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

Shu Kong
CS, ICS, UCI
Image Understanding --> Scene Parsing
Scene Parsing

**semantic segmentation**
classifying each pixel into one of defined categories
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
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
None of them consider “perspective” explicitly.
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
Attention to Perspective: Depth-aware Pooling

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

Attention to Perspective: Depth-aware Pooling
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".
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?
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).
Atrous convolution (skipping/inserting zero)
a trous (French) -- holes (English)

\[ y[i] = \sum_{k=1}^{K} x[i + r \cdot k] w[k] \]
2D atrous convolution of different dilate rates.
quantize the depth into five scales with dilate rates \{1, 2, 4, 8, 16\}
Alternatively, learning depth estimator, and testing without depth quantized depth scale classification softmax weight for multiplicative gating
Alternatively, learning depth estimator, and testing without depth reliable monocular depth estimation

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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$\delta &lt;$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.25</td>
<td>0.542</td>
<td>0.614</td>
<td>0.614</td>
<td>0.769</td>
<td>0.811</td>
<td>0.809</td>
<td>0.816</td>
</tr>
<tr>
<td>1.25$^2$</td>
<td>0.829</td>
<td>0.883</td>
<td>0.888</td>
<td>0.950</td>
<td>0.953</td>
<td>0.945</td>
<td>0.950</td>
</tr>
<tr>
<td>1.25$^3$</td>
<td>0.940</td>
<td>0.971</td>
<td>0.972</td>
<td>0.988</td>
<td>0.988</td>
<td>0.986</td>
<td>0.989</td>
</tr>
</tbody>
</table>

\[
\ell_{\text{depthReg}}(D, D^*) = \frac{1}{|M|} \sum_{(i,j) \in M} \left\| \log(D_{ij}) - \log(D_{ij}^*) \right\|_2^2
\]
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)
many possibilities to explore --

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

metric: Intersection over Union (IoU)

using the ground-truth disparity map, 5 discrete bins for 5 scales \{1,2,4,8,16\}
Cityscapes dataset
metric: Intersection over Union (IoU)
using the ground-truth disparity map, 5 discrete bins for 5 scales \{1, 2, 4, 8, 16\}

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

<table>
<thead>
<tr>
<th></th>
<th>deepLab (baseline)</th>
<th>avg.</th>
<th>gtDepth tiedKernel</th>
<th>gtDepth untied Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoU</td>
<td>0.738</td>
<td>0.747</td>
<td>0.748</td>
<td>0.753</td>
</tr>
</tbody>
</table>
train depth estimation branch to see if the estimated depth also helps

<table>
<thead>
<tr>
<th></th>
<th>deepLab (baseline)</th>
<th>avg.</th>
<th>gtDepth tiedKernel</th>
<th>gtDepth untied Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoU</td>
<td>0.738</td>
<td>0.747</td>
<td>0.748</td>
<td>0.753</td>
</tr>
</tbody>
</table>
Cityscapes dataset

metric: Intersection over Union (IoU)

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

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

<table>
<thead>
<tr>
<th></th>
<th>deepLab (baseline)</th>
<th>avg.</th>
<th>gtDepth tiedKernel</th>
<th>gtDepth untied Kernel</th>
<th>predDepth untied Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoU</td>
<td>0.738</td>
<td>0.747</td>
<td>0.748</td>
<td>0.753</td>
<td><strong>0.759</strong></td>
</tr>
</tbody>
</table>
Cityscapes dataset

metric: Intersection over Union (IoU)

using the ground-truth disparity map, 5 discrete bins for 5 scales \(\{1, 2, 4, 8, 16\}\)

**Why better?**

<table>
<thead>
<tr>
<th></th>
<th>deepLab (baseline)</th>
<th>avg.</th>
<th>gtDepth tiedKernel</th>
<th>gtDepth untied Kernel</th>
<th>predDepth untied Kernel</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoU</td>
<td>0.738</td>
<td>0.747</td>
<td>0.748</td>
<td>0.753</td>
<td><strong>0.759</strong></td>
</tr>
</tbody>
</table>
many possibilities to explore --

