Context Encoding for Semantic Segmentation

CVPR 2018, Salt Lake City

Hang Zhang\textsuperscript{1,2}, Kristin Dana\textsuperscript{1}, Jianping Shi\textsuperscript{3}, Zhongyue Zhang\textsuperscript{2}, Xiaogang Wang\textsuperscript{4}, Ambrish Tyagi\textsuperscript{2}, and Amit Agrawal\textsuperscript{2}

\textsuperscript{1}Rutgers University, \textsuperscript{2}Amzon Inc, \textsuperscript{3}Sensetime, \textsuperscript{4}CUHK
Semantic Segmentation

- Per-pixel predictions of object categories
- A comprehensive scene description (object category, location and shape)

Examples from ADE20K Dataset.
Fully Convolutional Network \(^1\) (FCN)

- Meta algorithm for Semantic Segmentation
- Pre-trained CNN + Decoder
- Translation equivariant

Figure credit: Long et al.

Difficulties in Predicting Categories and Shapes

- Work refining shapes/boundaries:
  - Dilated/Atrous Convolution \(^{[2,3]}\)
  - CRF Post-processing \(^{[4]}\)
  - Adding Lateral/Skip Connections \(^{[5]}\)
  - Enlarging Spatial Resolution \(^{[6]}\)
- Difficult to identifying categories

---


\(^{5}\)Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation.”

Challenges in Understanding Context

FCN results on ADE20K Dataset. (ResNet 50, stride 8)
Increasing Receptive Field?

Using pyramid representations

• PSPNet [7]
  Spatial Pyramid Pooling

• DeepLab-v3 [8]
  large rate Dilated/Atrous convolutions

“Is capturing contextual information the same as increasing the receptive-field size?”

Labeling an Image

Consider labeling a new image for ADE20K dataset with **150 categories**.
Design a “Labeling Tool” for CNN

- Scene Context
- Narrowing the list of probable categories

Examples from ADE20K Dataset.
Capturing Contextual Info in Computer Vision


Code available on GitHub
Context Encoding

- Encoding Layer \[^9\]
  - Considers $X \in \mathbb{R}^{C \times H \times W}$ as a set of $C$-dimensional features $X = \{x_1, \ldots, x_N\}$, where $N = H \times W$
  - Leans a codebook $D = \{d_1, \ldots, d_K\}$, smoothing factors $S = \{s_1, \ldots, s_K\}$
  - Outputs the residual encoder $e_k = \sum_{i=1}^{N} e_{ik}$:
    \[
    e_{ik} = \frac{\exp(-s_k \|r_{ik}\|^2)}{\sum_{j=1}^{K} \exp(-s_j \|r_{ij}\|^2)} r_{ik}
    \]
    Where the residuals are given by $r_{ik} = x_i - d_k$.

Context Encoding Network (EncNet)

Notation: FC fully connected layer, Conv convolutional layer, Encode Encoding Layer, $\otimes$ channel-wise multiplication

Network Training of EncNet

- ResNet with Dilation Strategy (stride 8)
- Synchronize Cross-GPU Batch Normalization\(^{10}\) (SyncBN)

Ablation Study of EncNet on PASCAL Context

<table>
<thead>
<tr>
<th>Method</th>
<th>BaseNet</th>
<th>Encoding</th>
<th>SE-loss</th>
<th>MS</th>
<th>pixAcc%</th>
<th>mIoU%</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCN</td>
<td>Res50</td>
<td></td>
<td></td>
<td></td>
<td>73.4</td>
<td>41.0</td>
</tr>
<tr>
<td>EncNet</td>
<td>Res50</td>
<td>✓</td>
<td></td>
<td></td>
<td>78.1</td>
<td>47.6</td>
</tr>
<tr>
<td>EncNet</td>
<td>Res50</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>79.4</td>
<td>49.2</td>
</tr>
<tr>
<td>EncNet</td>
<td>Res101</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>80.4</td>
<td>51.7</td>
</tr>
<tr>
<td>EncNet</td>
<td>Res101</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>81.2</td>
<td>52.6</td>
</tr>
</tbody>
</table>

