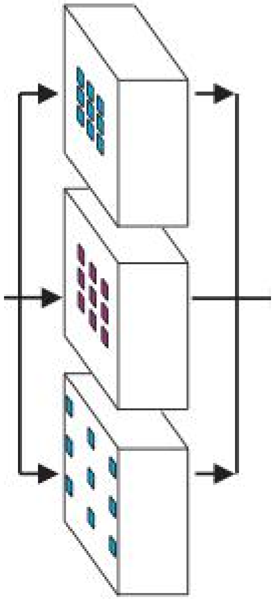
(b) depth-aware gating module using predicted depth map

UCIrvine
University of California, Irvine

# Depth-aware pooling module

many possibilities to explore --

1. sharing the parameters in this pooling module (multiPool)

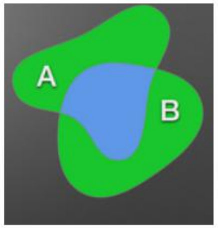


# Depth-aware pooling module

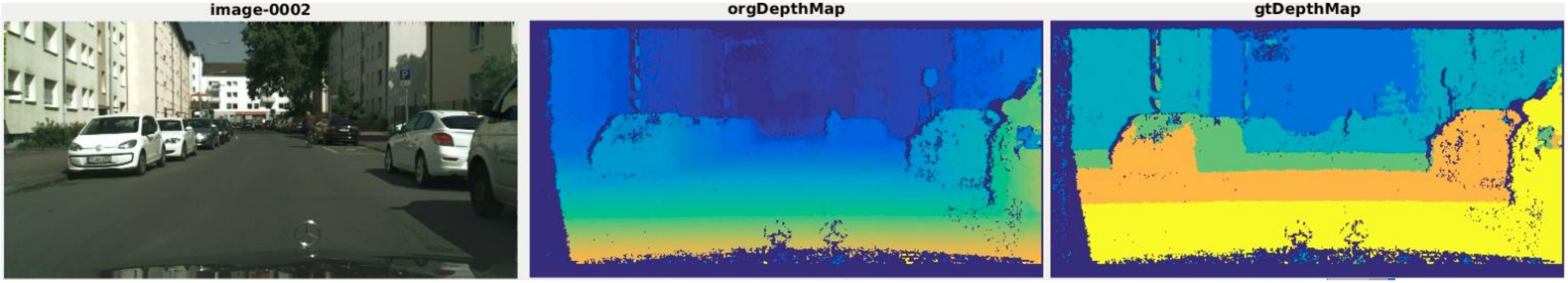
Cityscapes dataset

metric: Intersection over Union (IoU)

$$IOU(A,B) = \frac{|A \cap B|}{|A \cup B|}$$




using the ground-truth disparity map, 5 discete bins for 5 scales {1,2,4,8,16}



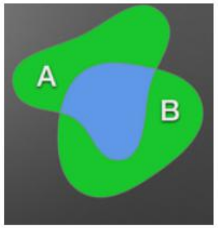

UCIrvine
UNIVERSITY OF CALIFORNIA, IRVINE

# Depth-aware pooling module

Cityscapes dataset

metric: Intersection over Union (IoU)

$$IOU(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

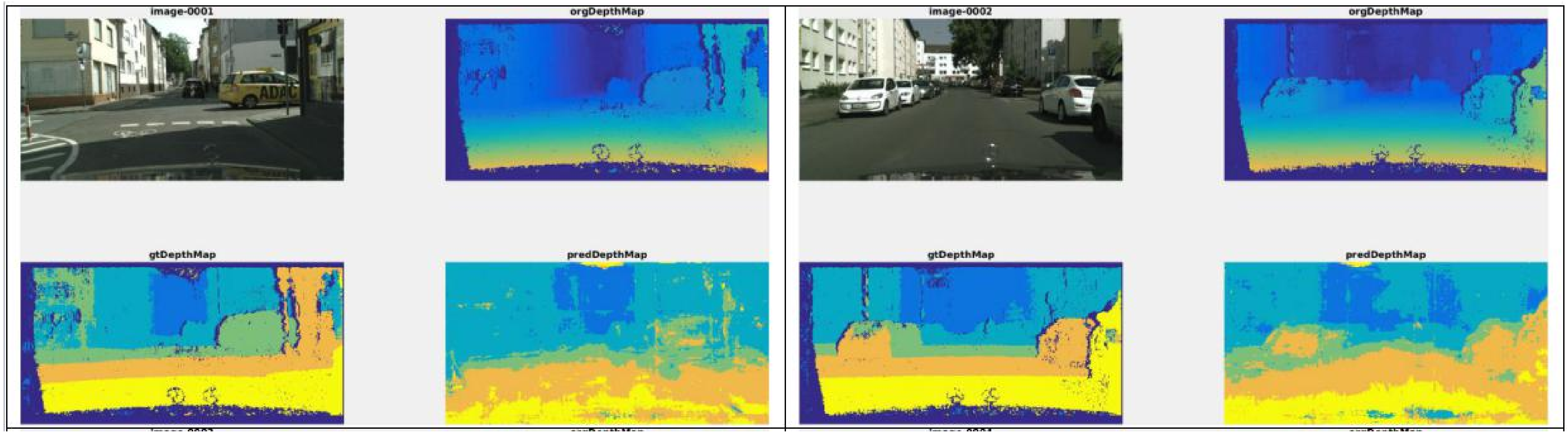



using the ground-truth disparity map, 5 discete bins for 5 scales {1,2,4,8,16}

| | deepLab (baseline) | avg. | gtDepth tiedKernel | gtDepth untied Kernel |
|---|---|---|---|---|
| IoU | 0.738 | 0.747 | 0.748 | 0.753 |

# Depth-aware pooling module

train depth estimation branch to see if the estimated depth also helps



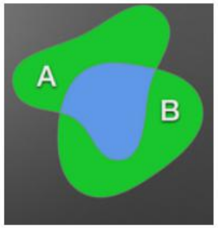

| | deepLab (baseline) | avg. | gtDepth tiedKernel | gtDepth untied Kernel |
|---|---|---|---|---|
| IoU | 0.738 | 0.747 | 0.748 | 0.753 |

# Depth-aware pooling module

Cityscapes dataset

metric: Intersection over Union (IoU)

$$IOU(A,B) = \frac{|A \cap B|}{|A \cup B|}$$




using the ground-truth disparity map, 5 discete bins for 5 scales {1,2,4,8,16}

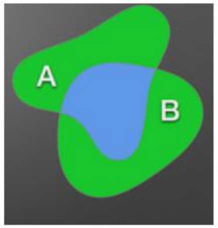
| | deepLab (baseline) | avg. | gtDepth tiedKernel | gtDepth untied Kernel | predDepth untied Kernel |
|---|---|---|---|---|---|
| IoU | 0.738 | 0.747 | 0.748 | 0.753 | **0.759** |

# Depth-aware pooling module

Cityscapes dataset

metric: Intersection over Union (IoU)

$$IOU(A,B) = \frac{|A \cap B|}{|A \cup B|}$$




using the ground-truth disparity map, 5 discete bins for 5 scales {1,2,4,8,16}

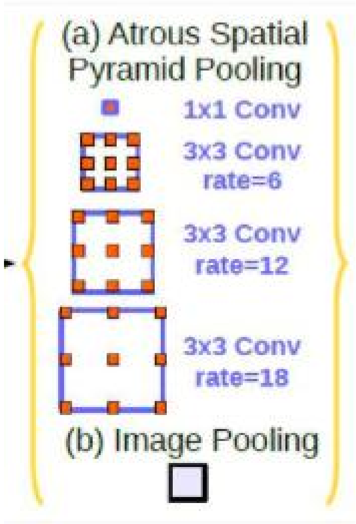
## Why better?

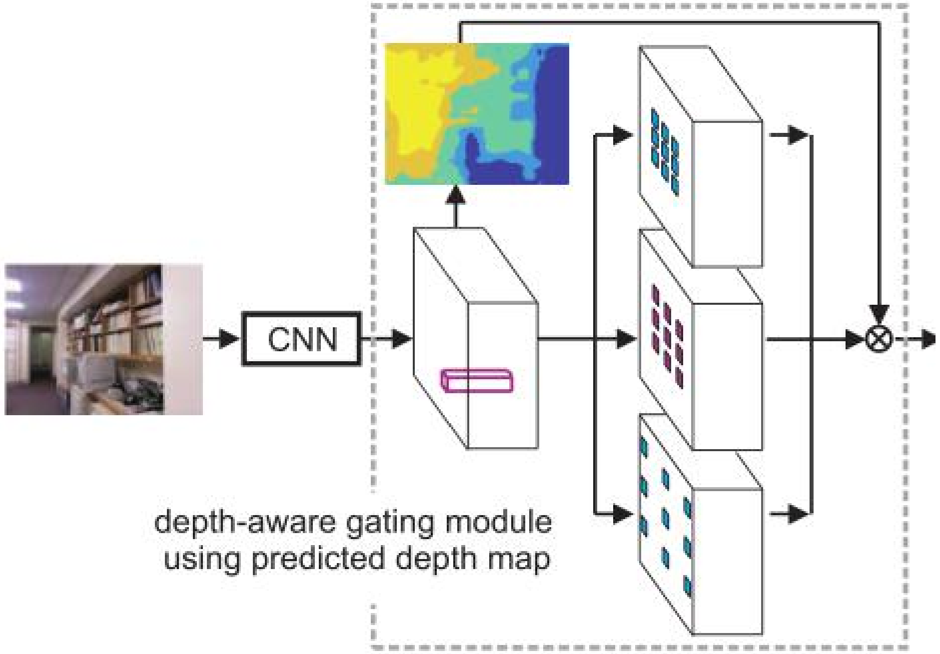
| | deepLab (baseline) | avg. | gtDepth tiedKernel | gtDepth untied Kernel | predDepth untied Kernel |
|---|---|---|---|---|---|
| IoU | 0.738 | 0.747 | 0.748 | 0.753 | **0.759** |

# Depth-aware pooling module

many possibilities to explore --

1. sharing the parameters in this pooling module (multiPool)
2. averaging the feature vs. attention vs. depth-aware gating



# Depth-aware pooling module

many possibilities to explore --

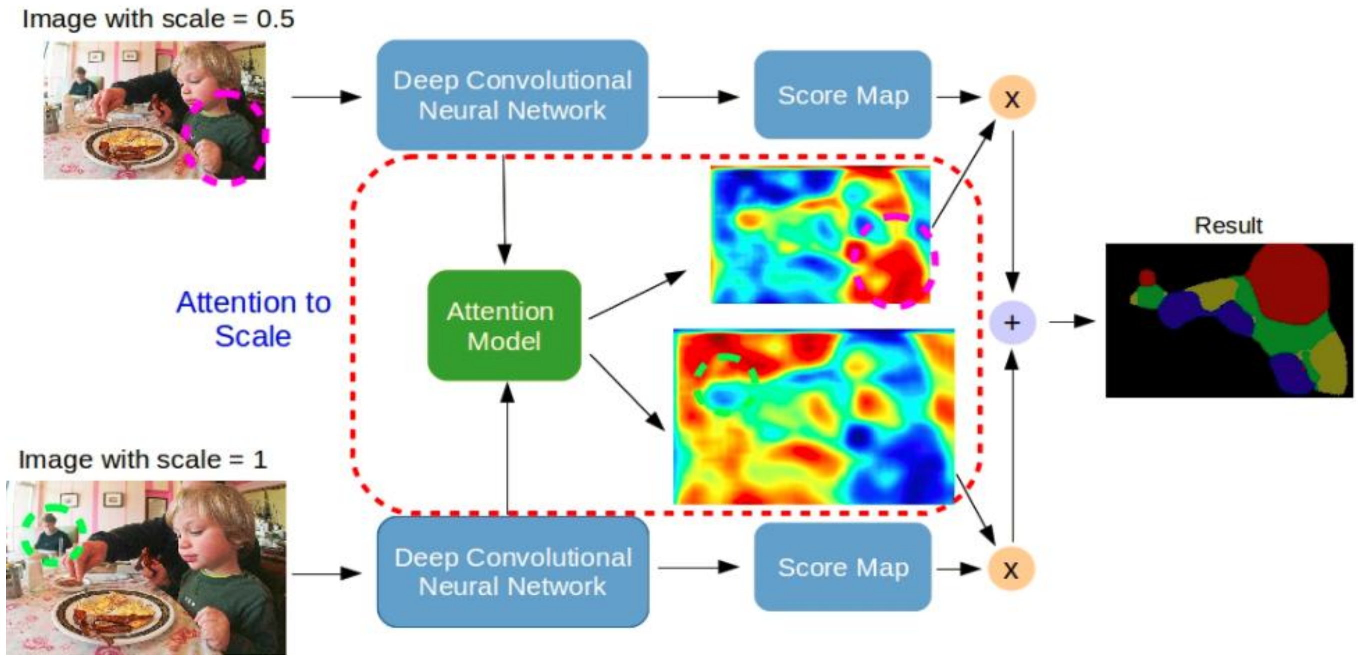
1. sharing the parameters in this pooling module (multiPool)
2. averaging the feature vs. attention vs. depth-aware gating

| | | | | |
|---|---|---|---|---|
| baseline | | | | 0.738 |
| MultiPool | tied weights | average | | 0.747 |
| | | depth-gating | | 0.748 |
| | untied weights | average | | 0.751 |
| | | attention | | 0.754 |
| | | depth-gating | gt-depth | 0.753 |
| | | | pred-depth | 0.759 |

# Depth-aware pooling module

many possibilities to explore --

1. sharing the parameters in this pooling module (multiPool)
2. averaging the feature vs. attention vs. depth-aware gating
3. MultiPool vs. MultiScale (input)



# Depth-aware pooling module

many possibilities to explore --

1. sharing the parameters in this pooling module (multiPool)
2. averaging the feature vs. attention vs. depth-aware gating
3. MultiPool vs. MultiScale (input)

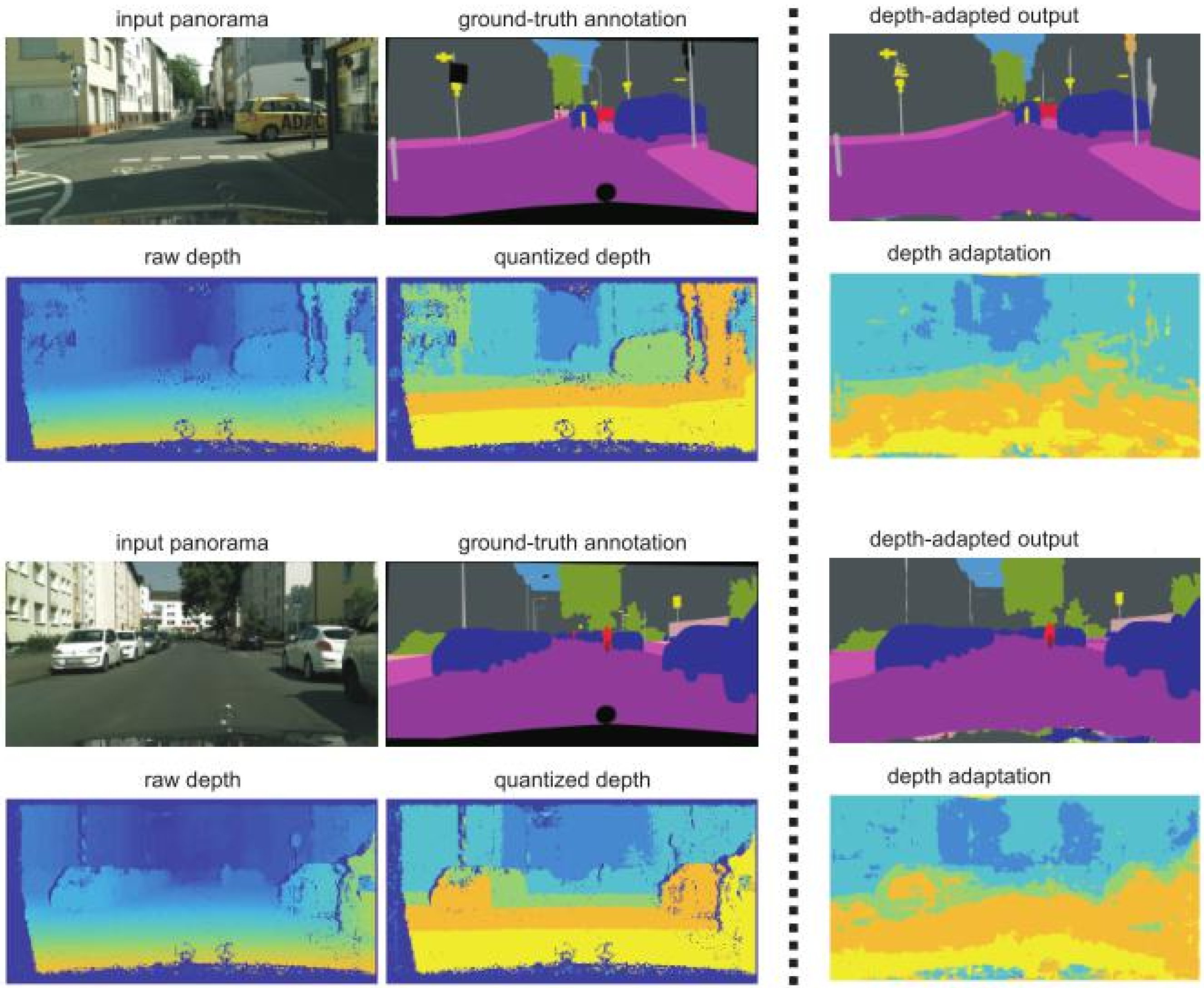
- baseline 0.738
- MultiPool
  - tied weights
    - average 0.747
    - depth-gating 0.748
  - untied weights
    - average 0.751
    - attention 0.754
    - depth-gating
      - gt-depth 0.753
      - pred-depth 0.759
- MultiScale
  - tied weights
    - average 0.750
    - depth-gating 0.751
  - untied weights
    - average ∅
    - depth-gating ∅