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

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

2. averaging the feature vs. attention vs. depth-aware gating
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)
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

Qualitative Results -- street images
Qualitative Results -- panorama images

Depth-aware pooling module
Depth-aware pooling module

Good enough?

input panorama

ground-truth annotation

depth-adapted output

raw depth

quantized depth

depth adaptation
Recurrent Refining with Perspective Understanding in the Loop

Recurrent Refining Module

input panorama  ground-truth annotation  depth-adapted output
raw depth  quantized depth  depth adaptation
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
Recurrently refining the results by adapting the predicted depth

Recurrent Refinement Module

input image

CNN backbone

feed-forward pathway

depth-aware gating module

recurrent module

loop-0, IoU=0.418
loop-1, IoU=0.427
loop-2, IoU=0.431

output difference

depth-aware gating module using predicted depth map
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
Recurrently refining the results by adapting the predicted depth

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IoU</td>
<td>pixel acc.</td>
<td>IoU</td>
<td>pixel acc.</td>
</tr>
<tr>
<td>baseline</td>
<td>0.406</td>
<td>0.703</td>
<td>0.402</td>
<td>0.776</td>
</tr>
<tr>
<td>w/ gt-depth</td>
<td>0.413</td>
<td>0.708</td>
<td>0.422</td>
<td>0.787</td>
</tr>
<tr>
<td>w/ pred-depth</td>
<td>0.418</td>
<td>0.711</td>
<td>0.423</td>
<td>0.789</td>
</tr>
<tr>
<td>loop1 w/o depth</td>
<td>0.419</td>
<td>0.706</td>
<td>0.432</td>
<td>0.793</td>
</tr>
<tr>
<td>loop1 w/ gt-depth</td>
<td>0.425</td>
<td>0.711</td>
<td>0.439</td>
<td>0.798</td>
</tr>
<tr>
<td>loop1 w/ pred-depth</td>
<td>0.427</td>
<td>0.712</td>
<td>0.440</td>
<td>0.798</td>
</tr>
<tr>
<td>loop2</td>
<td>0.431</td>
<td>0.713</td>
<td>0.443</td>
<td>0.799</td>
</tr>
<tr>
<td>loop2 (test-aug)</td>
<td>0.445</td>
<td>0.721</td>
<td>0.451</td>
<td>0.803</td>
</tr>
<tr>
<td>DeepLab [6]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>LRR [13]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Context [28]</td>
<td>0.406</td>
<td>0.700</td>
<td>0.423</td>
<td>0.784</td>
</tr>
<tr>
<td>PSPNet [38]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RefineNet-Res50 [27]</td>
<td>0.438</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RefineNet-Res101 [27]</td>
<td>0.447</td>
<td>-</td>
<td>0.457</td>
<td>0.804</td>
</tr>
<tr>
<td>RefineNet-Res152 [27]</td>
<td>0.465</td>
<td>0.736</td>
<td>0.459</td>
<td>0.806</td>
</tr>
</tbody>
</table>
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
Qualitative Results -- Cityscapes

yellow --> closer --> larger pooling size

S. Kong, C. Fowlkes, Recurrent Scene Parsing with Perspective Understanding in the Loop, CVPR, 2018
Qualitative Results -- Stanford-2D-3D (panoramas)

S. Kong, C. Fowlkes, Recurrent Scene Parsing with Perspective Understanding in the Loop, CVPR, 2018
Qualitative Results -- Stanford-2D-3D (panoramas)

Holes are filled!

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

Cityscapes

Baseline: 0.738
MultiPool
- Tied weights: 0.748
- Untied weights: 0.751

Attention: 0.754
Depth-gating: 0.754

Stanford-2D-3D
- GT-depth: 0.753
- Pred-depth: 0.759
Attentional maps prevent the model from pooling across different segments.
Attentional maps prevent the model from pooling across different segments.