Semantic segmentation results on PASCAL-Context dataset. (mIoU on 59 classes w/o background)

mIoU and pixAcc as a function of SE-loss weight $\alpha$. 
EncNet Results on PASCAL Context

Segmentation results on PASCAL-Context dataset. (mIoU on 60 classes w/ background)
EncNet Results on PASCAL VOC 2012

<table>
<thead>
<tr>
<th>Method</th>
<th>aero</th>
<th>bike</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCN [37]</td>
<td>76.8</td>
<td>34.2</td>
<td>68.9</td>
<td>49.4</td>
<td>60.3</td>
<td>62.2</td>
</tr>
<tr>
<td>DeepLabv2 [4]</td>
<td>84.4</td>
<td>54.5</td>
<td>81.5</td>
<td>63.6</td>
<td>65.9</td>
<td>71.6</td>
</tr>
<tr>
<td>CRF-RNN [60]</td>
<td>87.5</td>
<td>39.0</td>
<td>79.7</td>
<td>64.2</td>
<td>68.3</td>
<td>72.0</td>
</tr>
<tr>
<td>DeconvNet [41]</td>
<td>89.9</td>
<td>39.3</td>
<td>79.7</td>
<td>63.9</td>
<td>68.2</td>
<td>72.5</td>
</tr>
<tr>
<td>GCRF [49]</td>
<td>85.2</td>
<td>43.9</td>
<td>83.3</td>
<td>65.2</td>
<td>68.3</td>
<td>73.2</td>
</tr>
<tr>
<td>DPN [36]</td>
<td>87.7</td>
<td>59.4</td>
<td>78.4</td>
<td>64.9</td>
<td>70.3</td>
<td>74.1</td>
</tr>
<tr>
<td>Piecewise [32]</td>
<td>90.6</td>
<td>37.6</td>
<td>80.0</td>
<td>67.8</td>
<td>74.4</td>
<td>75.3</td>
</tr>
<tr>
<td>ResNet38 [52]</td>
<td>94.4</td>
<td>72.9</td>
<td>94.9</td>
<td>68.8</td>
<td>78.4</td>
<td>82.5</td>
</tr>
<tr>
<td>PSPNet [59]</td>
<td>91.8</td>
<td>71.9</td>
<td>94.7</td>
<td>71.2</td>
<td>75.8</td>
<td>82.6</td>
</tr>
<tr>
<td>EncNet (ours)</td>
<td>94.1</td>
<td>69.2</td>
<td><strong>96.3</strong></td>
<td><strong>76.7</strong></td>
<td><strong>86.2</strong></td>
<td><strong>82.9</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>aero</th>
<th>bike</th>
<th>bird</th>
<th>boat</th>
<th>bottle</th>
<th>mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>CRF-RNN [60]</td>
<td>90.4</td>
<td>55.3</td>
<td>88.7</td>
<td>68.4</td>
<td>69.8</td>
<td>74.7</td>
</tr>
<tr>
<td>Dilation8 [54]</td>
<td>91.7</td>
<td>39.6</td>
<td>87.8</td>
<td>63.1</td>
<td>71.8</td>
<td>75.3</td>
</tr>
<tr>
<td>DPN [36]</td>
<td>89.0</td>
<td>61.6</td>
<td>87.7</td>
<td>66.8</td>
<td>74.7</td>
<td>77.5</td>
</tr>
<tr>
<td>Piecewise [32]</td>
<td>94.1</td>
<td>40.7</td>
<td>84.1</td>
<td>67.8</td>
<td>75.9</td>
<td>78.0</td>
</tr>
<tr>
<td>DeepLabv2 [5]</td>
<td>92.6</td>
<td>60.4</td>
<td>91.6</td>
<td>63.4</td>
<td>76.3</td>
<td>79.7</td>
</tr>
<tr>
<td>RefineNet [31]</td>
<td>95.0</td>
<td>73.2</td>
<td>93.5</td>
<td>78.1</td>
<td>84.8</td>
<td>84.2</td>
</tr>
<tr>
<td>ResNet38 [52]</td>
<td>96.2</td>
<td>75.2</td>
<td><strong>95.4</strong></td>
<td>74.4</td>
<td>81.7</td>
<td>84.9</td>
</tr>
<tr>
<td>PSPNet [59]</td>
<td>95.8</td>
<td>72.7</td>
<td>95.0</td>
<td>78.9</td>
<td>84.4</td>
<td>85.4</td>
</tr>
<tr>
<td>DeepLabv3 [6]</td>
<td>96.4</td>
<td>76.6</td>
<td>92.7</td>
<td>77.8</td>
<td><strong>87.6</strong></td>
<td><strong>85.9</strong></td>
</tr>
<tr>
<td>EncNet (ours)</td>
<td>95.3</td>
<td><strong>76.9</strong></td>
<td>94.2</td>
<td><strong>80.2</strong></td>
<td>85.2</td>
<td><strong>85.9</strong></td>
</tr>
</tbody>
</table>