S. Kong, C. Fowlkes, Recurrent Scene Parsing with Perspective Understanding in the Loop, CVPR, 2018

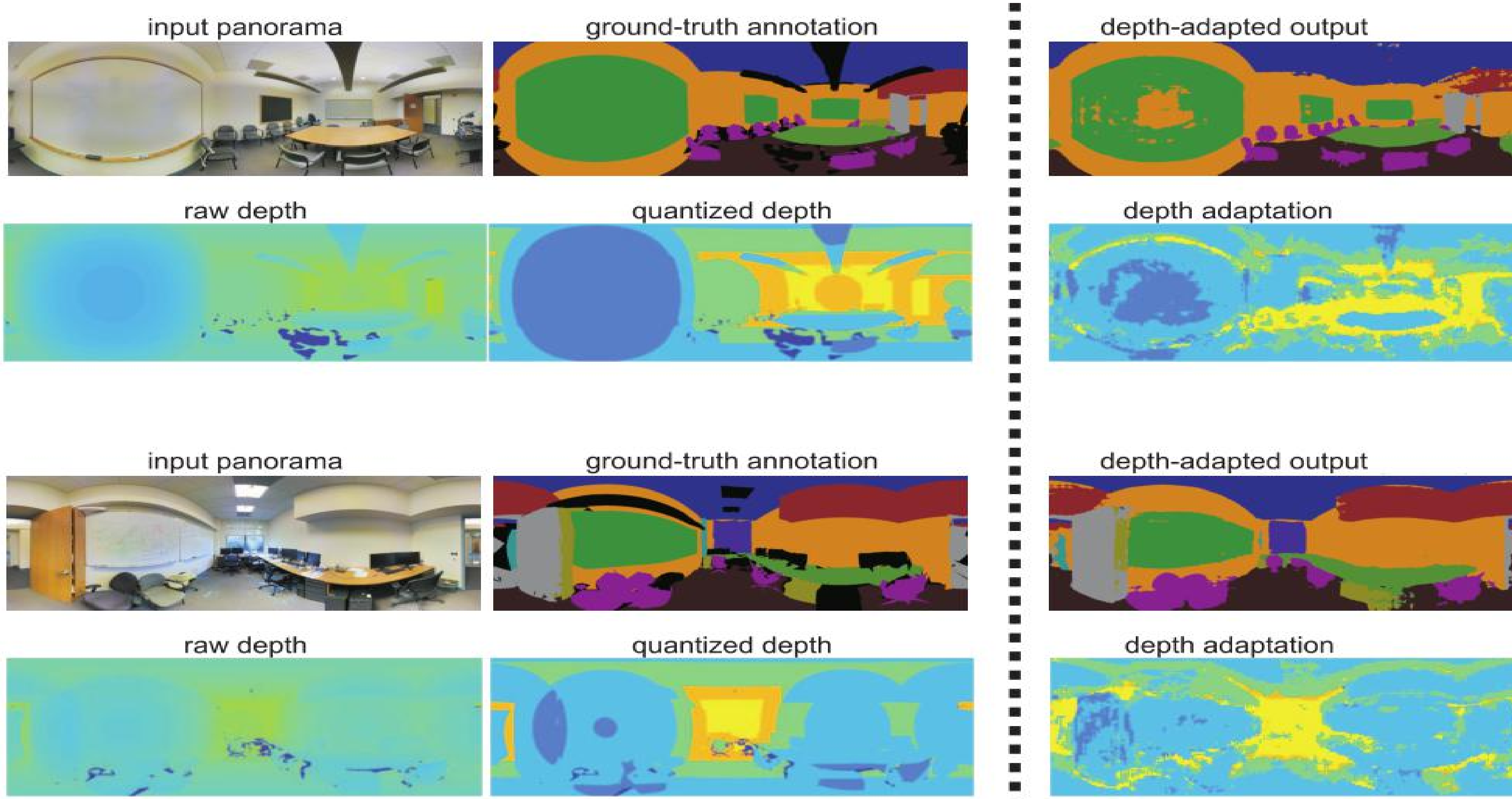
# Depth-aware pooling module

## Qualitative Results -- street images



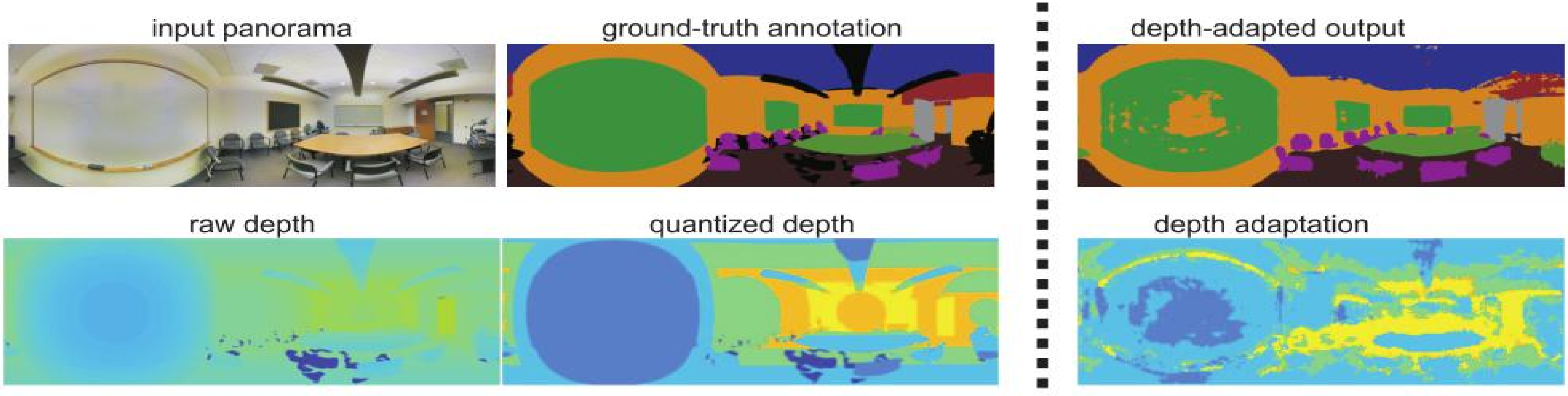
# Depth-aware pooling module

Qualitative Results -- panorama images



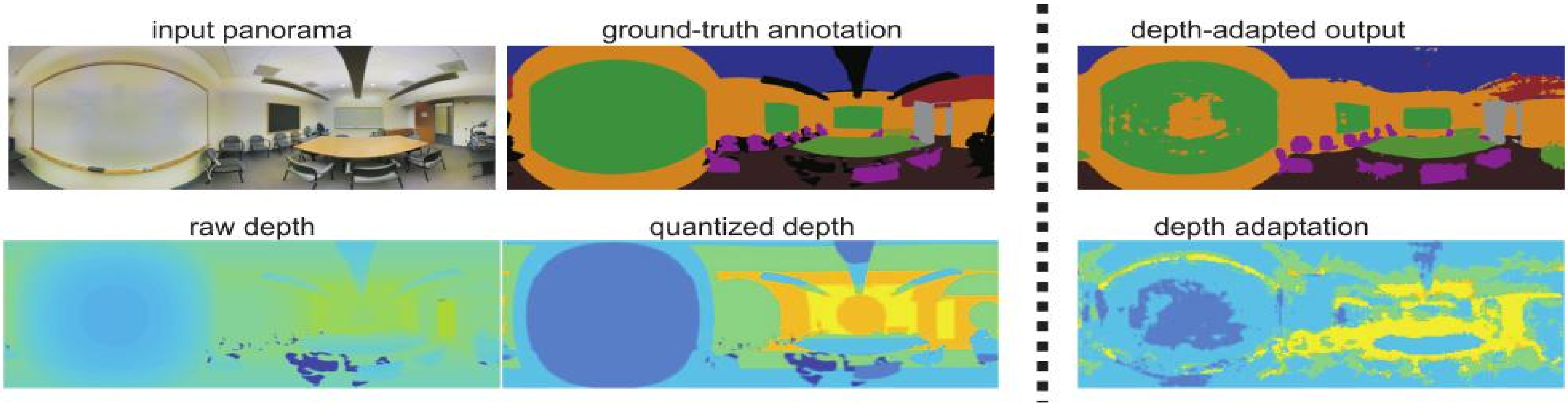
# Depth-aware pooling module

## Good enough?



# Recurrent Refining Module

## Recurrent Refining with Perspective Understanding in the Loop



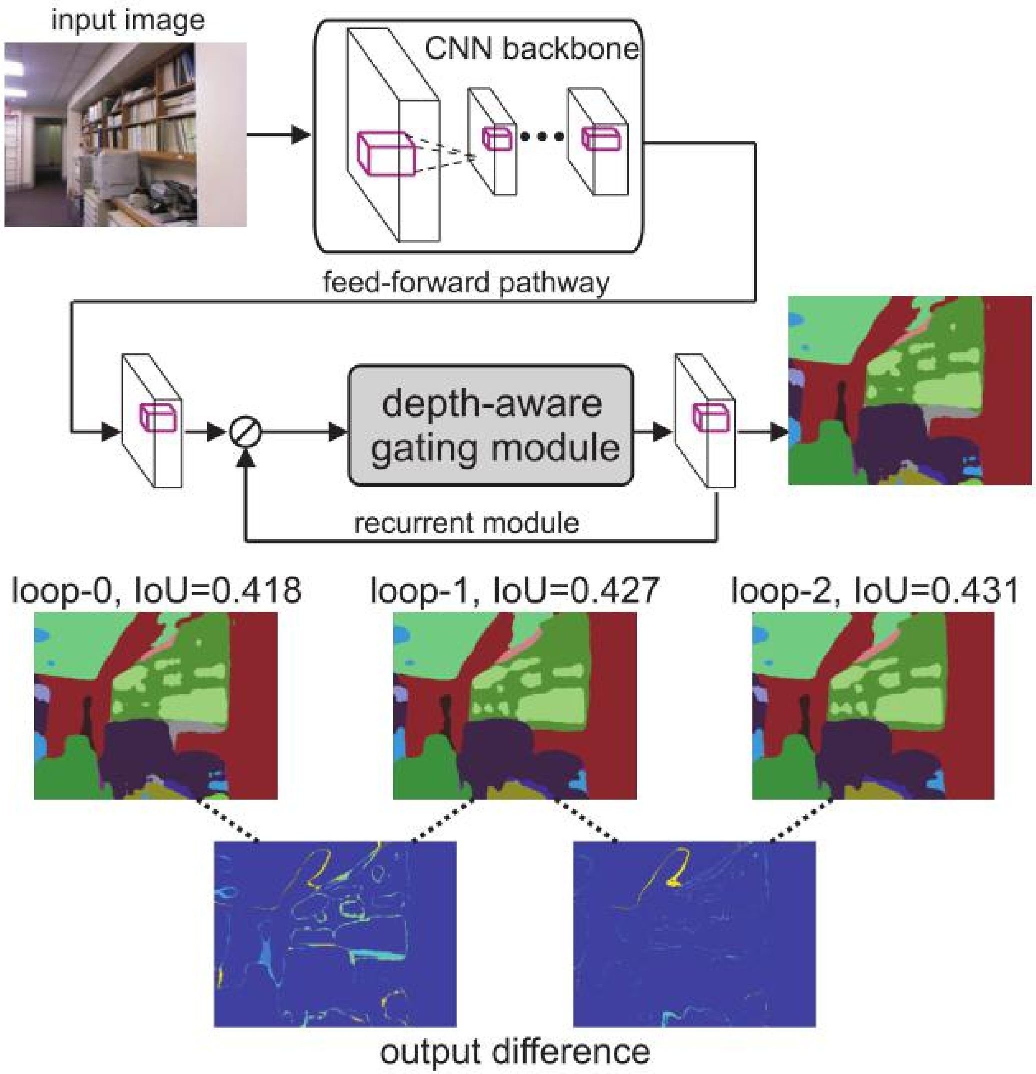

UCIrvine
University of California, Irvine

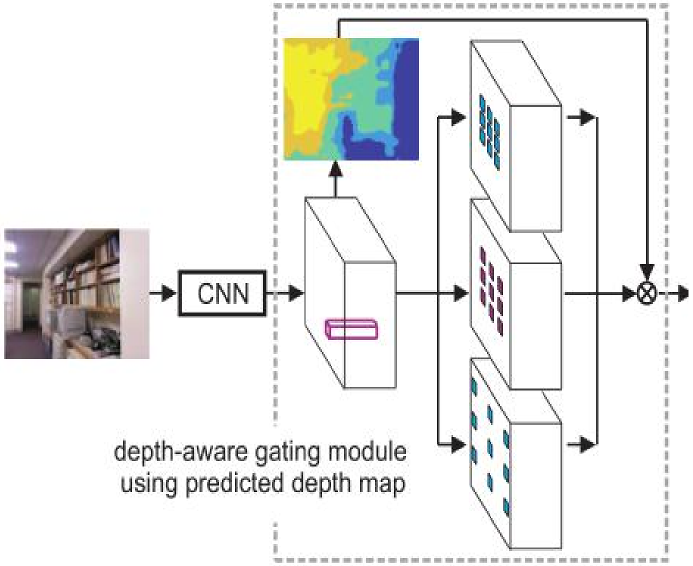
# Recurrent Refining Module

1. Background
2. Attention to Perspective: Depth-aware Pooling Module
3. Recurrent Refining with Perspective Understanding in the Loop
4. Attention to Perspective Again
5. Pixel-wise Attentional Gating (PAG)
6. Pixel-Level Dynamic Routing
7. Conclusion

# Recurrent Refinement Module

Recurrently refining the results by adapting the predicted depth



# Recurrent Refinement Module

unrolling the recurrent module during training

adding a loss to each unrolled loop

embedding the depth-aware gating module in the loops

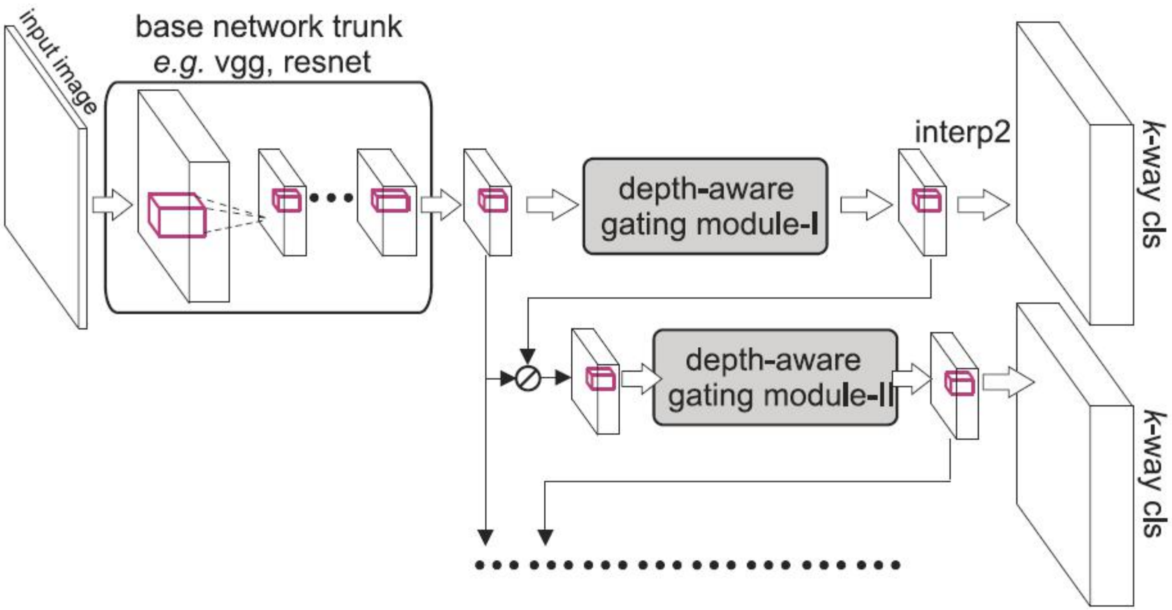



Figure 2: recurrentModule.

S. Kong, C. Fowlkes, Recurrent Scene Parsing with Perspective Understanding in the Loop, CVPR, 2018

# Recurrent Refinement Module

Recurrently refining the results by adapting the predicted depth

| | NYU-depth-v2 [35] | | SUN-RGBD [35] | | Stanford-2D-3D [1] | | Cityscapes [9] |
|---|---|---|---|---|---|---|---|
| | IoU | pixel acc. | IoU | pixel acc. | IoU | pixel acc. | IoU |
| baseline | 0.406 | 0.703 | 0.402 | 0.776 | 0.644 | 0.866 | 0.738 |
| w/ gt-depth | 0.413 | 0.708 | 0.422 | 0.787 | 0.730 | 0.897 | 0.753 |
| w/ pred-depth | 0.418 | 0.711 | 0.423 | 0.789 | 0.742 | 0.900 | 0.759 |
| loop1 w/o depth | 0.419 | 0.706 | 0.432 | 0.793 | 0.744 | 0.901 | 0.762 |
| loop1 w/ gt-depth | 0.425 | 0.711 | 0.439 | 0.798 | 0.747 | 0.902 | 0.769 |
| loop1 w/ pred-depth | 0.427 | 0.712 | 0.440 | 0.798 | 0.753 | 0.906 | 0.772 |
| loop2 | 0.431 | 0.713 | 0.443 | 0.799 | 0.760 | 0.908 | 0.776 |
| loop2 (test-aug) | 0.445 | 0.721 | 0.451 | 0.803 | 0.765 | 0.910 | 0.791 / 0.782$^{*}$ |
| DeepLab [6] | - | - | - | - | 0.698$^{\dagger}$ | 0.880$^{\dagger}$ | 0.704 / 0.704$^{*}$ |
| LRR [13] | - | - | - | - | - | - | 0.700 / 0.697$^{*}$ |
| Context [28] | 0.406 | 0.700 | 0.423 | 0.784 | - | - | - / 0.716$^{*}$ |
| PSPNet [38] | - | - | - | - | 0.674$^{\dagger}$ | 0.876$^{\dagger}$ | - / 0.784$^{*}$ |
| RefineNet-Res50 [27] | 0.438 | - | - | - | - | - | - / - |
| RefineNet-Res101 [27] | 0.447 | - | 0.457 | 0.804 | - | - | - / 0.736$^{*}$ |
| RefineNet-Res152 [27] | 0.465 | 0.736 | 0.459 | 0.806 | - | - | - / - |

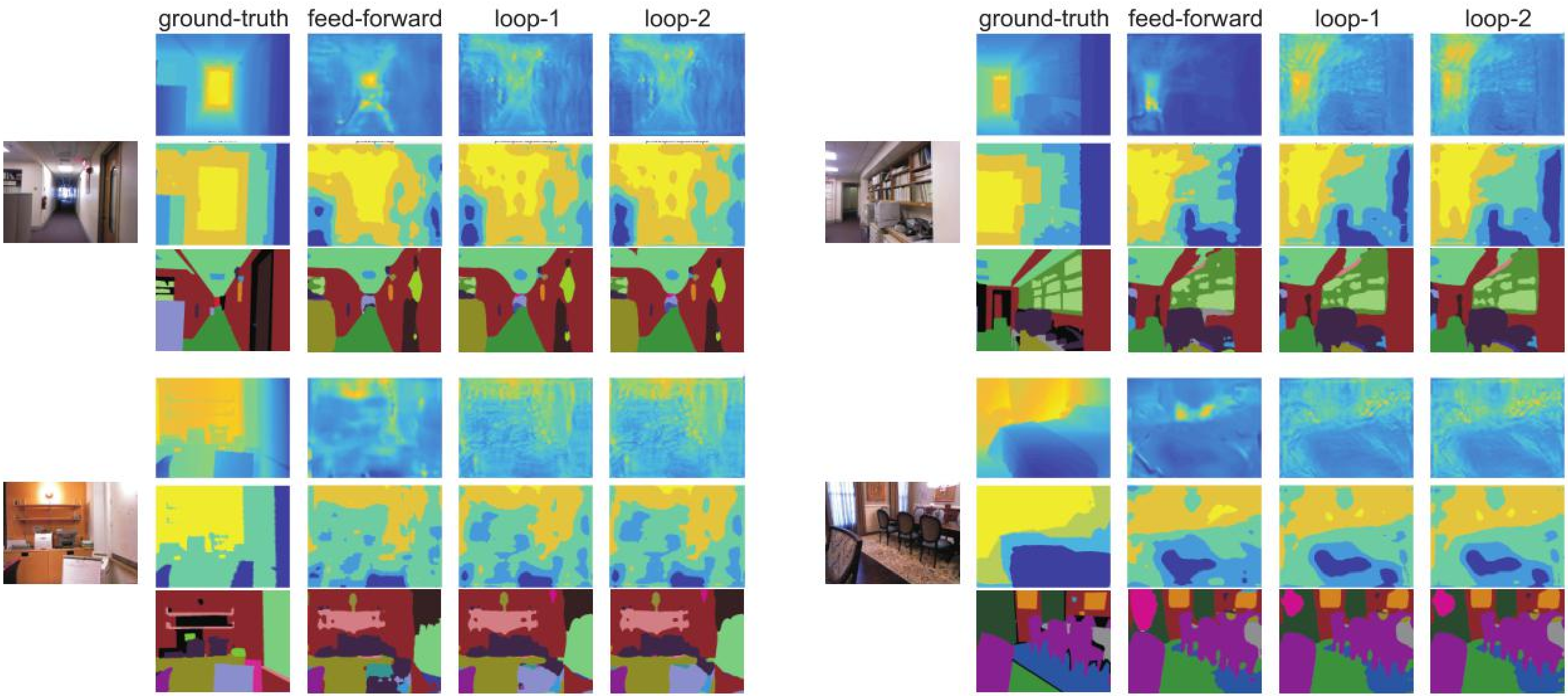


S. Kong, C. Fowlkes, Recurrent Scene Parsing with Perspective Understanding in the Loop, CVPR, 2018

# Recurrent Refinement Module

Qualitative Results -- NYU-depth-v2 indoor

***blue --> closer --> larger pooling size***

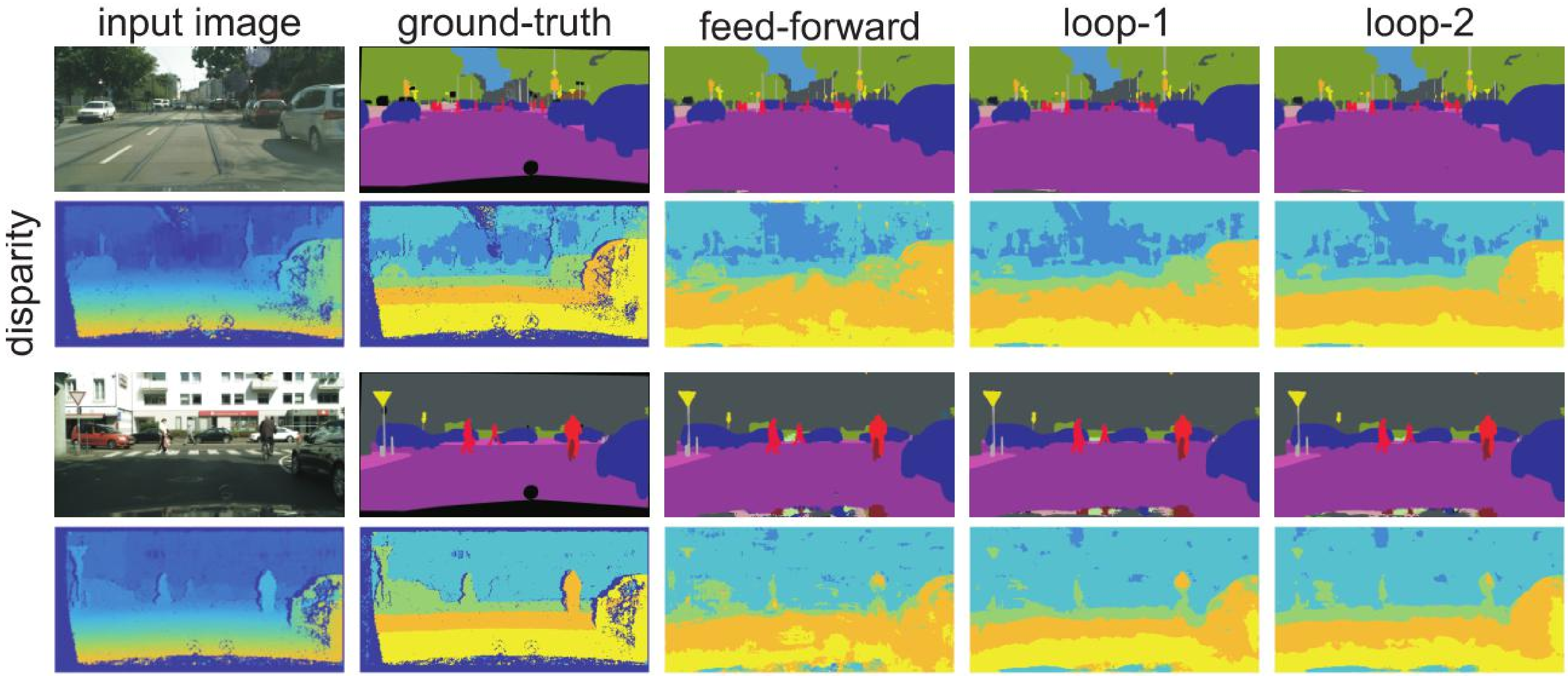



S. Kong, C. Fowlkes, Recurrent Scene Parsing with Perspective Understanding in the Loop, CVPR, 2018

# Recurrent Refinement Module

Qualitative Results -- Cityscapes

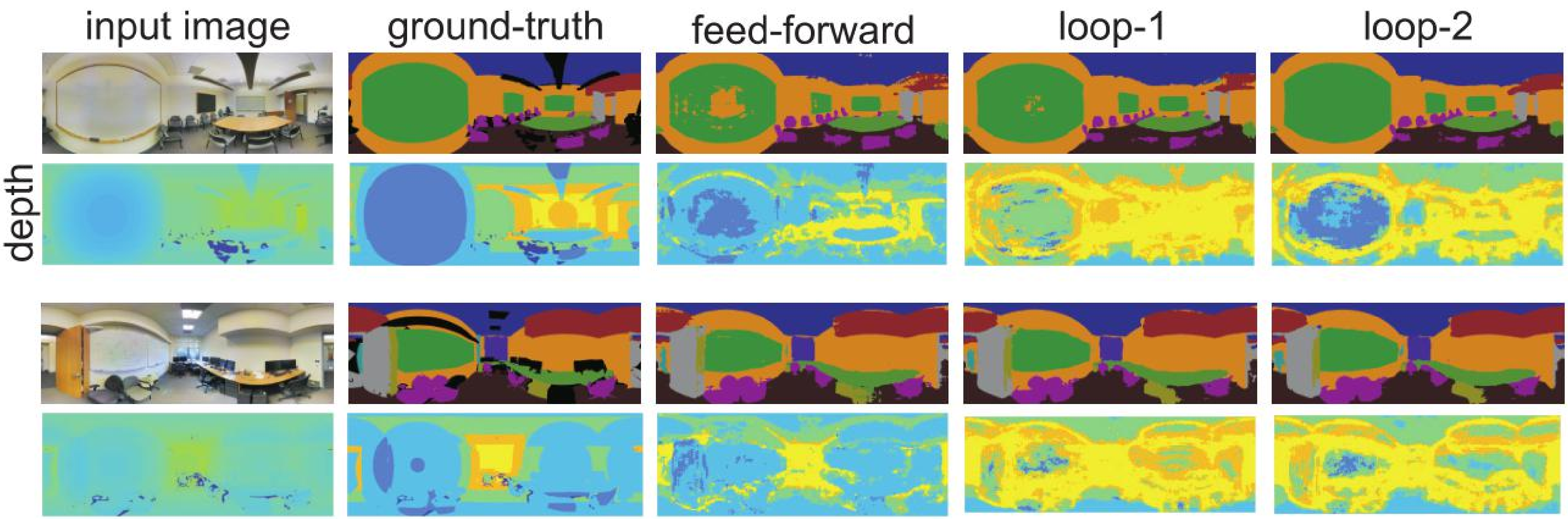
***yellow --> closer --> larger pooling size***