Some scales are rarely used.
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

<table>
<thead>
<tr>
<th></th>
<th>baseline</th>
<th>res6</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoU</td>
<td>0.4205</td>
<td>0.4599</td>
</tr>
</tbody>
</table>

(b) depth-aware gating module using predicted depth map
Which layer to insert this attentional gating module?

res1  res2  res3  res4  res5  res6
Which layer to insert this attentional gating module?

<table>
<thead>
<tr>
<th></th>
<th>baseline</th>
<th>res6</th>
<th>res5</th>
<th>res4</th>
<th>res3</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoU</td>
<td>0.4205</td>
<td>0.4599</td>
<td><strong>0.4652</strong></td>
<td>0.4567</td>
<td>0.4413</td>
</tr>
</tbody>
</table>

S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018
Which layer to insert this attentional gating module?

<table>
<thead>
<tr>
<th></th>
<th>baseline</th>
<th>res6</th>
<th>res5</th>
<th>res4</th>
<th>res3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IoU</strong></td>
<td>0.4205</td>
<td>0.4599</td>
<td><strong>0.4652</strong></td>
<td>0.4567</td>
<td>0.4413</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>56</th>
<th>45</th>
<th>345</th>
<th>456</th>
<th>3456</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IoU</strong></td>
<td>0.4644</td>
<td>0.4548</td>
<td>0.4483</td>
<td>0.4497</td>
<td>0.4402</td>
</tr>
</tbody>
</table>

S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018
It achieves the best performance when inserting attentional gating modules at the second last residual block.

<table>
<thead>
<tr>
<th></th>
<th>baseline</th>
<th>res5</th>
</tr>
</thead>
<tbody>
<tr>
<td>IoU</td>
<td>0.4205</td>
<td>0.4652</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>NYU-depth-v2 [35]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IoU</td>
</tr>
<tr>
<td>baseline</td>
<td>0.406</td>
</tr>
<tr>
<td>w/ gt-depth</td>
<td>0.413</td>
</tr>
<tr>
<td>w/ pred-depth</td>
<td>0.418</td>
</tr>
<tr>
<td>loop1 w/o depth</td>
<td>0.419</td>
</tr>
<tr>
<td>loop1 w/ gt-depth</td>
<td>0.425</td>
</tr>
<tr>
<td>loop1 w/ pred-depth</td>
<td>0.427</td>
</tr>
<tr>
<td>loop2</td>
<td>0.431</td>
</tr>
<tr>
<td>loop2 (test-aug)</td>
<td>0.445</td>
</tr>
<tr>
<td>DeepLab [6]</td>
<td>-</td>
</tr>
<tr>
<td>LRR [13]</td>
<td>-</td>
</tr>
<tr>
<td>Context [28]</td>
<td>0.406</td>
</tr>
<tr>
<td>PSPNet [38]</td>
<td>-</td>
</tr>
<tr>
<td>RefineNet-Res50 [27]</td>
<td>0.438</td>
</tr>
<tr>
<td>RefineNet-Res101 [27]</td>
<td>0.447</td>
</tr>
<tr>
<td>RefineNet-Res152 [27]</td>
<td>0.465</td>
</tr>
</tbody>
</table>
Qualitative Results -- res6
Qualitative Results -- res5

Attention to Scale Again
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\}
Qualitative Results -- $\text{res}\{5,6\}$
Qualitative Results -- \( \text{res}\{5,6\} \)
Can we choose the region to process at specific scale, in stead of computing over the whole feature maps?
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.
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
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

Gumbel distribution if \( m \equiv -\log(-\log(u)) \)

where \( u \sim \mathcal{U}[0, 1] \)

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

\[
P(g = k) \propto a_k
\]
Pixel-wise Attentional Gating (PAG)

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

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

where \( u \sim U[0, 1] \)

Let \( g \) be a discrete random variable with probabilities
\[
P(g = k) \propto a_k
\]

let \( \{m_k\}_{k=1,...,K} \) be a sequence of
i.i.d. Gumbel random variables
\[
g = \arg\max_{k=1,...,K} (\log a_k + m_k)
\]
using the Gumbel-Max trick for discrete (binary) masks

\[
g = \arg\max_{k=1,\ldots,K} (\log \alpha_k + m_k)
\]

\[
g = \text{softmax}(\frac{(\log(\alpha + m))/\tau)}{\tau})
\]

\[
\alpha = [\alpha_1, \ldots, \alpha_K]
\]

\[
m = [m_1, \ldots, m_K]
\]