Results on PASCAL VOC 2012, showing per-class IoU on first 5 categories.

Results on PASCAL VOC 2012 with COCO pre-training, showing per-class IoU on first 5 categories.

EncNet Results on ADE20K

<table>
<thead>
<tr>
<th>Method</th>
<th>BaseNet</th>
<th>pixAcc %</th>
<th>mIoU %</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCN [36]</td>
<td></td>
<td>71.32</td>
<td>29.39</td>
</tr>
<tr>
<td>SegNet [3]</td>
<td></td>
<td>71.00</td>
<td>21.64</td>
</tr>
<tr>
<td>DilatedNet [52]</td>
<td></td>
<td>73.55</td>
<td>32.31</td>
</tr>
<tr>
<td>CascadeNet [59]</td>
<td></td>
<td>74.52</td>
<td>34.90</td>
</tr>
<tr>
<td>RefineNet [31]</td>
<td>Res152</td>
<td>-</td>
<td>40.7</td>
</tr>
<tr>
<td>PSPNet [57]</td>
<td>Res101</td>
<td>81.39</td>
<td>43.29</td>
</tr>
<tr>
<td>PSPNet [57]</td>
<td>Res269</td>
<td>81.69</td>
<td>44.94</td>
</tr>
<tr>
<td>FCN (baseline)</td>
<td>Res50</td>
<td>74.57</td>
<td>34.38</td>
</tr>
<tr>
<td>EncNet (ours)</td>
<td>Res50</td>
<td>79.73</td>
<td>41.11</td>
</tr>
<tr>
<td>EncNet (ours)</td>
<td>Res101</td>
<td>81.69</td>
<td>44.65</td>
</tr>
</tbody>
</table>

Results on ADE20K test set, ranks in COCO-Place challenge 2017. Our single model surpass the winning entry of the COCO-Place challenge and PSPNet-269 (1st place in 2016).


[Image of a table and a chart showing the results]
Visual Examples of EncNet in ADE20K

(a) Image  (b) Ground Truth  (c) FCN  (d) EncNet (ours)  (e) Legend
Conclusion

• Context Encoding Module with EncNet
  • straightforward, light-weight
  • compatible with FCN based approaches

• Superior performance on gold-standard benchmarks.

• The complete systems are publicly available (including SyncBN)
  • Source training/evaluation code and pretrained models
    https://github.com/zhanghang1989/PyTorch-Encoding

• Poster #A5

The authors would like to thank Sean Liu from Amazon Lab 126, Sheng Zha and Mu Li from Amazon AI for helpful discussions and comments. We thank Amazon Web Service (AWS) for providing free EC2 access.
More EncNet Examples on ADE20K Dataset
More EncNet Examples on ADE20K Dataset
More EncNet Examples on ADE20K Dataset
More EncNet Examples on ADE20K Dataset
Conclusion

• Context Encoding Module with EncNet
  • straightforward, light-weight
  • compatible with FCN based approaches

• Superior performance on gold-standard benchmarks.