S. Kong, C. Fowlkes, Recurrent Scene Parsing with Perspective Understanding in the Loop, CVPR, 2018

# Recurrent Refinement Module

Qualitative Results -- Stanford-2D-3D (panoramas)



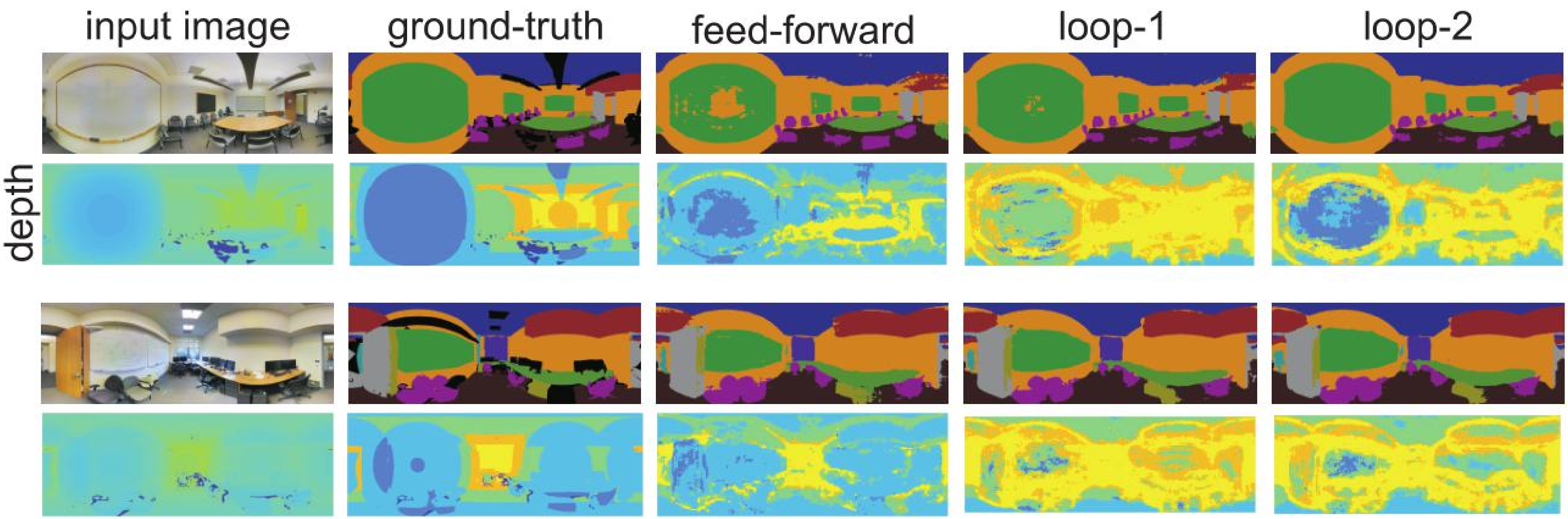

UCIrvine
University of California, Irvine

S. Kong, C. Fowlkes, Recurrent Scene Parsing with Perspective Understanding in the Loop, CVPR, 2018

# Recurrent Refinement Module

Qualitative Results -- Stanford-2D-3D (panoramas)

## Holes are filled!






S. Kong, C. Fowlkes, Recurrent Scene Parsing with Perspective Understanding in the Loop, CVPR, 2018

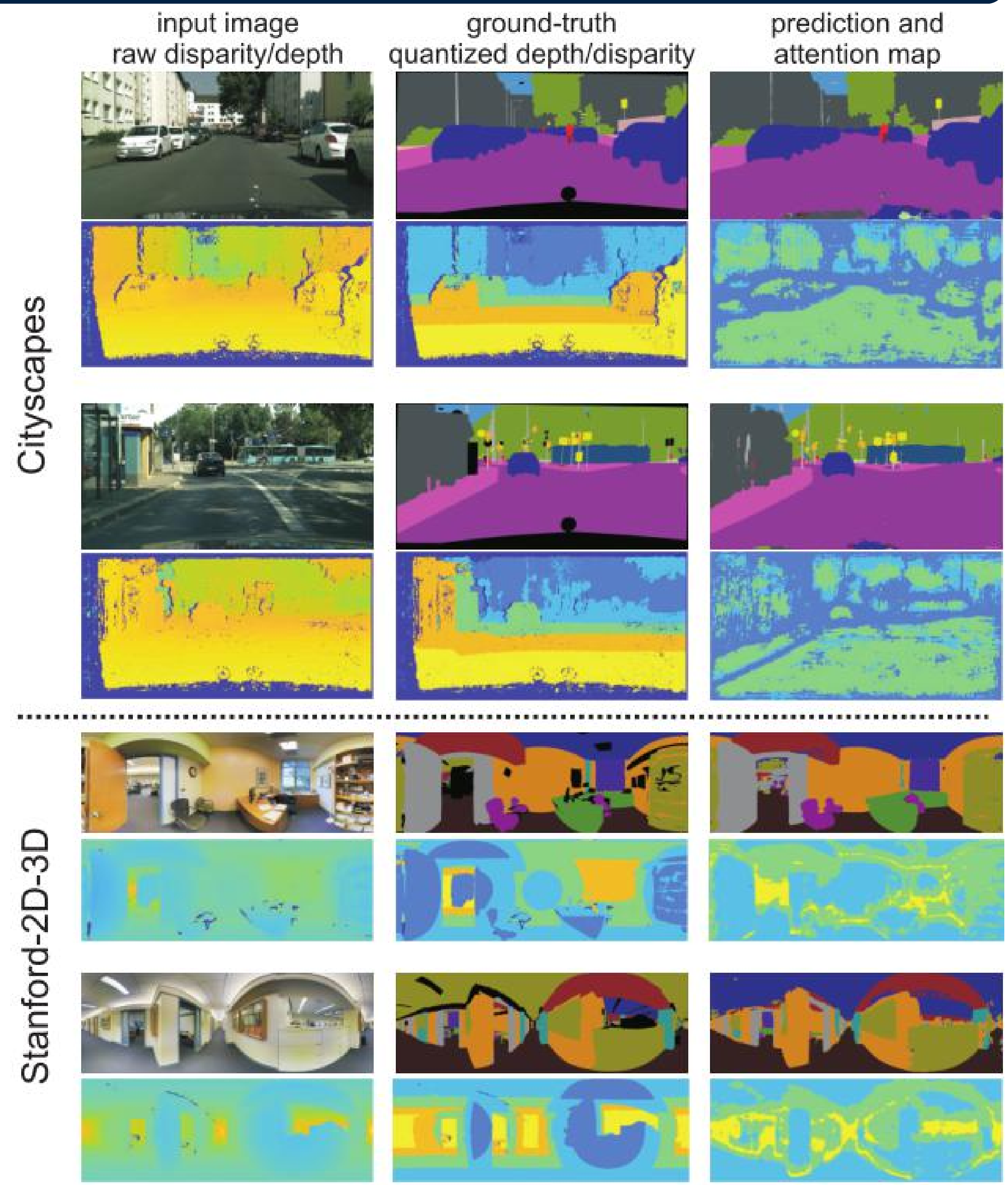
# Outline

1. Background
2. Attention to Perspective: Depth-aware Pooling Module
3. Recurrent Refining with Perspective Understanding in the Loop
4. Attention to Scale Again
5. Pixel-wise Attentional Gating (PAG)
6. Pixel-Level Dynamic Routing
7. Conclusion

UCIrvine
University of California, Irvine

# Attention to Scale Again

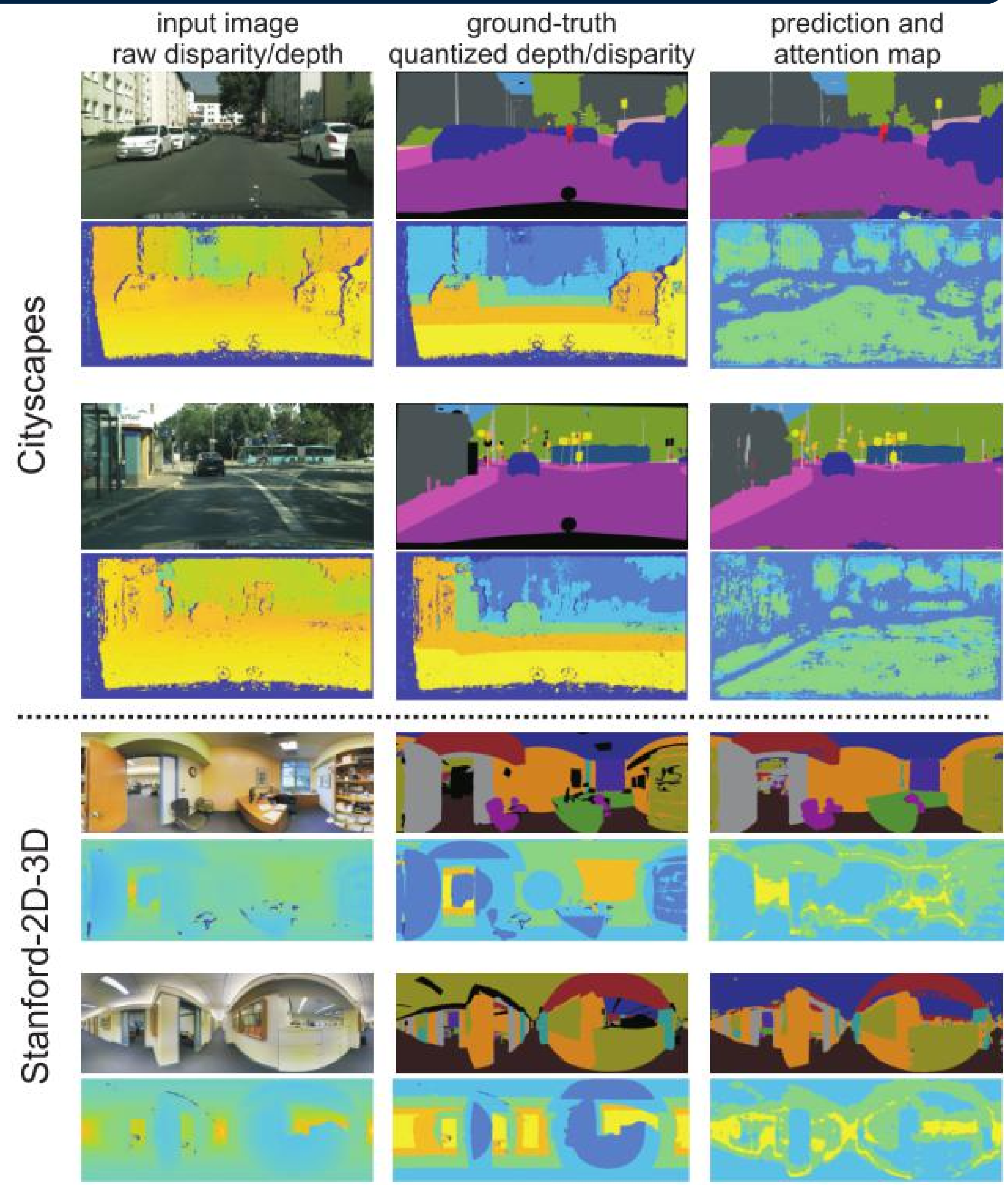
- baseline 0.738
- MultiPool
  - tied weights
    - average 0.747
    - depth-gating 0.748
  - untied weights
    - average 0.751
    - attention 0.754
    - depth-gating
      - gt-depth 0.753
      - pred-depth 0.759



# Attention to Scale Again

Attentional maps prevent the model from pooling across different segments.

- baseline 0.738
- MultiPool
  - tied weights
    - average 0.747
    - depth-gating 0.748
  - untied weights
    - average 0.751
    - attention 0.754
    - depth-gating
      - gt-depth 0.753
      - pred-depth 0.759

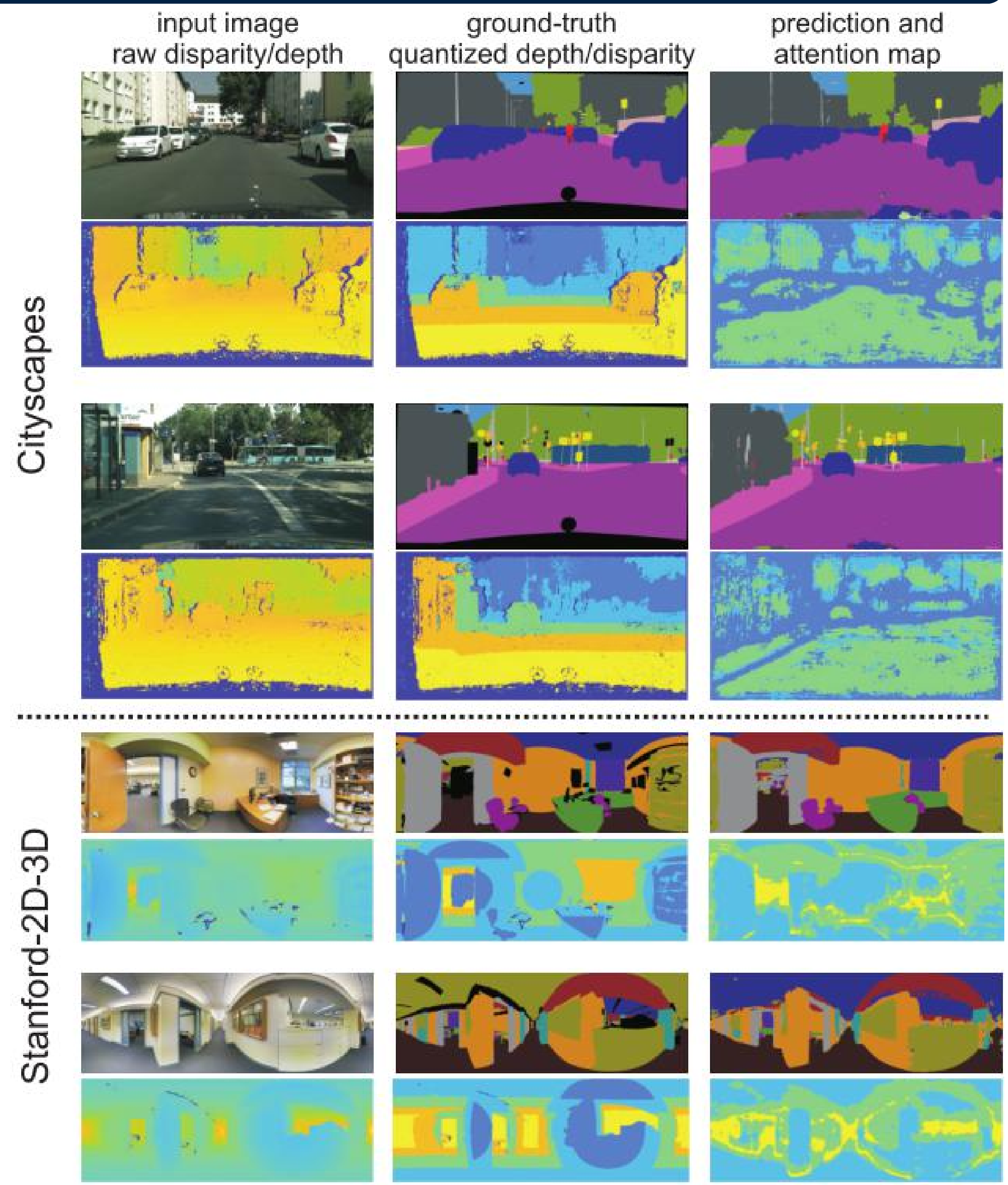


# Attention to Scale Again

Attentional maps prevent the model from pooling across different segments.

Some scales are rarely used.

- baseline 0.738
- MultiPool
  - tied weights
    - average 0.747
    - depth-gating 0.748
  - untied weights
    - average 0.751
    - attention 0.754
    - depth-gating
      - gt-depth 0.753
      - pred-depth 0.759




UCIrvine
University of California, Irvine

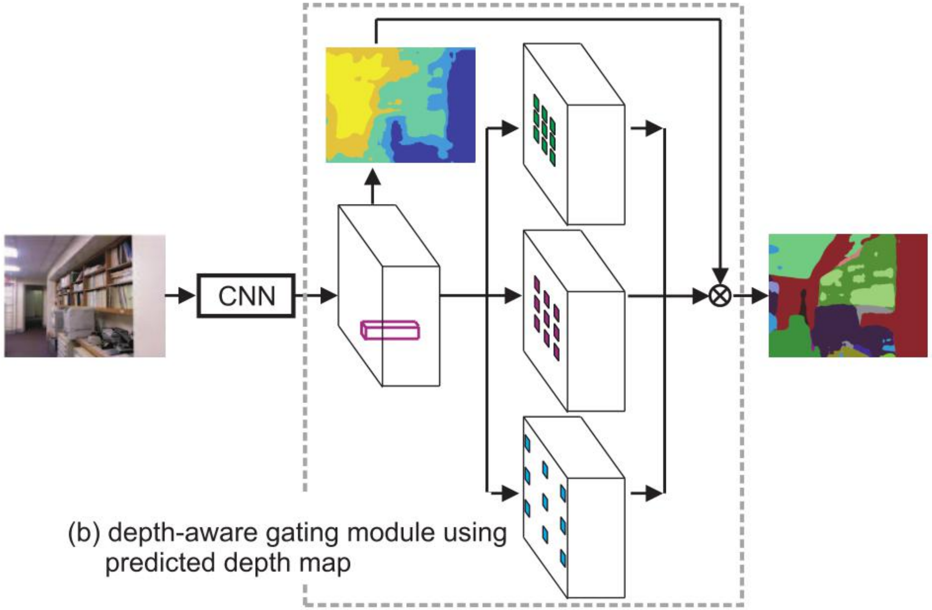
# Attention to Scale Again

learning attentional module to aggregate info

six scales with dilate rates {1, 2, 4, 6, 8, 10}

NYU-depth-v2 dataset (indoor scene parsing)

ResNet50 backbone




(b) depth-aware gating module using predicted depth map

# Attention to Scale Again

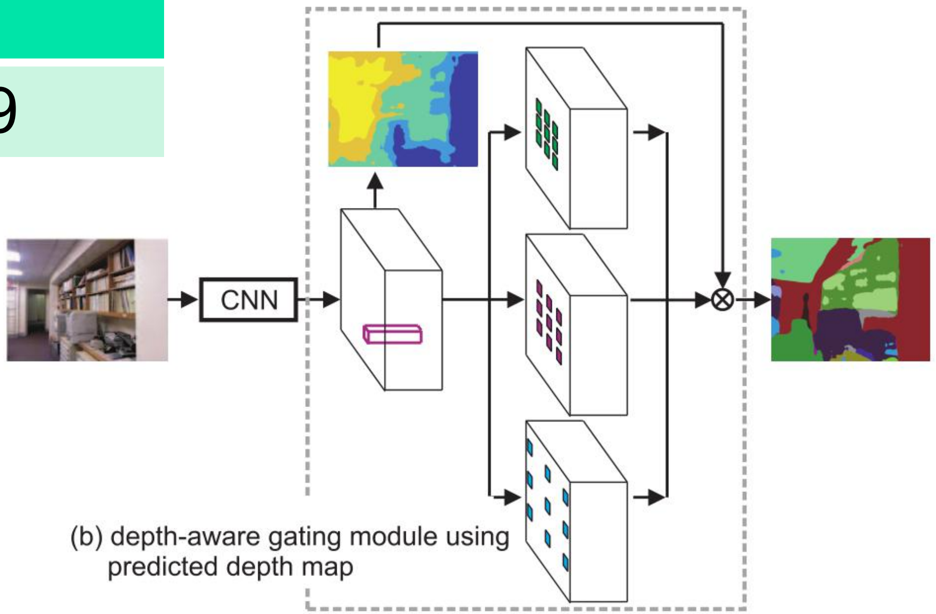
learning attentional module to choose the “correct” pooling scale

six scales with dilate rates {1, 2, 4, 6, 8, 10}

NYU-depth-v2 dataset (indoor scene parsing)

ResNet50 backbone

| | baseline | res6 |
|---|---|---|
| IoU | 0.4205 | 0.4599 |

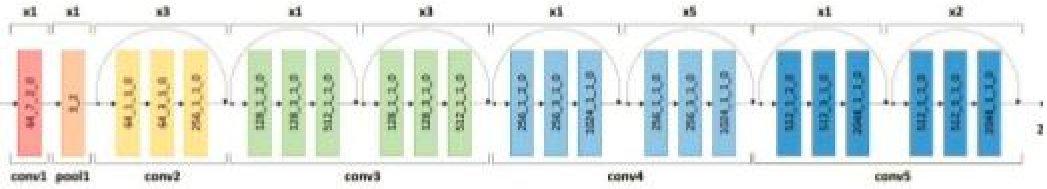



(b) depth-aware gating module using predicted depth map

# Attention to Scale Again

Which layer to insert this attentional gating module?