\(\tau\) is the “temperature” parameter.
Pixel-wise Attentional Gating (PAG)

Multiplicative gating as weighted average

Attentional Gating to select

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

attention to pooling size at each pixel
Pixel-wise Attentional Gating (PAG)

Perforated convolution in low-level implementation

Tensor $U$, data matrix $M$, kernel $K$, tensor $V$
pooling using a set of $3 \times 3$-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.
### Pixel-wise Attentional Gating (PAG)

Semantic segmentation

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IoU</td>
<td>pixel acc.</td>
<td>IoU</td>
</tr>
<tr>
<td>baseline</td>
<td>42.1</td>
<td>71.1</td>
<td>79.5</td>
</tr>
<tr>
<td>MP@Res5 (w-Avg.)</td>
<td>46.3</td>
<td>73.4</td>
<td>83.7</td>
</tr>
<tr>
<td>MP@Res5 (PAG)</td>
<td>46.5</td>
<td>73.5</td>
<td>83.7</td>
</tr>
</tbody>
</table>
## Pixel-wise Attentional Gating (PAG)

Monocular depth estimation

![Image](image_url)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1.25</td>
<td>1.25²</td>
<td>1.25³</td>
</tr>
<tr>
<td>baseline</td>
<td>71.1</td>
<td>93.2</td>
<td>98.5</td>
</tr>
<tr>
<td>MP@Res5 (w-Avg.)</td>
<td>74.5</td>
<td>94.4</td>
<td>98.8</td>
</tr>
<tr>
<td>MP@Res5 (PAG)</td>
<td>75.1</td>
<td>94.4</td>
<td>98.8</td>
</tr>
</tbody>
</table>
### Pixel-wise Attentional Gating (PAG)

#### surface normal estimation

![Image of a room showing a desk with computers and a wall with a fire alarm]

**Curricular Data**

<table>
<thead>
<tr>
<th>methods/metrics</th>
<th>NYUv2 [45]</th>
<th></th>
<th></th>
<th></th>
<th>Stanford-2D-3D [47]</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ang. err. ↘</td>
<td>11.25°</td>
<td>22.50°</td>
<td>30.00°</td>
<td>ang. err. ↘</td>
<td>11.25°</td>
<td>22.50°</td>
<td>30.00°</td>
</tr>
<tr>
<td>baseline</td>
<td>22.3</td>
<td>34.4</td>
<td>62.5</td>
<td>74.4</td>
<td>19.0</td>
<td>51.5</td>
<td>68.6</td>
<td>76.3</td>
</tr>
<tr>
<td>MP@Res5 (w-Avg.)</td>
<td>21.9</td>
<td>35.9</td>
<td>63.8</td>
<td>75.3</td>
<td>16.5</td>
<td>58.2</td>
<td>74.2</td>
<td>80.4</td>
</tr>
<tr>
<td>MP@Res5 (PAG)</td>
<td>21.7</td>
<td>36.1</td>
<td>64.2</td>
<td>75.5</td>
<td>16.5</td>
<td>58.3</td>
<td>74.2</td>
<td>80.4</td>
</tr>
</tbody>
</table>

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
More qualitatively results on NYU-depth-v2

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

S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018
More qualitatively results on Cityscapes

S. Kong, C. Fowlkes, Pixel-wise Attentional Gating for Parsimonious Pixel Labeling, 2018
PAG achieves better performance while maintaining the computation.
PAG achieves better performance while maintaining the computation.