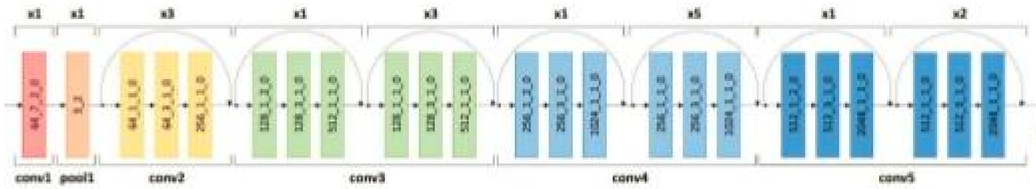
res1 res2 res3 res4 res5 res6



# Attention to Scale Again

Which layer to insert this attentional gating module?

res1 res2 res3 res4 res5 res6



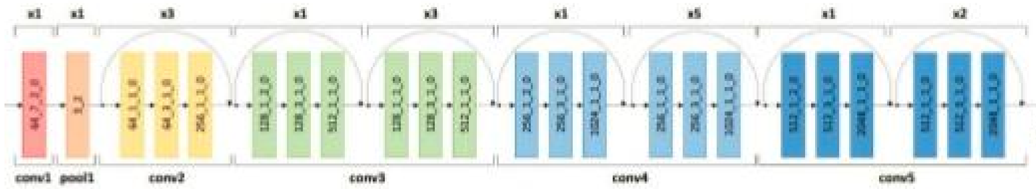

| | baseline | res6 | res5 | res4 | res3 |
|---|---|---|---|---|---|
| IoU | 0.4205 | 0.4599 | 0.4652 | 0.4567 | 0.4413 |

S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018

# Attention to Scale Again

Which layer to insert this attentional gating module?

res1 res2 res3 res4 res5 res6



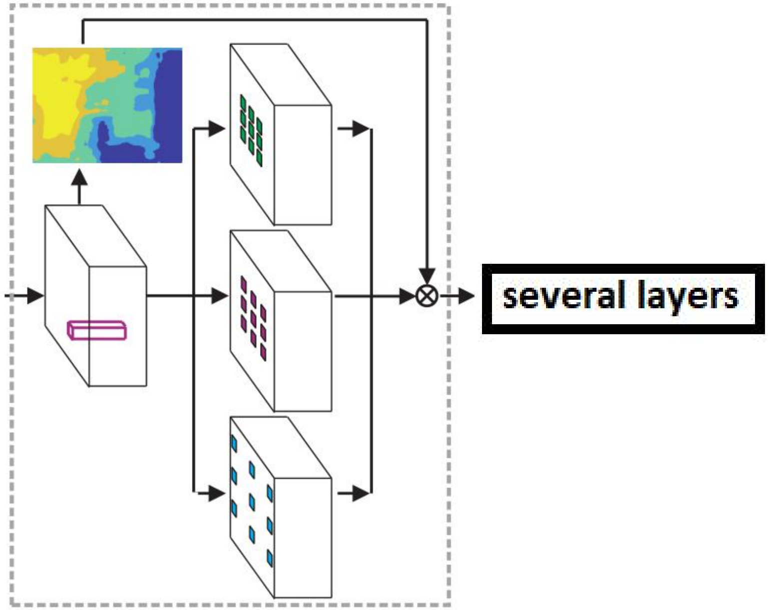

| | baseline | res6 | res5 | res4 | res3 |
|---|---|---|---|---|---|
| IoU | 0.4205 | 0.4599 | 0.4652 | 0.4567 | 0.4413 |

| | 56 | 45 | 345 | 456 | 3456 |
|---|---|---|---|---|---|
| IoU | 0.4644 | 0.4548 | 0.4483 | 0.4497 | 0.4402 |

S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018

# Attention to Scale Again

It achieves the best performance when inserting attentional gating modules at the second last residual block.




| | baseline | res5 |
|---|---|---|
| IoU | 0.4205 | 0.4652 |

| | NYU-depth-v2 [35] | |
|---|---|---|
| | IoU | pixel acc. |
| baseline | 0.406 | 0.703 |
| w/ gt-depth | 0.413 | 0.708 |
| w/ pred-depth | 0.418 | 0.711 |
| loop1 w/o depth | 0.419 | 0.706 |
| loop1 w/ gt-depth | 0.425 | 0.711 |
| loop1 w/ pred-depth | 0.427 | 0.712 |
| loop2 | 0.431 | 0.713 |
| loop2 (test-aug) | 0.445 | 0.721 |
| DeepLab [6] | - | - |
| LRR [13] | - | - |
| Context [28] | 0.406 | 0.700 |
| PSPNet [38] | - | - |
| RefineNet-Res50 [27] | 0.438 | - |
| RefineNet-Res101 [27] | 0.447 | - |
| RefineNet-Res152 [27] | 0.465 | 0.736 |

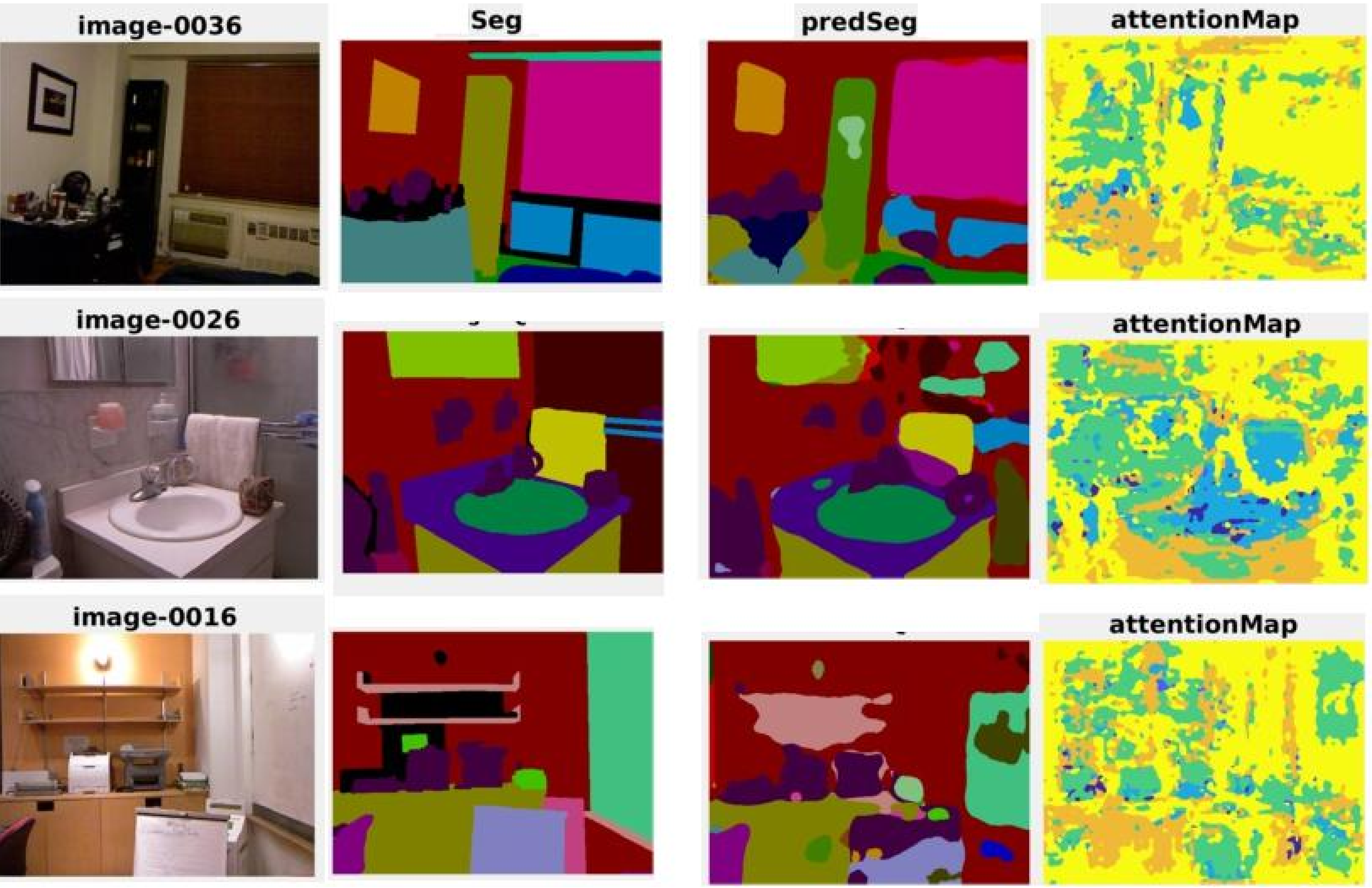
UCIrvine
UNIVERSITY OF CALIFORNIA, IRVINE

# Attention to Scale Again

Qualitative Results -- res6

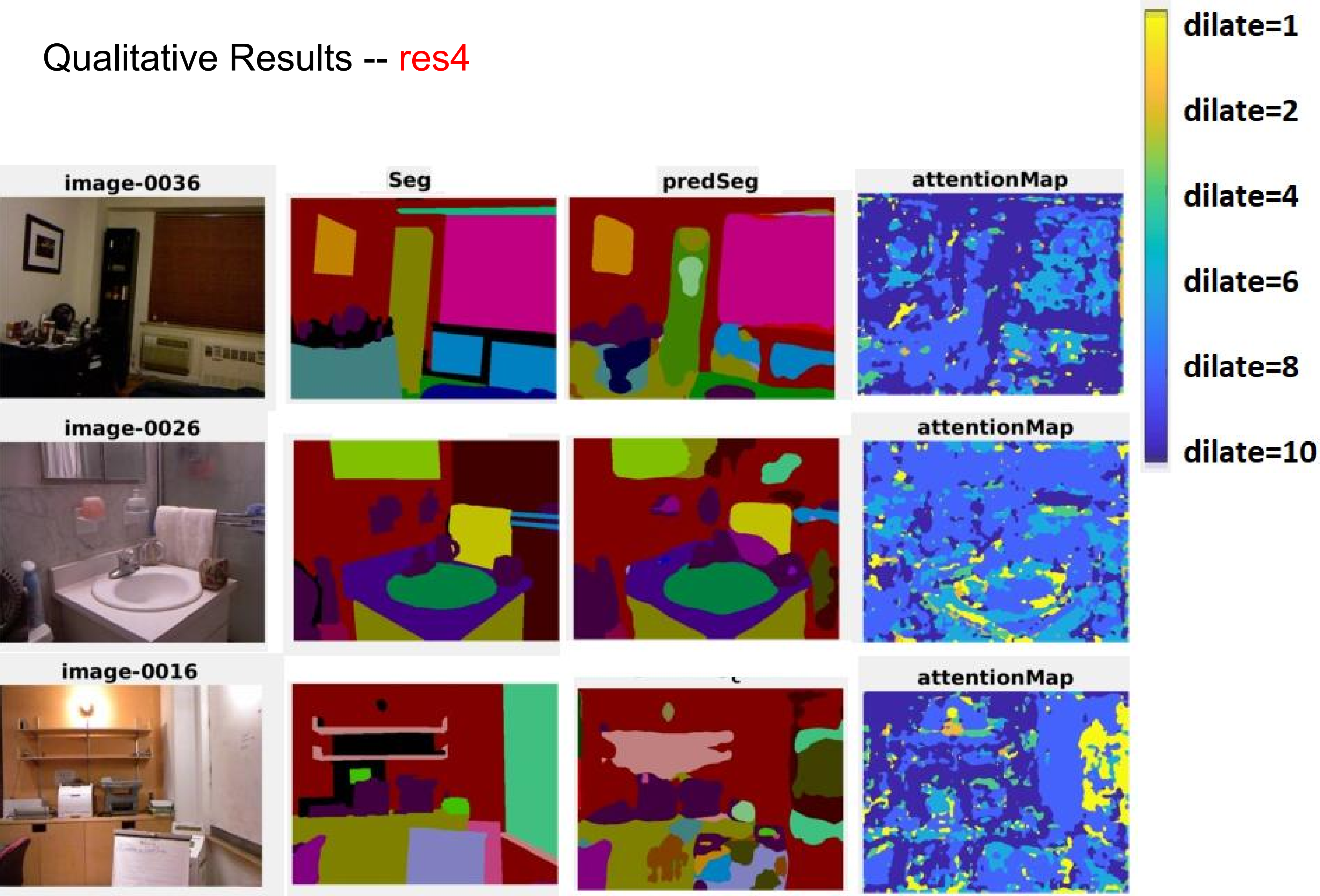
# Attention to Scale Again

Qualitative Results -- res5



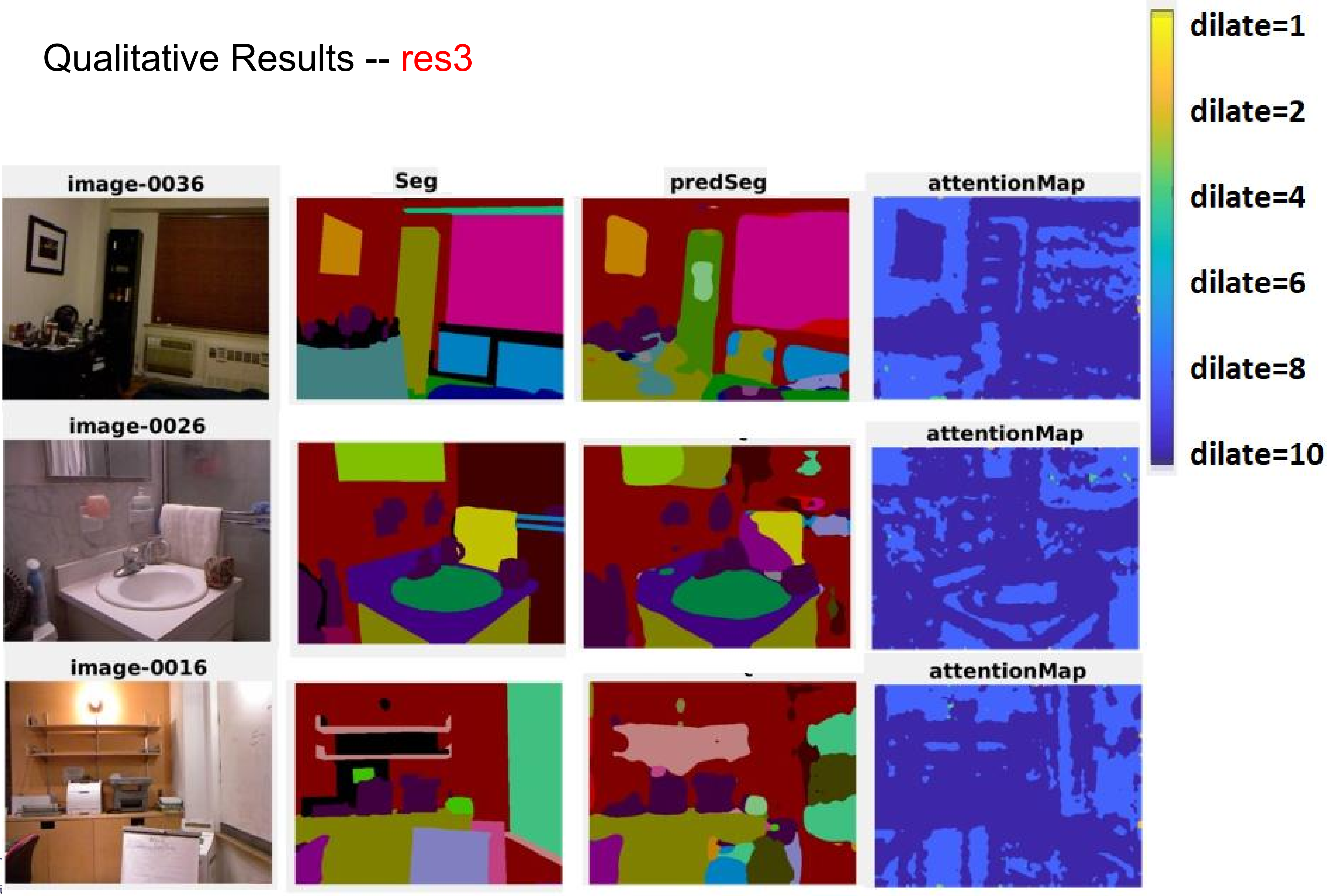

# Attention to Scale Again

Qualitative Results -- res4



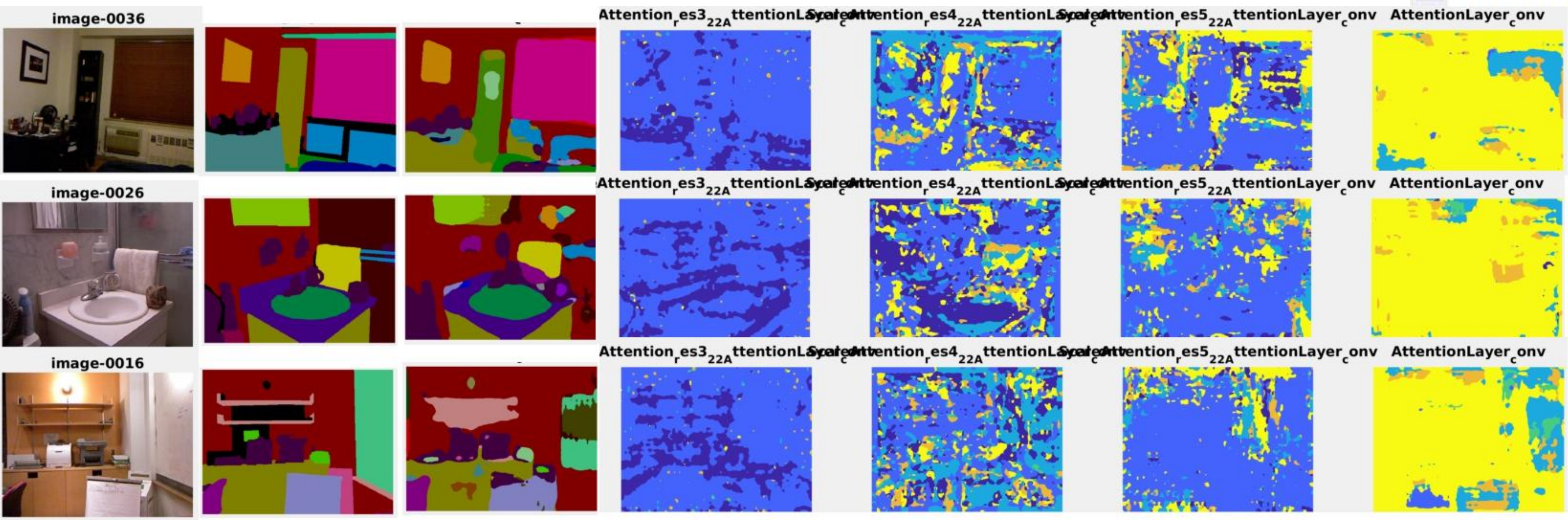
# Attention to Scale Again

Qualitative Results -- res3



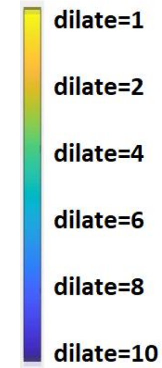
# Attention to Scale Again

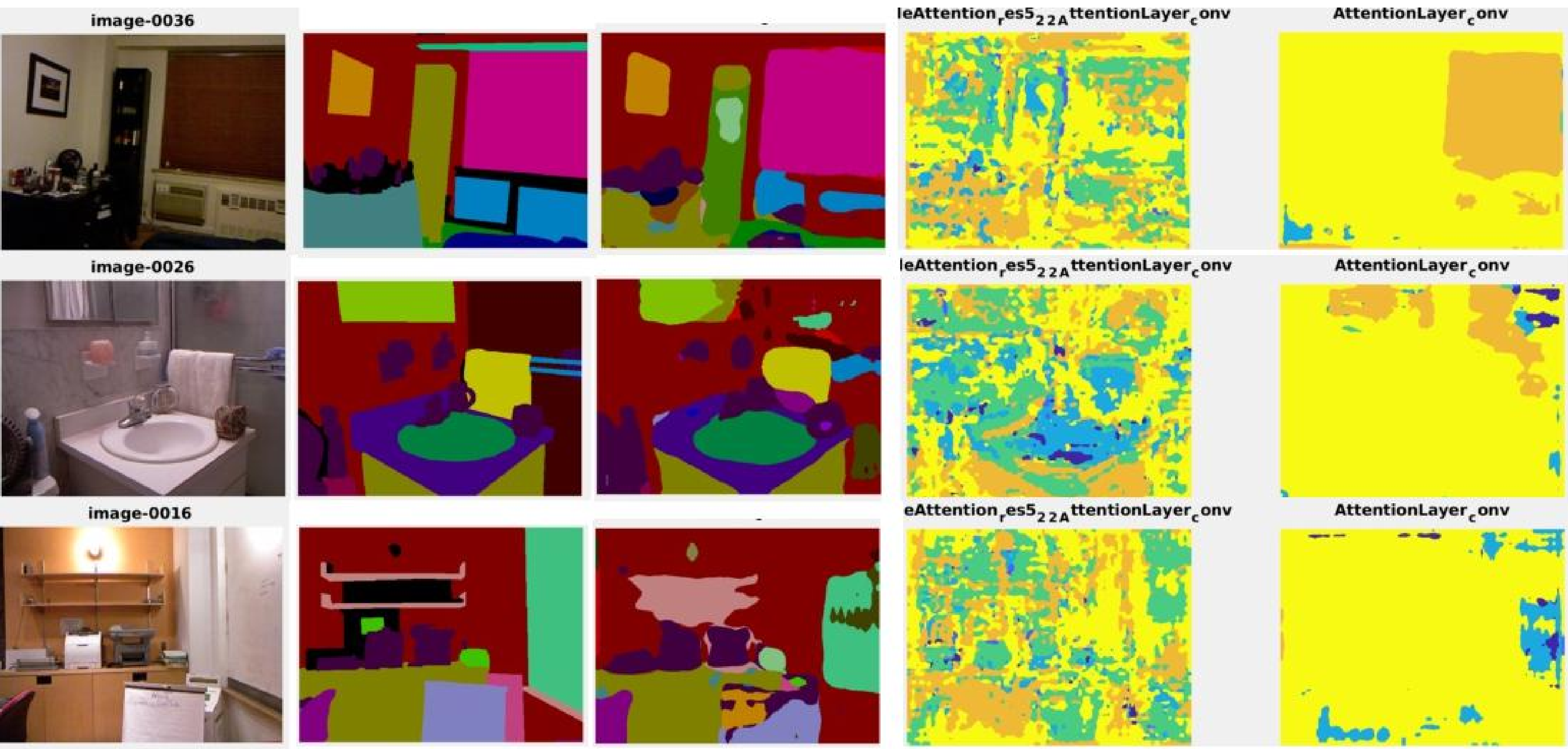
Qualitative Results -- res{3,4,5,6}

# Attention to Scale Again

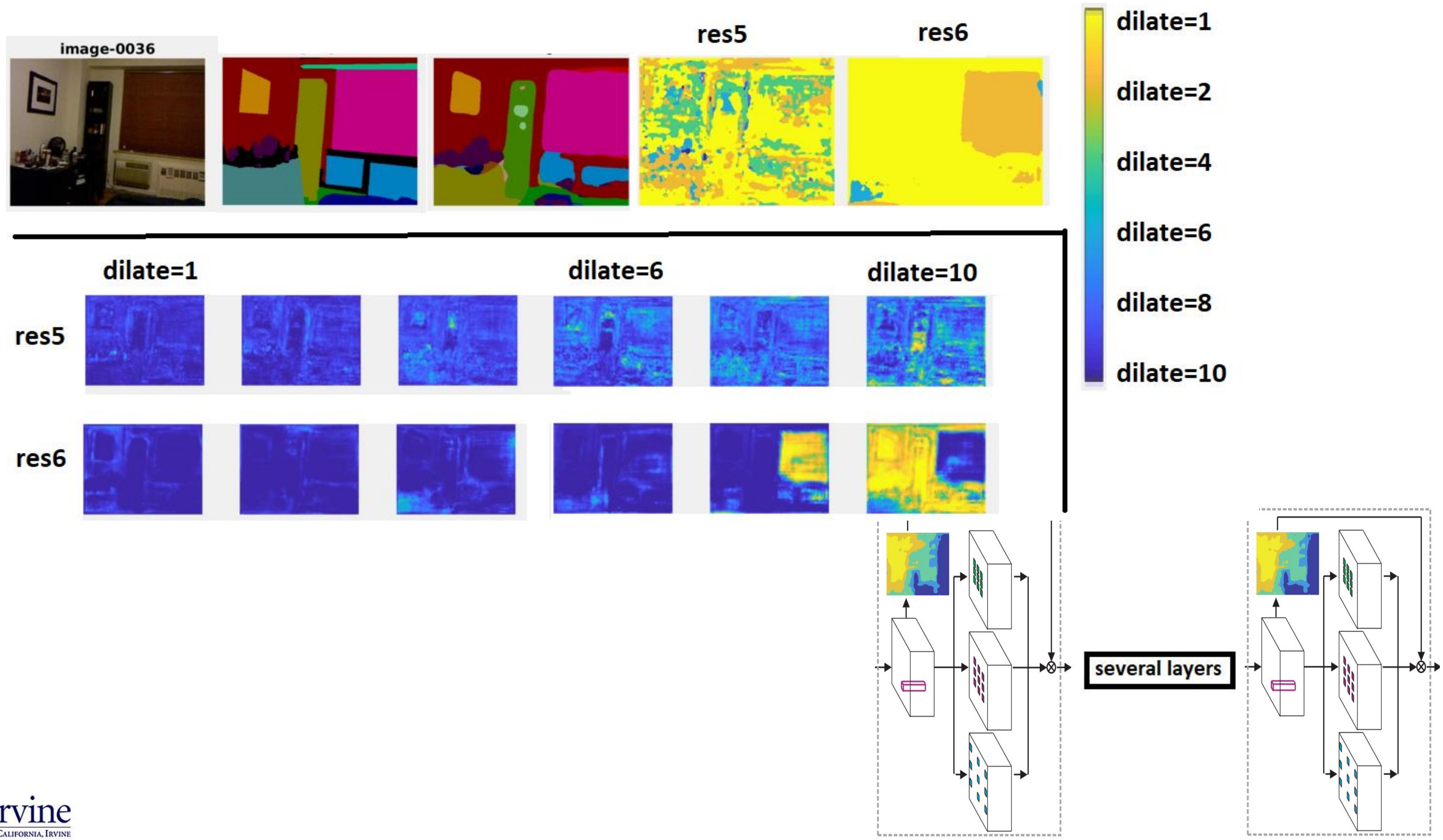
Qualitative Results -- res{5,6}



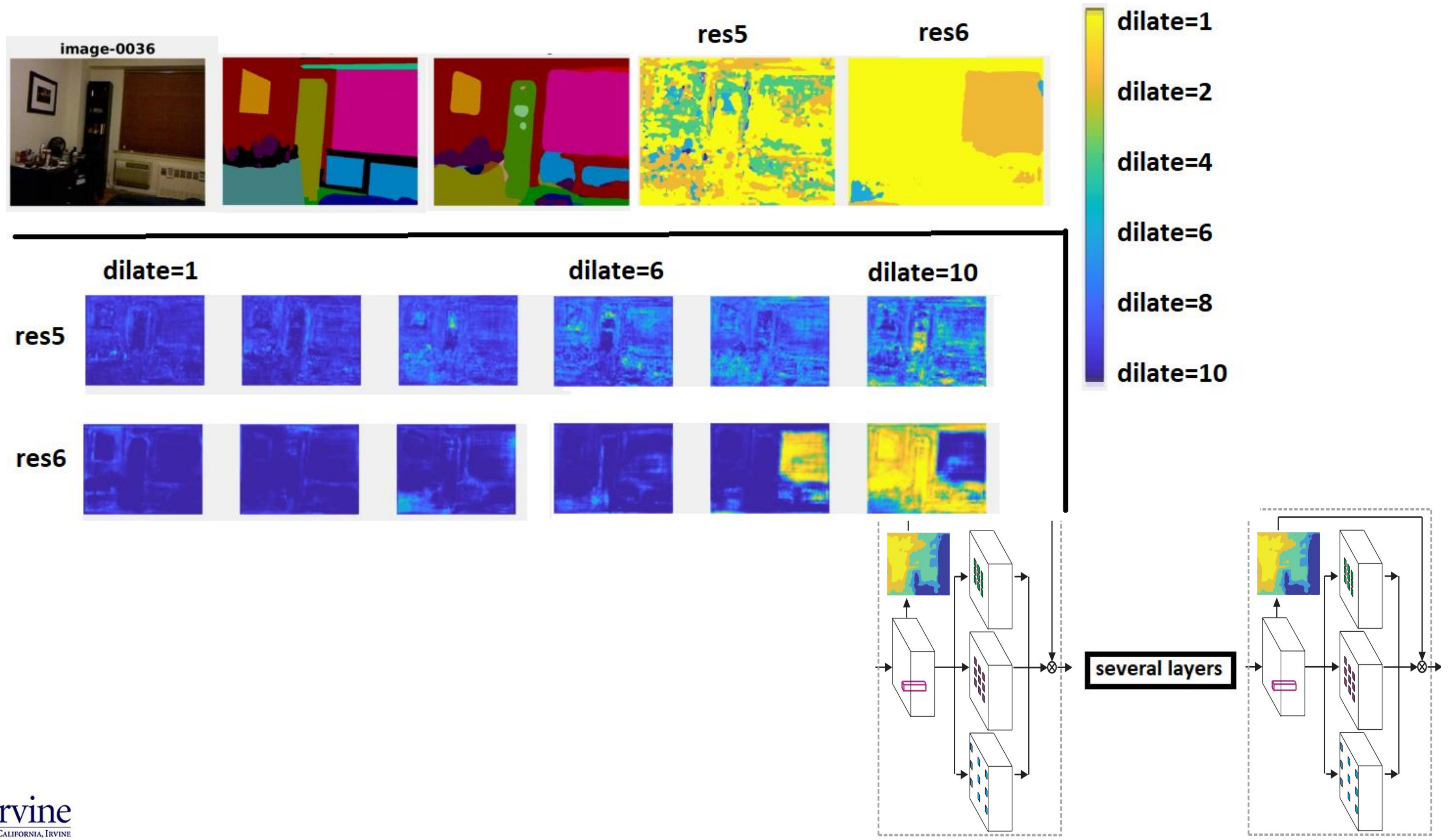
# Attention to Scale Again

## Qualitative Results -- res{5,6}



# Attention to Scale Again

Can we choose the region to process at specific scale, in stead of computing over the whole feature maps?

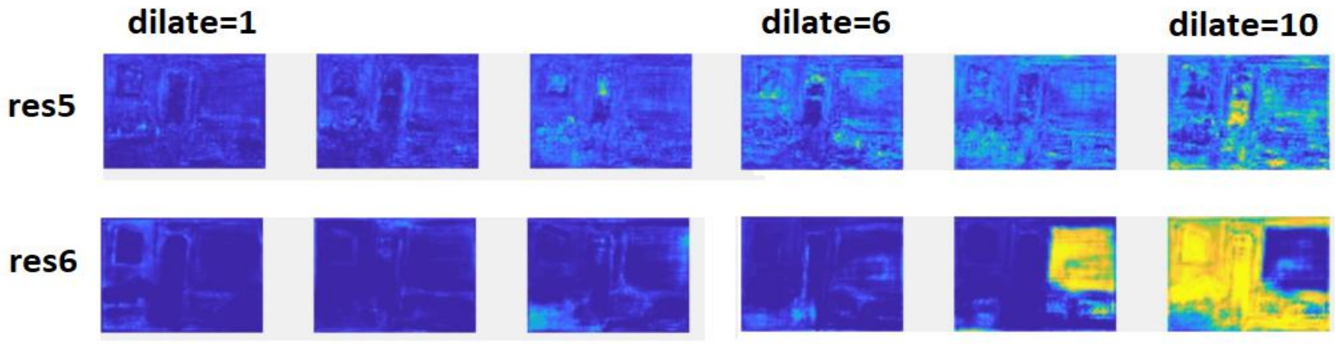


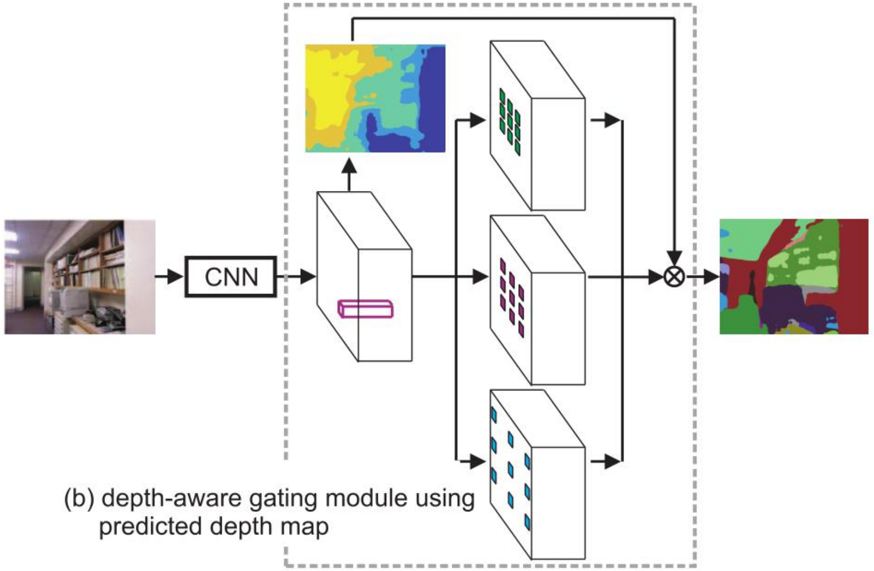

UCIrvine
University of California, Irvine

# Attention to Scale Again

Can we choose the region to process at specific scale, in stead of computing over the whole feature maps?

**Yes, we can! Just make them binary**.




(b) depth-aware gating module using predicted depth map

UCIrvine
University of California, Irvine

# Outline

# Pixel-wise Attentional Gating (PAG)

The difficulty is how to produce binary masks while still allowing for back-propagation for end-to-end training.

# Pixel-wise Attentional Gating (PAG)

using the Gumbel-Max trick for discrete (binary) masks

$$\text{Gumbel distribution if } m \equiv -\log(-\log(u))$$

$$\text{where } \; u \sim \mathcal{U}[0,1]$$

UCIrvine
University of California, Irvine

Gumbel, E.J.: Statistics of extremes. Courier Corporation (2012)

# Pixel-wise Attentional Gating (PAG)

using the Gumbel-Max trick for discrete (binary) masks

$$\text{Gumbel distribution if } m \equiv -\log(-\log(u))$$

$$\text{where } u \sim \mathcal{U}[0,1]$$

Let $g$ be a discrete random variable with probabilities

$$P(g = k) \propto a_k$$

UCIrvine
UNIVERSITY OF CALIFORNIA, IRVINE

Categorical reparameterization with gumbel-softmax, ICLR, 2017
The concrete distribution: A continuous relaxation of discrete random variables, ICLR, 2017

# Pixel-wise Attentional Gating (PAG)

using the Gumbel-Max trick for discrete (binary) masks

$$\text{Gumbel distribution if } m \equiv -\log(-\log(u))$$

$$\text{where } u \sim \mathcal{U}[0,1]$$

Let $g$ be a discrete random variable with probabilities

$$P(g = k) \propto a_k$$

let $\{m_k\}_{k=1,\ldots,K}$ be a sequence of i.i.d. Gumbel random variables

$$g = \operatorname*{argmax}_{k=1,\ldots,K}(\log \alpha_k + m_k)$$

Categorical reparameterization with gumbel-softmax, ICLR, 2017
The concrete distribution: A continuous relaxation of discrete random variables, ICLR, 2017

# Pixel-wise Attentional Gating (PAG)

using the Gumbel-Max trick for discrete (binary) masks

$$\cancel{g = \underset{k=1,\ldots,K}{\operatorname{argmax}} \left( \log \alpha_k + m_k \right)}$$

$$\mathbf{g} = softmax((\log(\boldsymbol{\alpha} + \mathbf{m}))/\tau)$$

$$\boldsymbol{\alpha} = [\alpha_1, \ldots, \alpha_K]$$

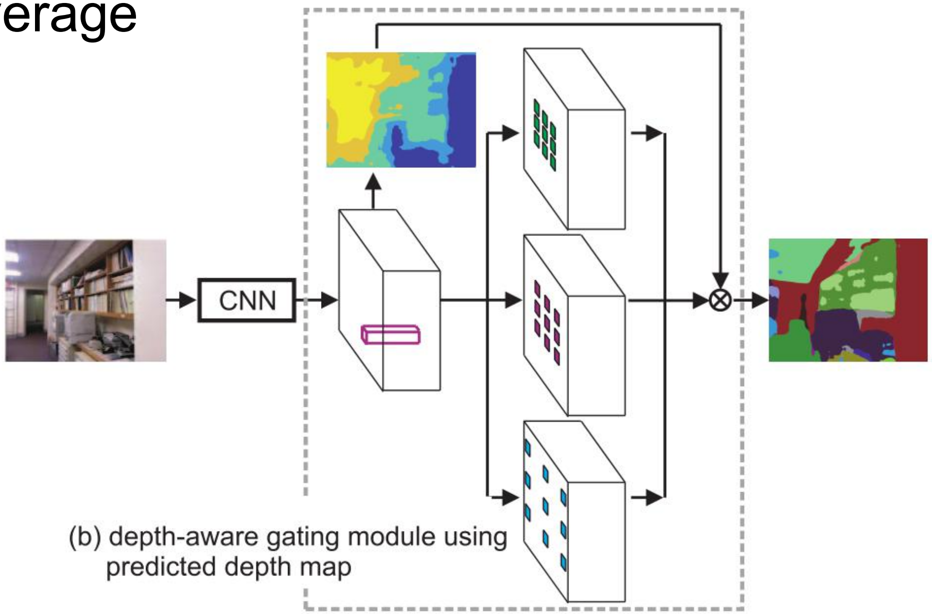
$$\mathbf{m} = [m_1, \ldots, m_K]$$

$\tau$ is the "temperature" parameter.

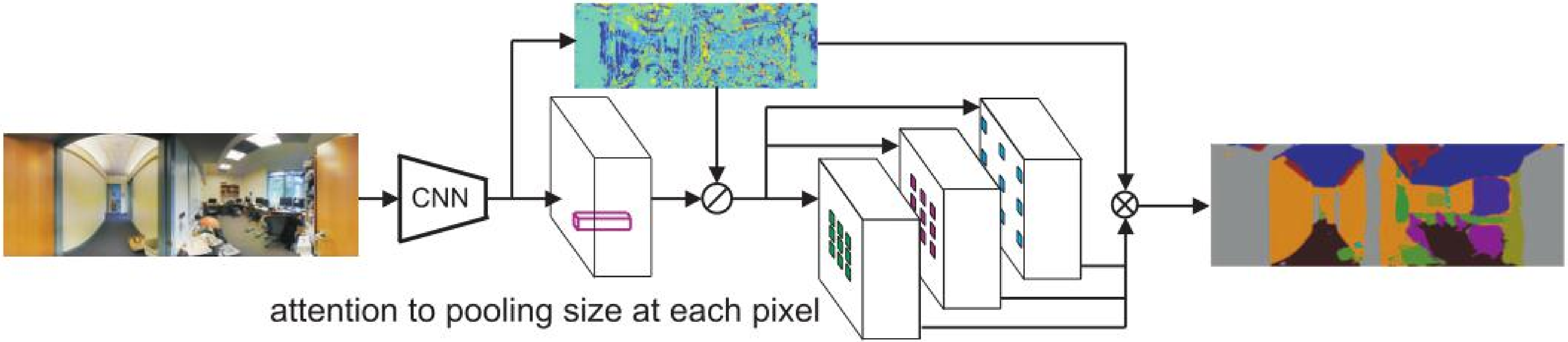
Categorical reparameterization with gumbel-softmax, ICLR, 2017
The concrete distribution: A continuous relaxation of discrete random variables, ICLR, 2017