It also offers parsimonious inference under limited computation budget.
<table>
<thead>
<tr>
<th>1.</th>
<th>Background</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.</td>
<td>Attention to Perspective: Depth-aware Pooling Module</td>
</tr>
<tr>
<td>3.</td>
<td>Recurrent Refining with Perspective Understanding in the Loop</td>
</tr>
<tr>
<td>4.</td>
<td>Attention to Perspective Again</td>
</tr>
<tr>
<td>5.</td>
<td>Pixel-wise Attentional Gating (PAG)</td>
</tr>
<tr>
<td>6.</td>
<td><strong>Pixel-Level Dynamic Routing</strong></td>
</tr>
<tr>
<td>7.</td>
<td>Conclusion</td>
</tr>
</tbody>
</table>
Parsimonious inference as dynamic computation
Parsimonious inference as dynamic computation

[1] BlockDrop: Dynamic Inference Paths in Residual Networks
More generally, can we allocate dynamic computation time to each pixel of each image instance?
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

\[
X = F^1(I) \\
Y = F^2(X) \\
Z = F^3(Y) \\
O = I + Z
\]

(b) Residual Block with plug-in PAG

\[
X = F^1(I), \quad G = G(I) \\
Y = F^2_G(X) \\
Z = F^3_G(\tilde{G} \odot X + G \odot Y) \\
O = I + Z
\]
sparse binary masks for perforated convolution

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

we compute the empirical sparsity

$$g = \frac{1}{H \times W} \sum_{h,w}^{H,W} G_{h,w}$$

Using KL-divergence term for sparse masks.

$$KL(\rho \| g) = \rho \log \left( \frac{\rho}{g} \right) + (1 - \rho) \log \left( \frac{1 - \rho}{1 - g} \right)$$

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

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.

<table>
<thead>
<tr>
<th>hyper param.</th>
<th>FLOPs</th>
<th>consumption</th>
<th>truncated</th>
<th>layer-skipping</th>
<th>MP@Res5 (PAG)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>IoU</td>
<td>IoU</td>
<td>IoU</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>acc.</td>
<td>acc.</td>
<td>acc.</td>
</tr>
<tr>
<td>ρ = 0.5</td>
<td>6.29</td>
<td>67.69</td>
<td>36.30</td>
<td>37.78</td>
<td>40.89</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>67.36</td>
<td>67.31</td>
<td>69.44</td>
</tr>
<tr>
<td>ρ = 0.7</td>
<td>8.27</td>
<td>86.20</td>
<td>37.69</td>
<td>39.84</td>
<td>43.61</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>67.44</td>
<td>69.00</td>
<td>71.41</td>
</tr>
<tr>
<td>ρ = 0.9</td>
<td>8.95</td>
<td>93.36</td>
<td>40.29</td>
<td>41.27</td>
<td>45.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>69.66</td>
<td>70.01</td>
<td>72.93</td>
</tr>
<tr>
<td>ρ = 1.0</td>
<td>9.63</td>
<td>100.00</td>
<td>—</td>
<td>—</td>
<td>46.52</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>—</td>
<td>—</td>
<td>73.50</td>
</tr>
</tbody>
</table>

\[
l = l_{task} + \lambda \sum_{l=1}^{L} KL(\rho \| g_l)
\]
Boundary detection on BSDS500

\[ \ell = \ell_{task} + \lambda \sum_{l=1}^{L} KL(\rho||g_l) \]
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
Boundary detection on BSDS500 dataset

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

NYU-depth-v2 dataset

- Segmentation
- Depth
- Normal

Annotation

Prediction

Ponder map @Res2

Ponder map @Res3

Ponder map @Res4

Ponder map @Res5

Overall ponder map

MultiPool
Dynamic Computation

Stanford-2D-3D dataset
## Dynamic Computation

### Cityscapes dataset

<table>
<thead>
<tr>
<th>Annotation</th>
<th>Segmentation</th>
<th>Disparity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ponder map @Res2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ponder map @Res3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ponder map @Res4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ponder map @Res5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall ponder map</td>
<td></td>
<td></td>
</tr>
<tr>
<td>multiPool</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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
1. Scene parsing means more than semantic segmentation, geometry and inter-object relation

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

Q&A

Shu Kong  Charless Fowlkes