# Pixel-wise Attentional Gating (PAG)

Multiplicative gating as weighted average

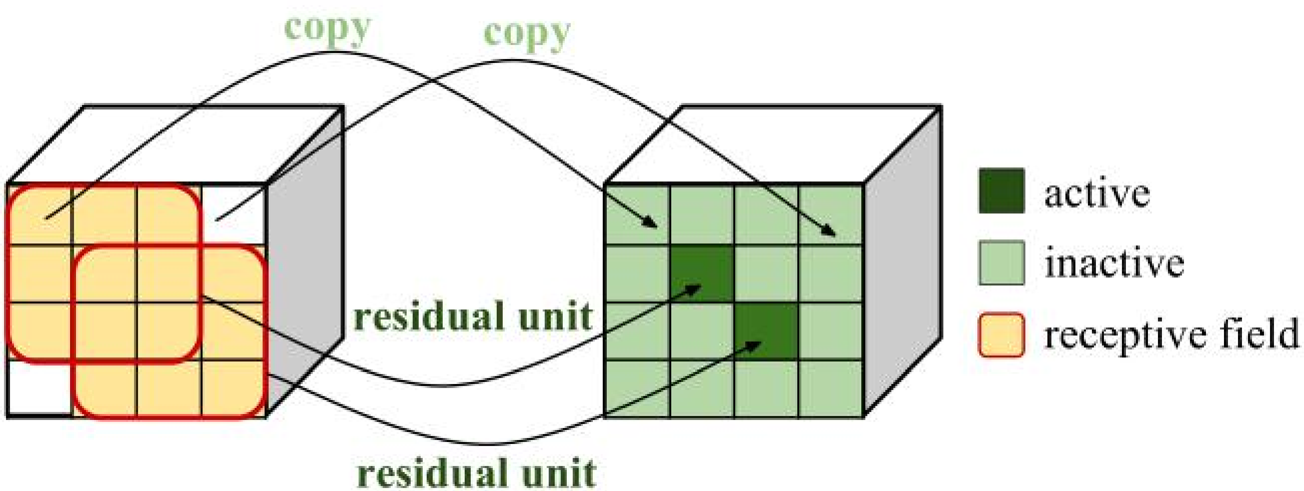


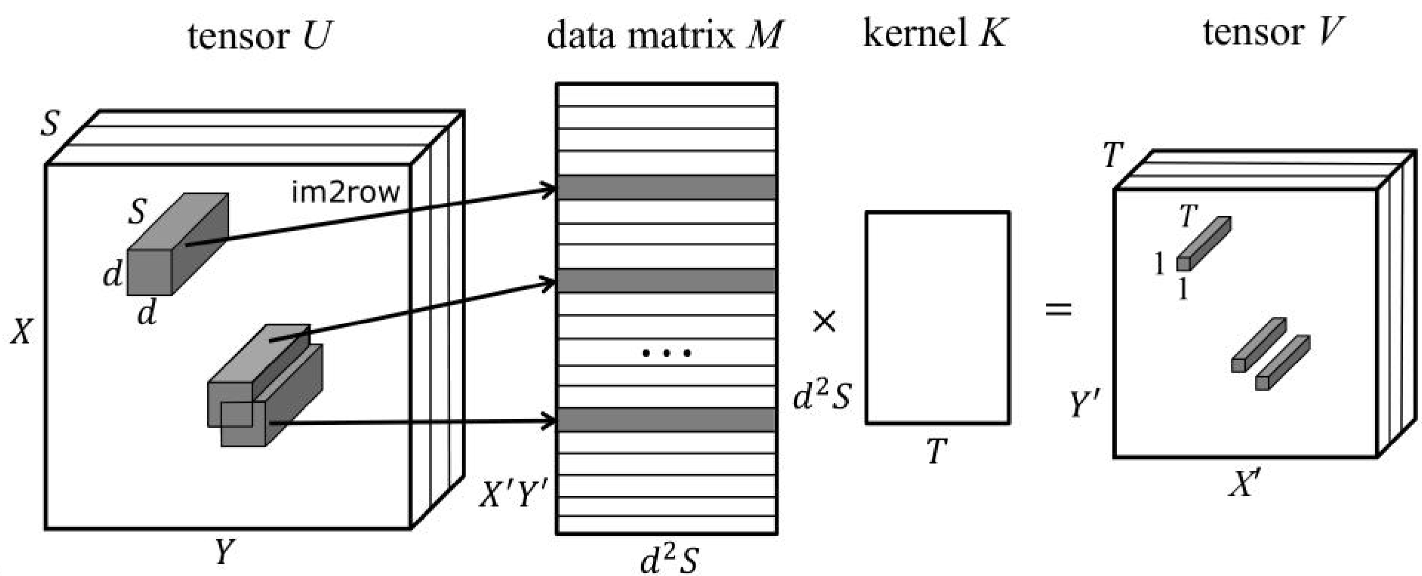

Attentional Gating to select



# Pixel-wise Attentional Gating (PAG)

Perforated convolution in low-level implementation



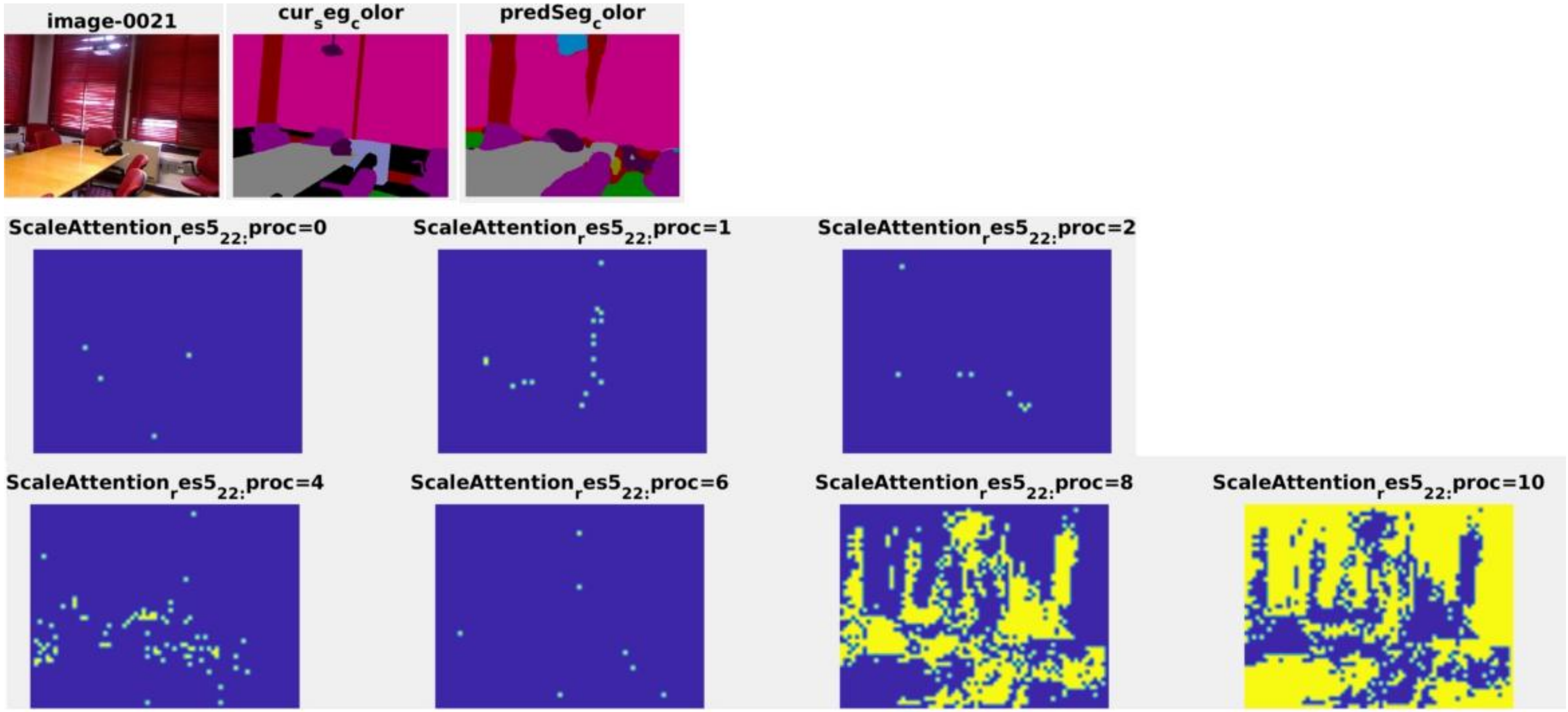

PerforatedCNNs: Acceleration through Elimination of Redundant Convolutions, NIPS 2016

UCIrvine
University of California, Irvine

# Pixel-wise Attentional Gating (PAG)

pooling using a set of 3 × 3-kernels with a set of dilation rates [0,1,2,4,6,8,10]

0 means the input feature is simply copied into the output feature map



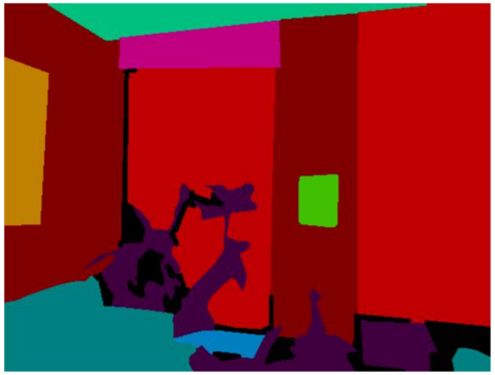

UCIrvine
University of California, Irvine

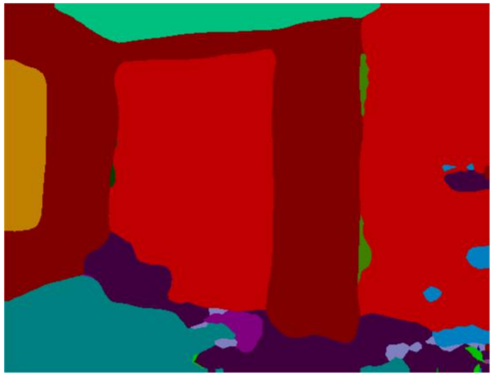
S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018

# Pixel-wise Attentional Gating (PAG)

semantic segmentation

| | NYUv2 [45] | | Stanford-2D-3D [47] | | Cityscapes [45] | |
|---|---|---|---|---|---|---|
| methods/metrics | IoU | pixel acc. | IoU | pixel acc. | IoU | iIoU |
| baseline | 42.1 | 71.1 | 79.5 | 92.1 | 73.8 | 54.7 |
| MP@Res5 (w-Avg.) | 46.3 | 73.4 | 83.7 | 93.6 | 75.8 | 56.9 |
| MP@Res5 (PAG) | 46.5 | 73.5 | 83.7 | 93.7 | 75.7 | 55.8 |



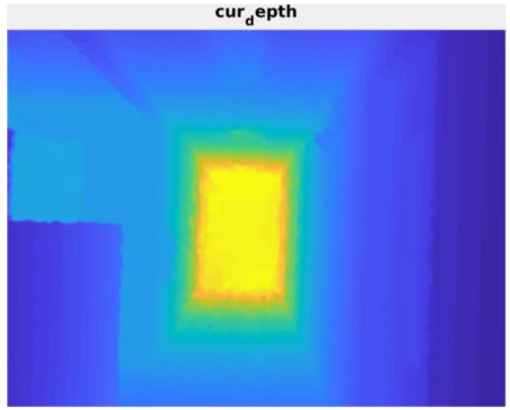


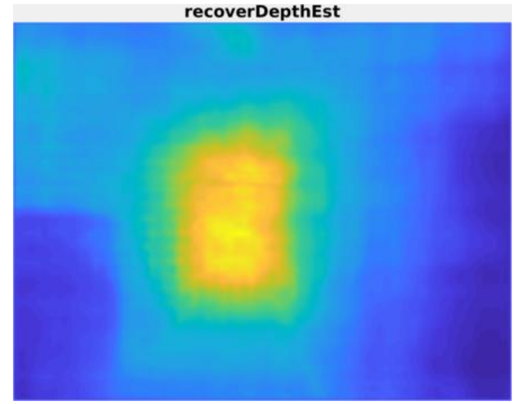
UCIrvine
University of California, Irvine

S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018

# Pixel-wise Attentional Gating (PAG)

monocular depth estimation



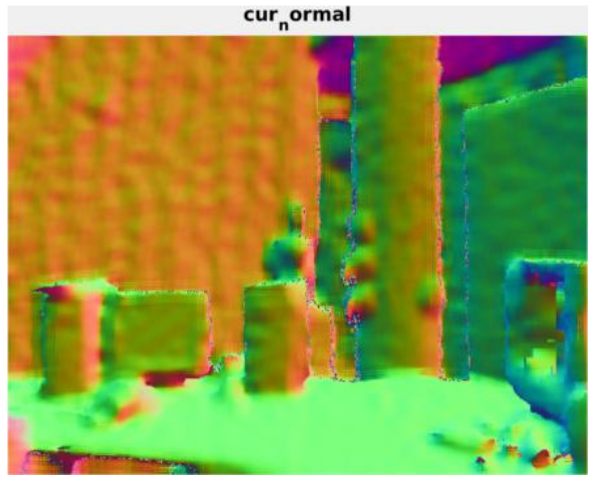

| | NYUv2 [45] | | | Stanford-2D-3D [47] | | | Cityscapes [45] | | |
|---|---|---|---|---|---|---|---|---|---|
| methods/metric ($\delta < \tau$) | $1.25$ | $1.25^2$ | $1.25^3$ | $1.25$ | $1.25^2$ | $1.25^3$ | $1.25$ | $1.25^2$ | $1.25^3$ |
| baseline | 71.1 | 93.2 | 98.5 | 73.1 | 92.1 | 97.5 | 29.0 | 53.8 | 75.8 |
| MP@Res5 (w-Avg.) | 74.5 | 94.4 | 98.8 | 77.5 | 94.1 | 97.9 | 33.7 | 65.9 | 76.9 |
| MP@Res5 (PAG) | 75.1 | 94.4 | 98.8 | 77.6 | 94.1 | 97.9 | 34.6 | 66.2 | 77.2 |

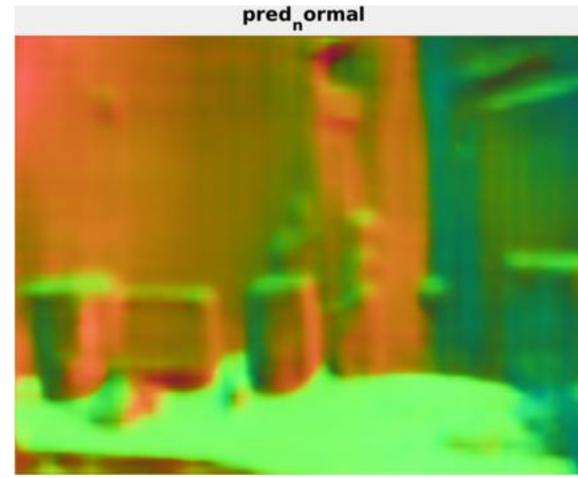


S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018

# Pixel-wise Attentional Gating (PAG)

surface normal estimation



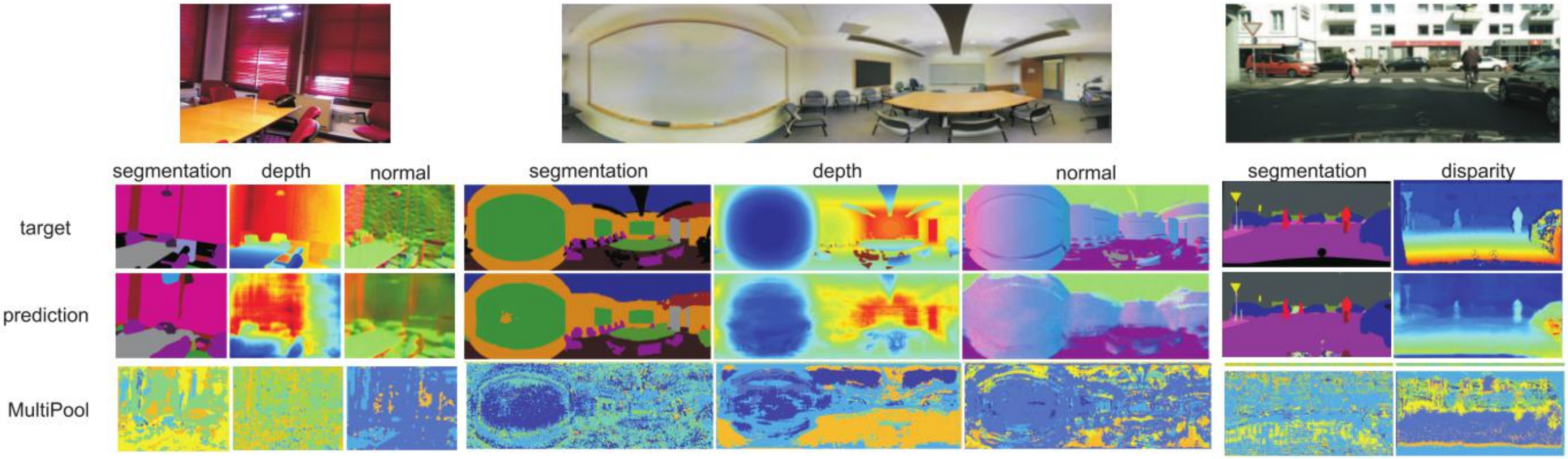

| | NYUv2 [45] | | | | Stanford-2D-3D [47] | | | |
|---|---|---|---|---|---|---|---|---|
| methods/metrics | ang. err.↓ | $11.25^\circ$ | $22.50^\circ$ | $30.00^\circ$ | ang. err.↓ | $11.25^\circ$ | $22.50^\circ$ | $30.00^\circ$ |
| baseline | 22.3 | 34.4 | 62.5 | 74.4 | 19.0 | 51.5 | 68.6 | 76.3 |
| MP@Res5 (w-Avg.) | 21.9 | 35.9 | 63.8 | 75.3 | 16.5 | 58.2 | 74.2 | 80.4 |
| MP@Res5 (PAG) | 21.7 | 36.1 | 64.2 | 75.5 | 16.5 | 58.3 | 74.2 | 80.4 |



S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018

# Pixel-wise Attentional Gating (PAG)

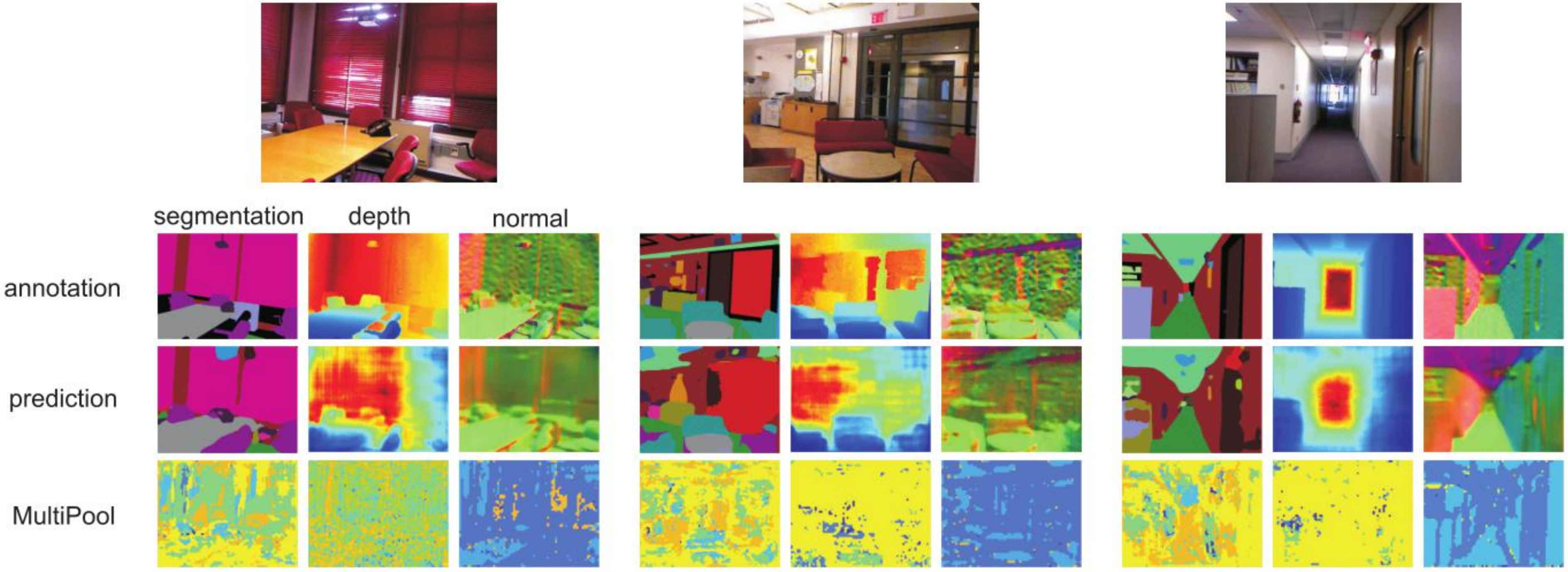
Visual summary of three tasks on three different datasets




S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018

# Pixel-wise Attentional Gating (PAG)

More qualitatively results on NYU-depth-v2

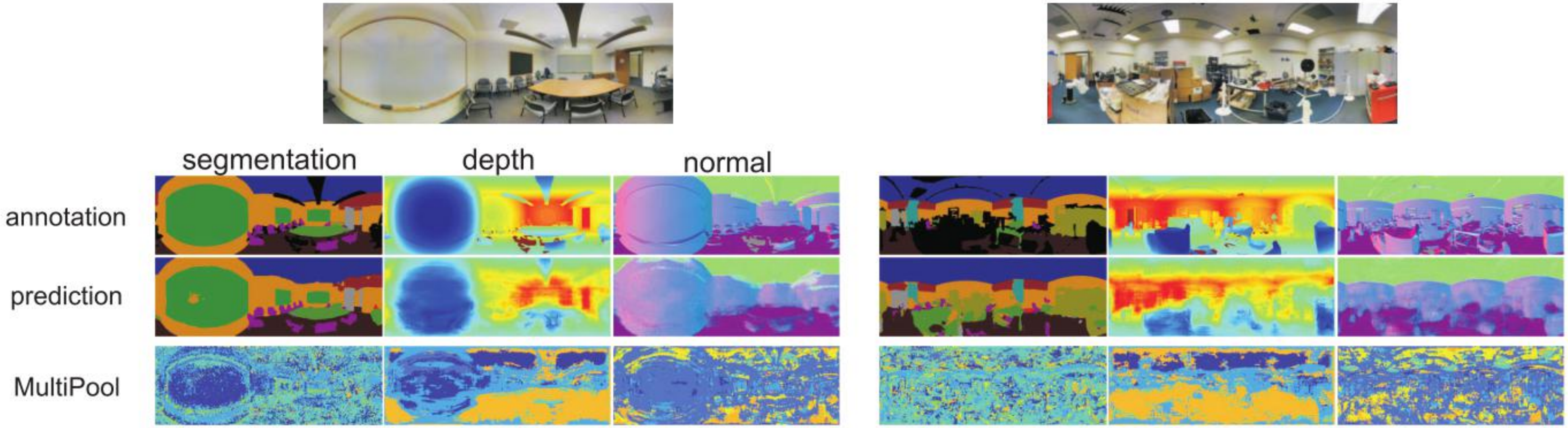



UCIrvine University of California, Irvine

S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018

# Pixel-wise Attentional Gating (PAG)

More qualitatively results on Stanford-2D-3D dataset

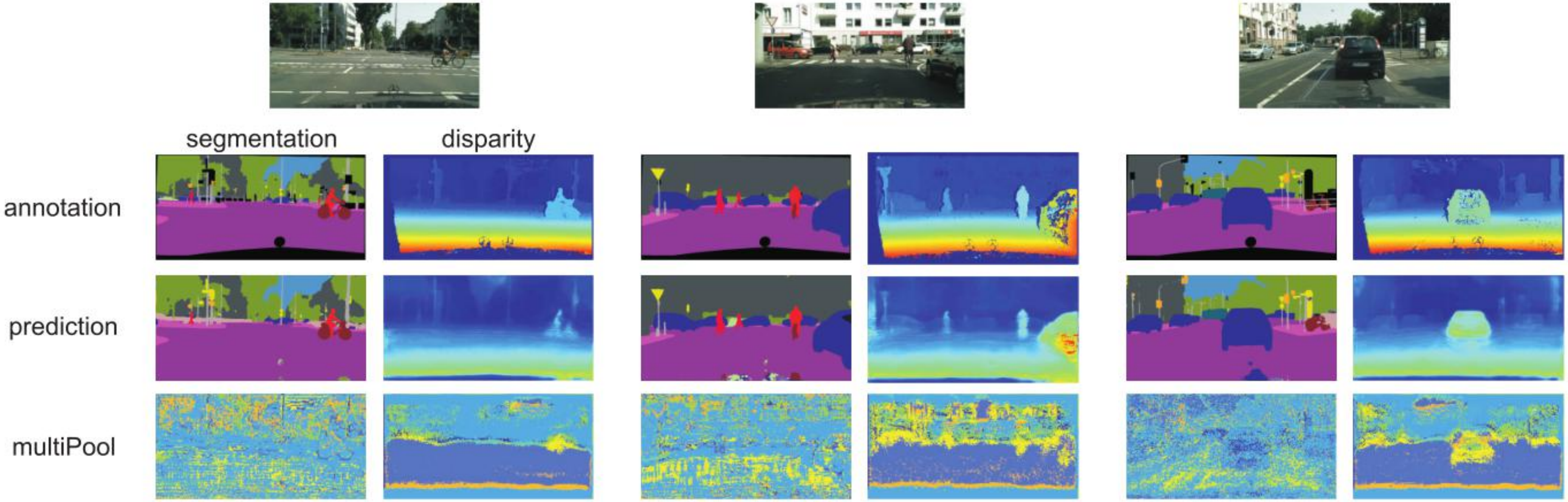



UCIrvine
University of California, Irvine

S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018

# Pixel-wise Attentional Gating (PAG)

More qualitatively results on Cityscapes




S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018

# Pixel-Level Dynamic Routing

PAG achieves better performance while maintaining the computation.



S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018

# Pixel-Level Dynamic Routing

PAG achieves better performance while maintaining the computation.

It also offers parsimonious inference under limited computation budget.



S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018

# Outline

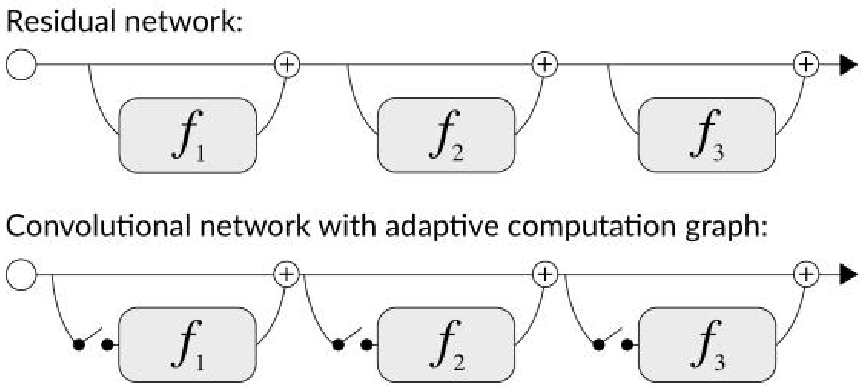
1. Background
2. Attention to Perspective: Depth-aware Pooling Module
3. Recurrent Refining with Perspective Understanding in the Loop
4. Attention to Perspective Again
5. Pixel-wise Attentional Gating (PAG)
6. Pixel-Level Dynamic Routing
7. Conclusion

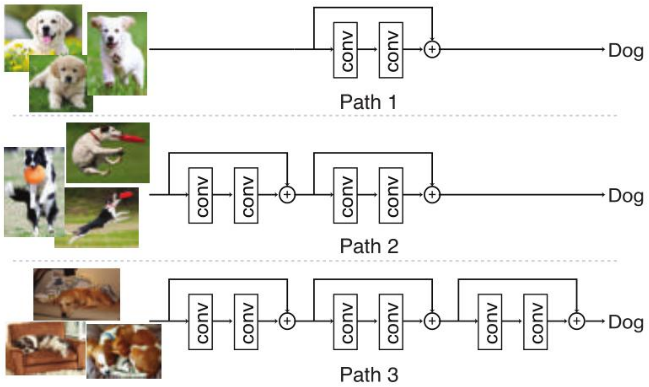
# Dynamic Computation

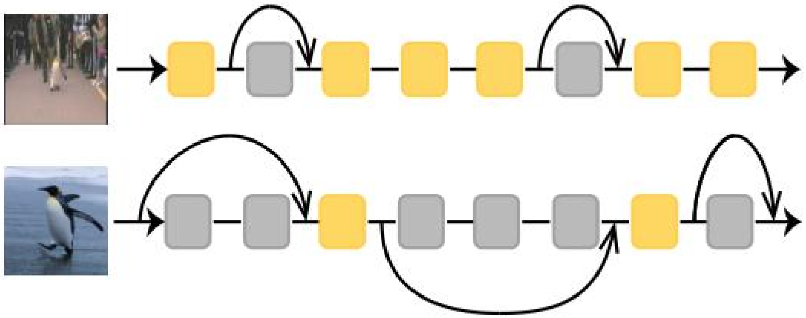
Parsimonious inference as dynamic computation

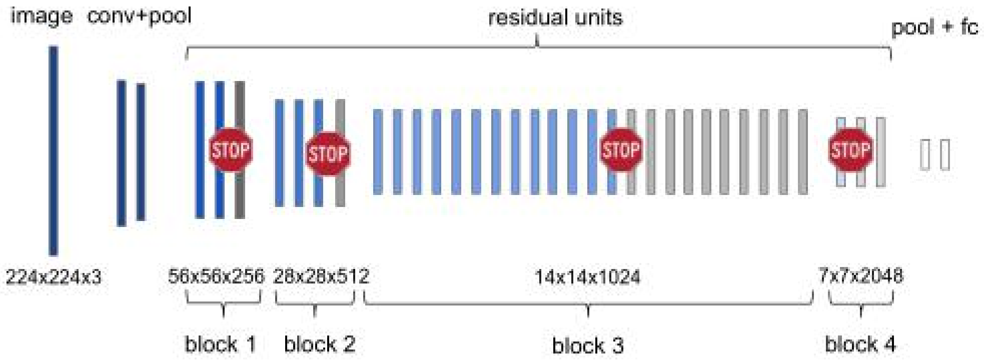
# Dynamic Computation

Parsimonious inference as dynamic computation







[1] BlockDrop: Dynamic Inference Paths in Residual Networks
[2] Convolutional Networks with Adaptive Computation Graphs
[3] SkipNet: Learning Dynamic Routing in Convolutional Networks
[4] Spatially Adaptive Computation Time for Residual Networks


UCIrvine
University of California, Irvine

# Pixel-Level Dynamic Routing

More generally, can we allocate dynamic computation time to each pixel of each image instance?

# Pixel-Level Dynamic Routing

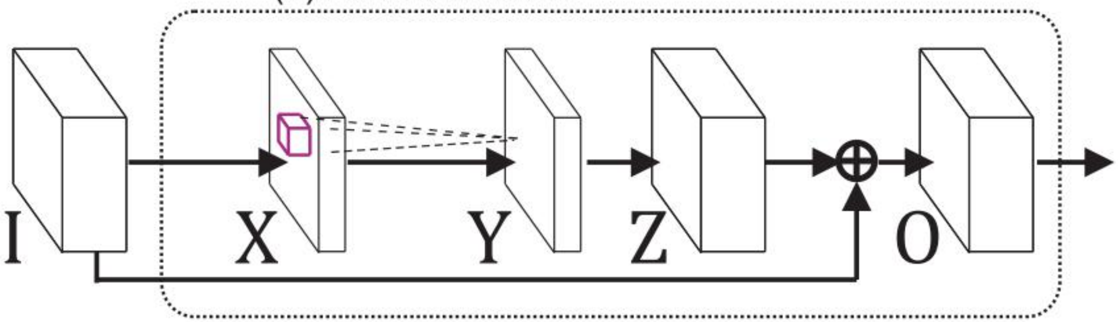
More generally, can we allocate dynamic computation time to each pixel of each image instance?

**PAG can do this!**

# Dynamic Computation

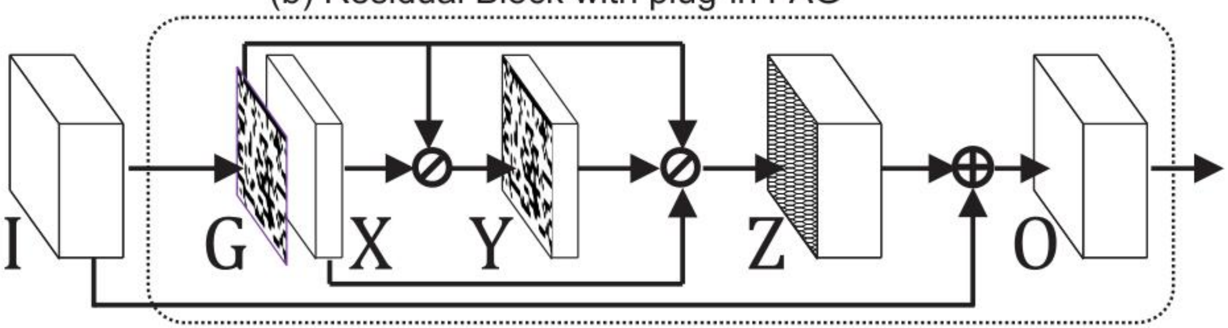
## Inserting PAG at each residual block for fine-tuning

(a) Residual Block




$$\mathbf{X} = \mathcal{F}^1(\mathbf{I})$$
$$\mathbf{Y} = \mathcal{F}^2(\mathbf{X})$$
$$\mathbf{Z} = \mathcal{F}^3(\mathbf{Y})$$
$$\mathbf{O} = \mathbf{I} + \mathbf{Z}$$

(b) Residual Block with plug-in PAG




$$\mathbf{X} = \mathcal{F}^1(\mathbf{I}), \quad \mathbf{G} = \mathcal{G}(\mathbf{I})$$
$$\mathbf{Y} = \mathcal{F}^2_{\mathbf{G}}(\mathbf{X})$$
$$\mathbf{Z} = \mathcal{F}^3_{\mathbf{G}}(\bar{\mathbf{G}} \odot \mathbf{X} + \mathbf{G} \odot \mathbf{Y})$$
$$\mathbf{O} = \mathbf{I} + \mathbf{Z}$$



S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018

# Dynamic Computation

sparse binary masks for perforated convolution

For a binary mask $\mathbf{G} \in \{0,1\}^{H \times W}$

we compute the empirical sparsity

$$g = \frac{1}{H*W} \sum_{h,w}^{H,W} \mathbf{G}_{h,w}$$

Using KL-divergence term for sparse masks.

$$KL(\rho||g) \equiv \rho \log(\frac{\rho}{g}) + (1-\rho) \log(\frac{1-\rho}{1-g})$$

jointly minimize

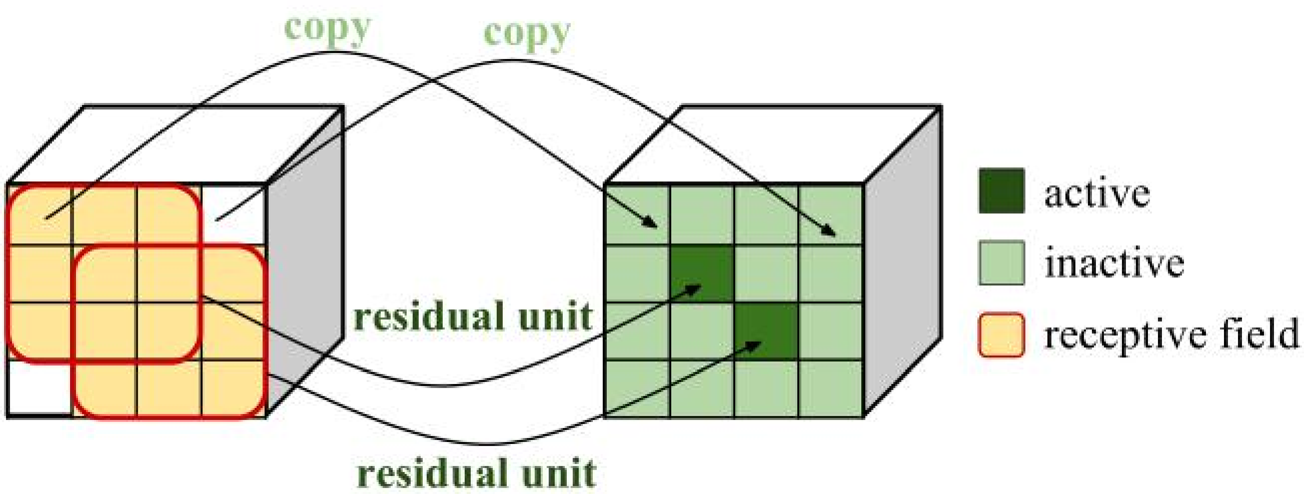
$$\ell = \ell_{task} + \lambda \sum_{l=1}^{L} KL(\rho||g_l)$$

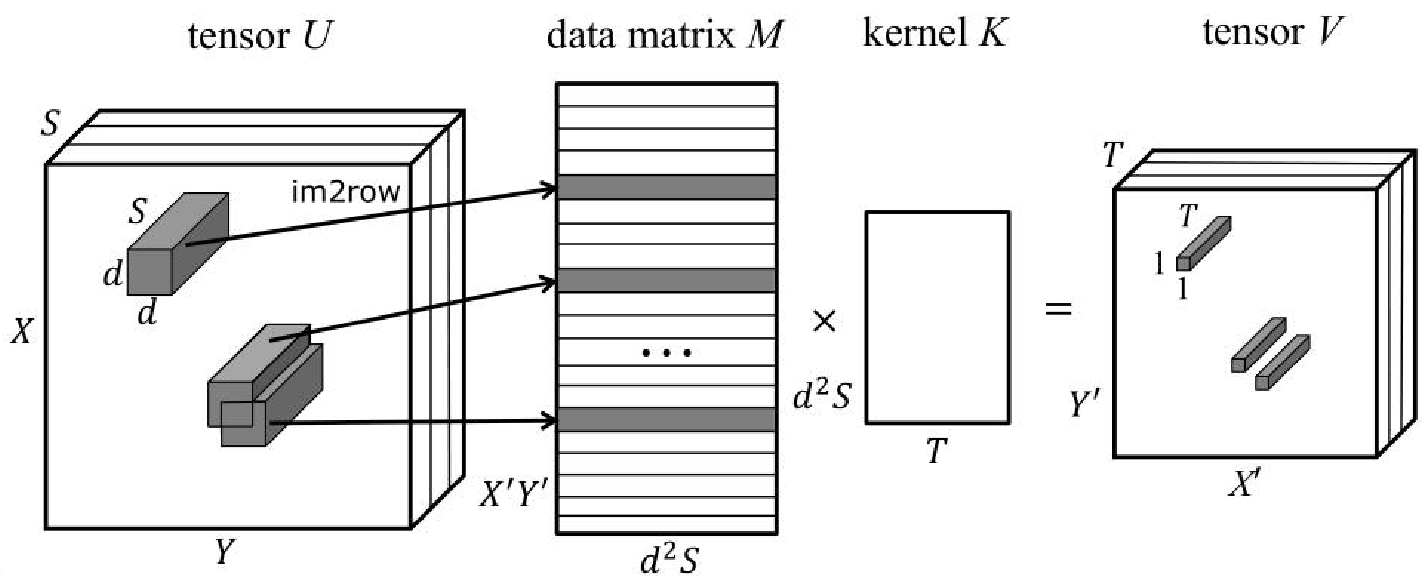


S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018

# Pixel-wise Attentional Gating (PAG)

Perforated convolution in low-level implementation




UCIrvine
UNIVERSITY OF CALIFORNIA, IRVINE

PerforatedCNNs: Acceleration through Elimination of Redundant Convolutions, NIPS 2016

# Dynamic Computation

Semantic segmentation on NYU-depth-v2 dataset

**Table 2.** Computational parsimony compared with truncated ResNet and models learning to drop/skip whole layers. Evaluation is performed on NYUv2 dataset for semantic segmentation.

| hyper param. | FLOPs | consumption | truncated | | layer-skipping | | MP@Res5 (PAG) | |
|---|---|---|---|---|---|---|---|---|
| $\rho$ | $1e10$ | % | IoU | acc. | IoU | acc. | IoU | acc. |
| $\rho = 0.5$ | 6.29 | 67.69 | 36.30 | 67.36 | 37.78 | 67.31 | 40.89 | 69.44 |
| $\rho = 0.7$ | 8.27 | 86.20 | 37.69 | 67.44 | 39.84 | 69.00 | 43.61 | 71.41 |
| $\rho = 0.9$ | 8.95 | 93.36 | 40.29 | 69.66 | 41.27 | 70.01 | 45.75 | 72.93 |
| $\rho = 1.0$ | 9.63 | 100.00 | — | — | — | — | 46.52 | 73.50 |

$$\ell = \ell_{task} + \lambda \sum_{l=1}^{L} KL(\rho || g_l)$$

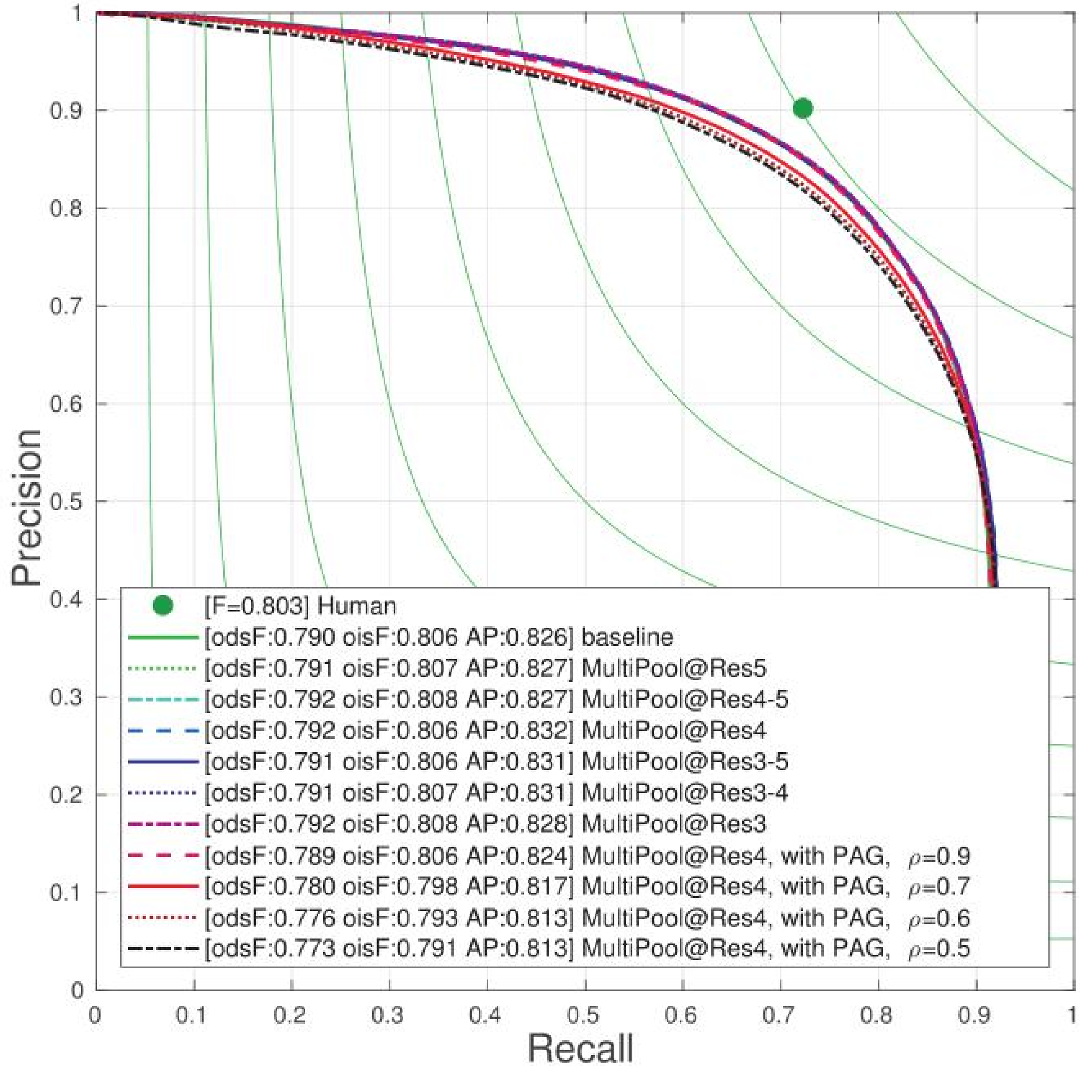


S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018

# Dynamic Computation

Boundary detection on BSDS500

$$\ell = \ell_{task} + \lambda \sum_{l=1}^{L} KL(\rho \| g_l)$$

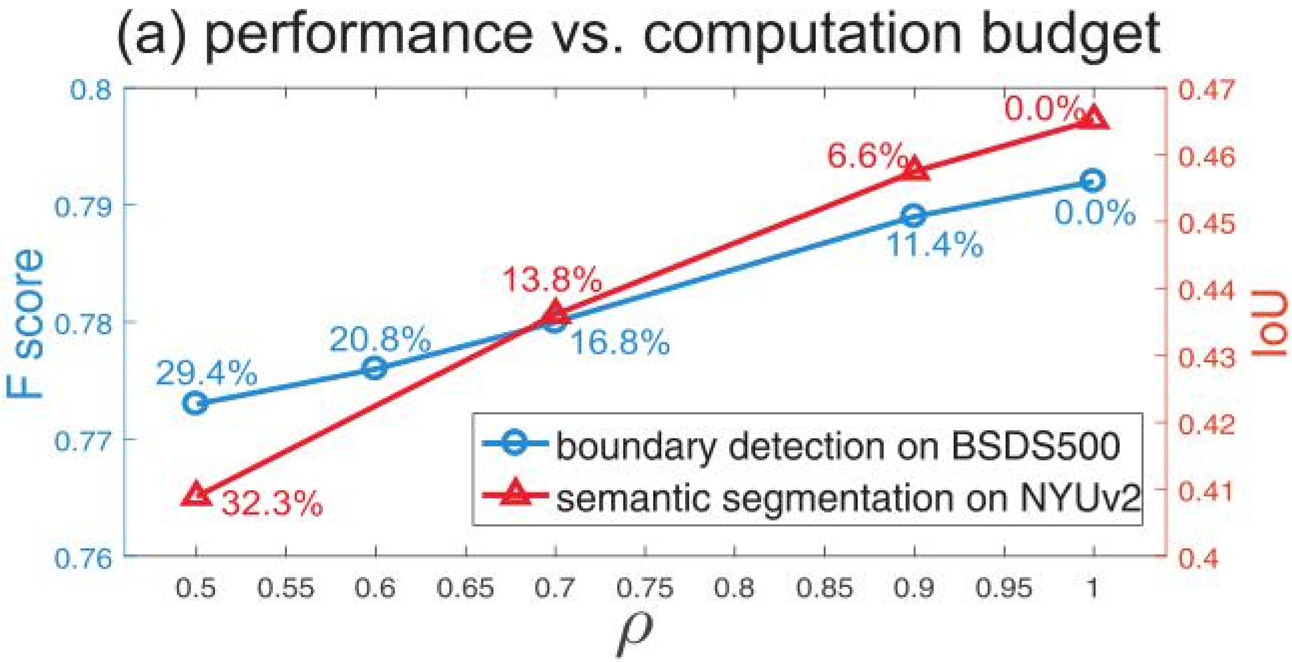



S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018

# Dynamic Computation

Semantic segmentation on NYU-depth-v2

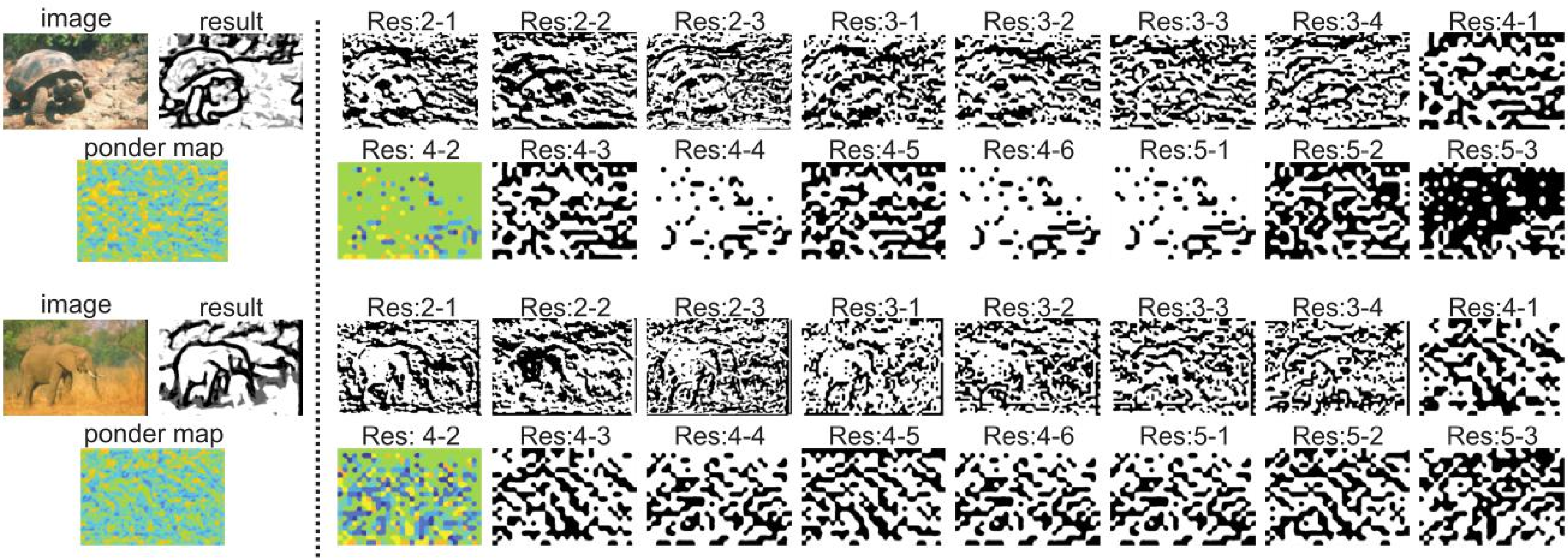
Boundary detection on BSDS500




S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018

# Dynamic Computation

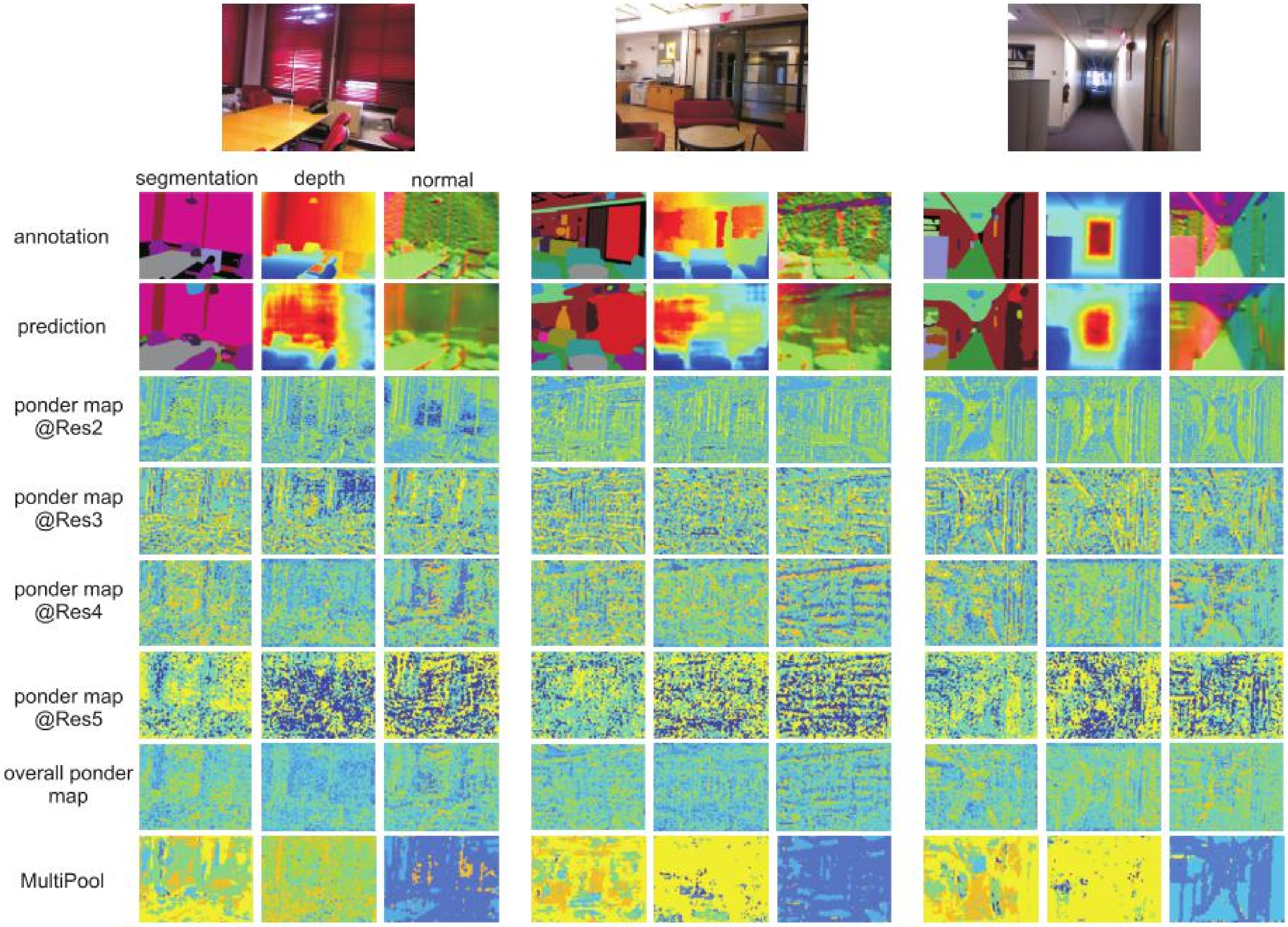
Boundary detection on BSDS500 dataset




S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018

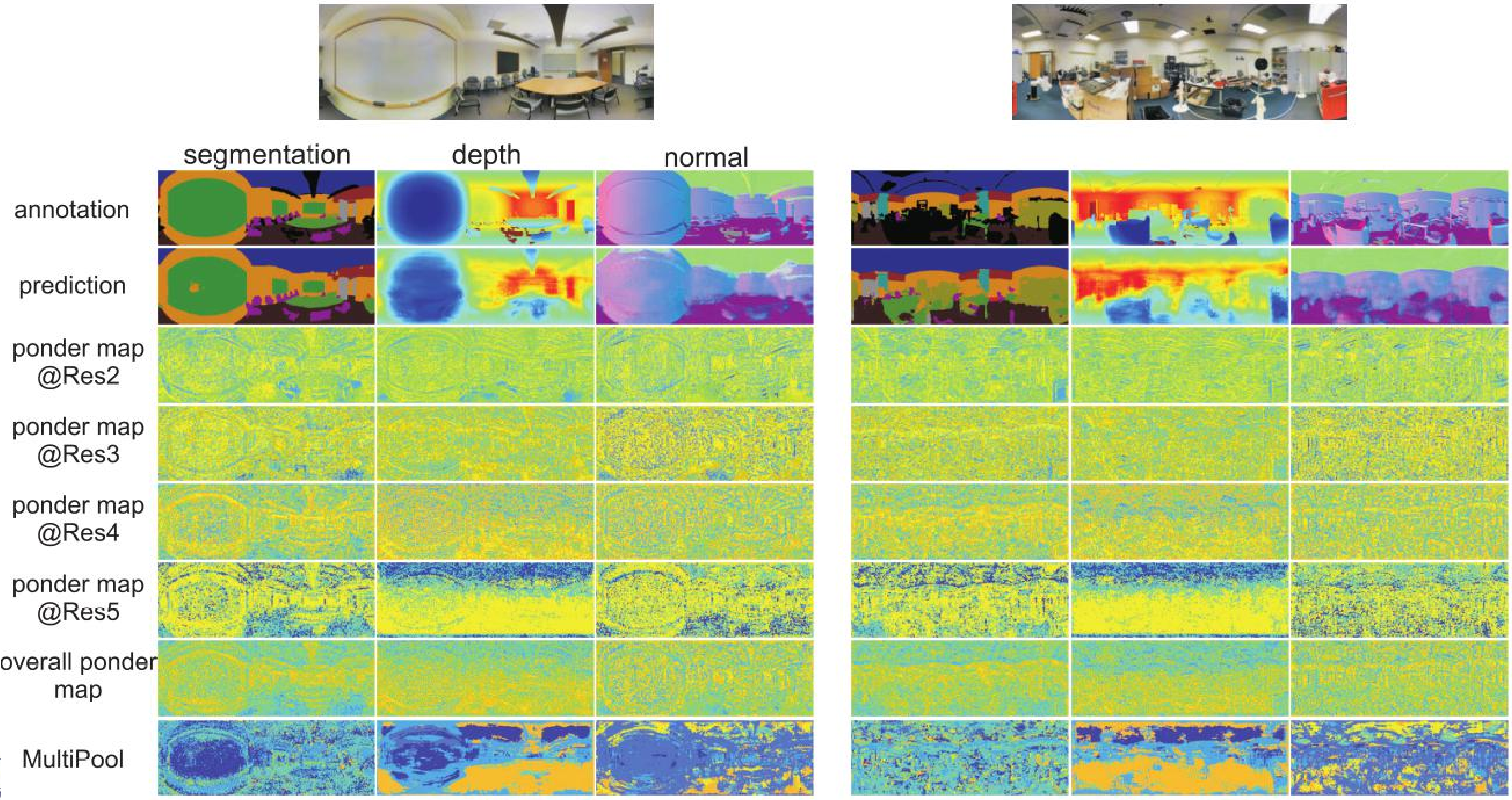
# Dynamic Computation

NYU-depth-v2 dataset



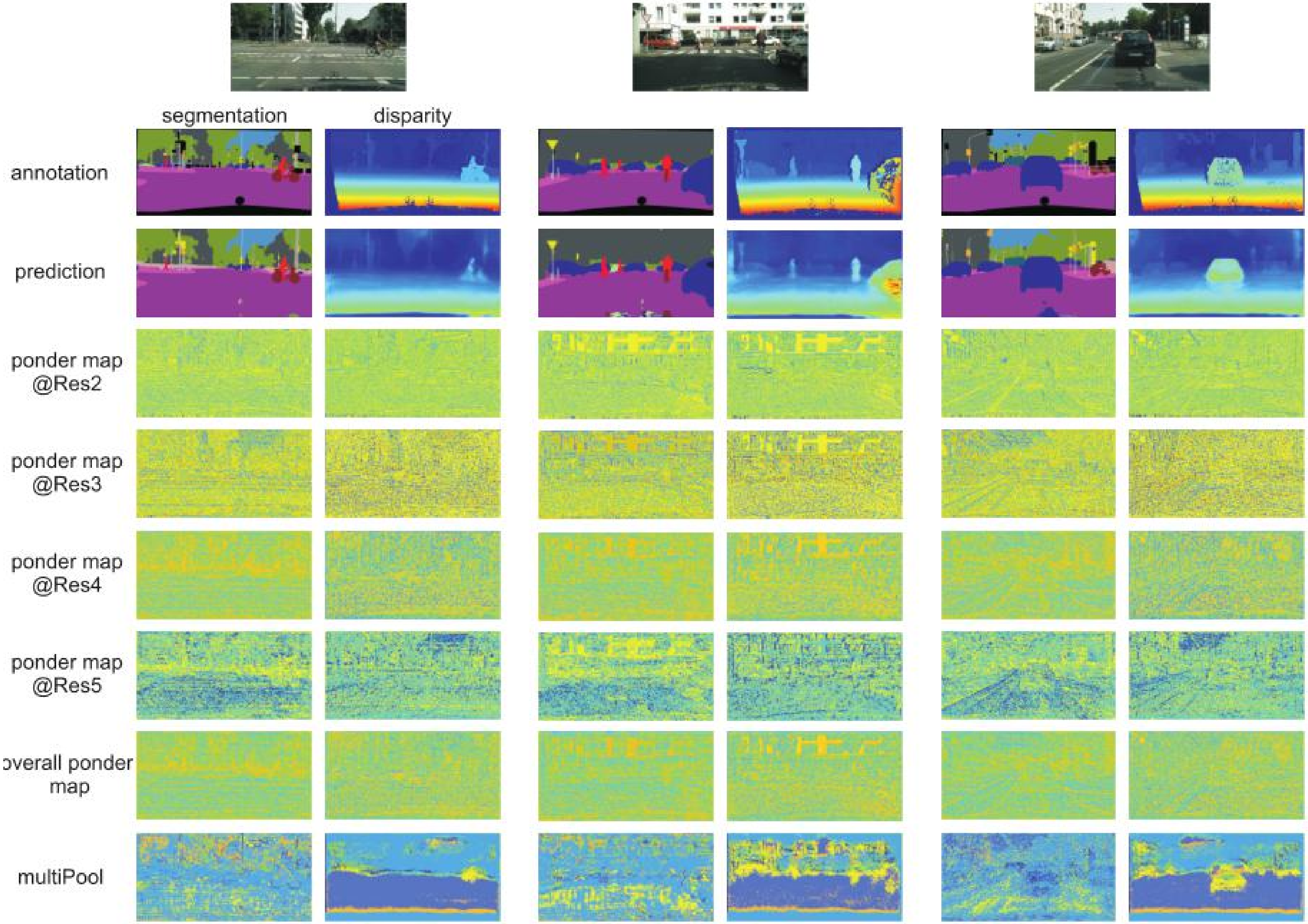
# Dynamic Computation

Stanford-2D-3D dataset



# Dynamic Computation

Cityscapes dataset




UCIrvine
UNIVERSITY OF CALIFORNIA, IRVINE

# Outline

1. Background
2. Attention to Perspective: Depth-aware Pooling Module
3. Recurrent Refining with Perspective Understanding in the Loop
4. Pixel-wise Attentional Gating (PAG)
5. Pixel-Level Dynamic Routing
6. Conclusion

# Conclusion and Future Work

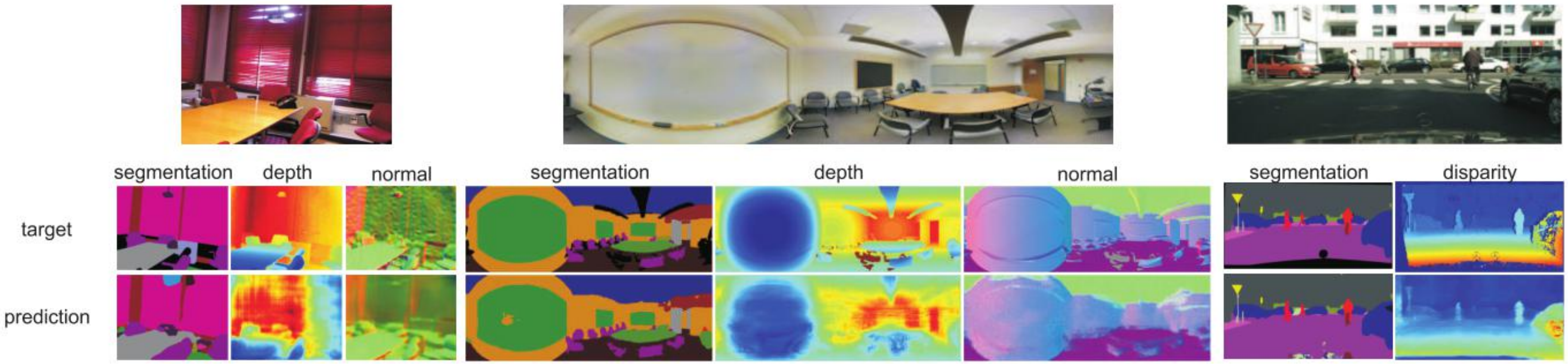
1. Scene parsing means more than semantic segmentation, geometry and inter-object relation

semantic segmentation (*what*)
localization (*where*)
support, surface normal (*relation*)

# Conclusion and Future Work

1. Scene parsing means more than semantic segmentation, geometry and inter-object relation

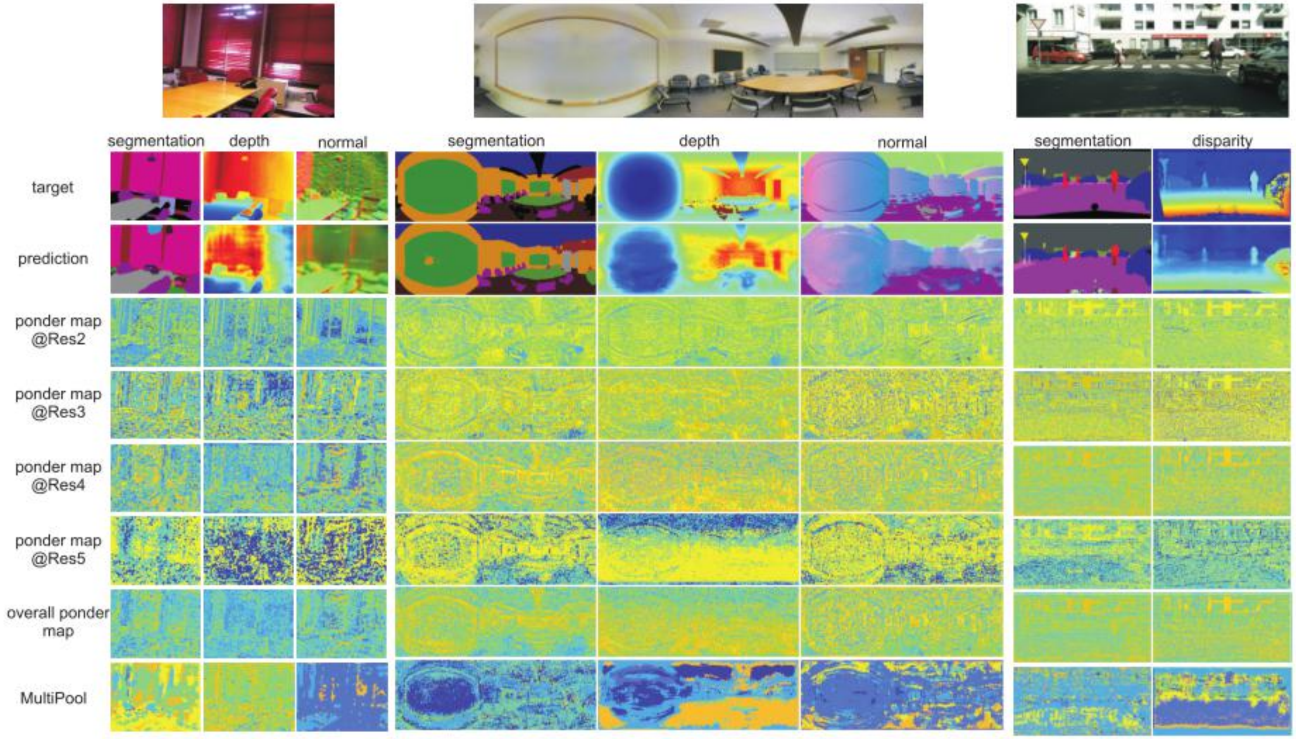
2. Potentially unified model for all these tasks




But for learning knowledge from different tasks? How to wire them up?

# Conclusion and Future Work

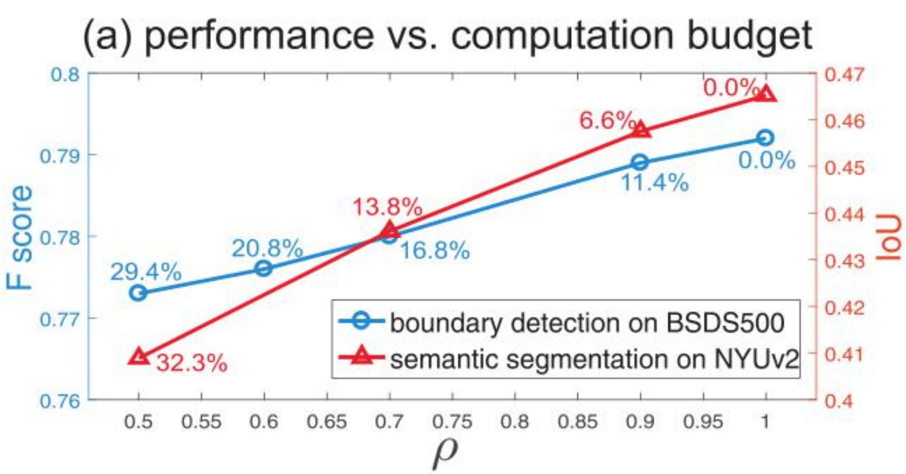
1. Scene parsing means more than semantic segmentation, geometry and inter-object relation
2. Potentially unified model for all these tasks
3. Pixel-wise Attentional Gating unit (PAG) allocates dynamic computation for pixels; it is general, agnostic to architectures and problems.




UCIrvine
University of California, Irvine

# Conclusion and Future Work

1. Scene parsing means more than semantic segmentation, geometry and inter-object relation
2. Potentially unified model for all these tasks
3. Pixel-wise Attentional Gating unit (PAG) allocates dynamic computation for pixels; it is general, agnostic to architectures and problems.
4. PAG reduces computation by **10%** without noticeable loss in accuracy and performance degrades gracefully when imposing stronger computational constraints.




UCIrvine
University of California, Irvine

# Conclusion and Future Work

1. Scene parsing means more than semantic segmentation, geometry and inter-object relation
2. Potentially unified model for all these tasks
3. Pixel-wise Attentional Gating unit (PAG) allocates dynamic computation for pixels; it is general, agnostic to architectures and problems.
4. PAG reduces computation by 10% without noticeable loss in accuracy and performance degrades gracefully when imposing stronger computational constraints.

But for real-time inference...?

UCIrvine
University of California, Irvine

# Thanks

Q&A

Shu Kong Charless Fowlkes

UCIrvine
UNIVERSITY OF CALIFORNIA, IRVINE