• The complete systems are publicly available (including SyncBN)
  • Source training/evaluation code and pretrained models
    https://github.com/zhanghang1989/PyTorch-Encoding

• Poster #A5

The authors would like to thank Sean Liu from Amazon Lab 126, Sheng Zha and Mu Li from Amazon AI for helpful discussions and comments. We thank Amazon Web Service (AWS) for providing free EC2 access.
Prior Work in Featuremap Attention

• Spatial Attention: Spatial Transformer Network
• Channel-wise manipulation:
  • AdaIN or MSG-Net in style transfer
  • SE-Net
• Relations and Differences with SE-Net:
  • Semantic Encoding, an explicit representations for global context
  • EncNet directly highlight the class-dependent feature.
EncNet Experiments on CIFAR-10

<table>
<thead>
<tr>
<th>Method</th>
<th>Depth</th>
<th>Params</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet (pre-act) [19]</td>
<td>1001</td>
<td>10.2M</td>
<td>4.62</td>
</tr>
<tr>
<td>Wide ResNet 28×10 [56]</td>
<td>28</td>
<td>36.5M</td>
<td>3.89</td>
</tr>
<tr>
<td>ResNeXt-29 16×64d [53]</td>
<td>29</td>
<td>68.1M</td>
<td>3.58</td>
</tr>
<tr>
<td>DenseNet-BC (k=40) [21]</td>
<td>190</td>
<td>25.6M</td>
<td>3.46</td>
</tr>
<tr>
<td>ResNet 64d (baseline)</td>
<td>14</td>
<td>2.7M</td>
<td>4.93</td>
</tr>
<tr>
<td>Se-ResNet 64d (baseline)</td>
<td>14</td>
<td>2.8M</td>
<td>4.65</td>
</tr>
<tr>
<td>EncNet 16k64d (ours)</td>
<td>14</td>
<td>3.5M</td>
<td>3.96</td>
</tr>
<tr>
<td>EncNet 32k128d (ours)</td>
<td>14</td>
<td>16.8M</td>
<td>3.45</td>
</tr>
</tbody>
</table>

Comparison of model depth, number of parameters, test errors (%) on CIFAR-10.

Train and validation curves of EncNet-32k64d and the baseline Se-ResNet-64d on CIFAR-10 dataset.
Context Encoding

• Encoding Layer \[^9\]
  • Outputs the residual encoder as encoded semantics \( e = \sum_{k=1}^{K} \phi(e_k) \)

• Featuremap Attention
  • \( FC \) on encoded semantics, outputs scaling factors \( \gamma = \delta(We) \), where \( W \) is the layer weight and \( \delta \) is sigmoid function.
  • Channel-wise multiplication \( Y = X \otimes \gamma \)

Standard BN and Data Parallelism

Batch Normalization \(^5\) in training mode.

\[
\begin{align*}
\frac{d_f}{dx} &\rightarrow \mu & \frac{d_f}{d\mu} &\rightarrow \sigma^2 \\
\frac{d_f}{d\sigma^2} &\rightarrow \sigma & \frac{d_f}{dx} &\rightarrow y
\end{align*}
\]

Standard BN with data parallel implementation.

\[
\begin{align*}
\ell &\rightarrow \mu \rightarrow \sigma \\
\ell' &\rightarrow \mu' \rightarrow \sigma'
\end{align*}
\]

\(x = \{x_1, x_2, x_3, x_4\}\)

\(y = \{y_1, y_2, y_3, y_4\}\)

Cross-GPU Batch Norm ("Sync twice")

\[ \mu = \frac{\sum x_i}{N} \]
\[ \sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{N} + \epsilon} \]

6Peng, Chao, et al. "MegDet: A Large Mini-Batch Object Detector." CVPR2018
7Liu, Shu, et al. "Path Aggregation Network for Instance Segmentation." CVPR2018
Cross-GPU Batch Norm ("Sync once")

\[ \mu = \frac{\sum x_i}{N} \]

\[ \sigma = \sqrt{\frac{\sum (x_i - \mu)^2}{N} + \epsilon} \]

\[ = \sqrt{\frac{\sum x_i^2}{N} - \mu^2 + \epsilon} \]

\[ = \sqrt{\frac{\sum x_i^2}{N} - \left(\frac{\sum x_i}{N}\right)^2 + \epsilon} \]

6Peng, Chao, et al. "MegDet: A Large Mini-Batch Object Detector." CVPR2018

7Liu, Shu, et al. "Path Aggregation Network for Instance Segmentation." CVPR2018
More EncNet Examples on ADE20K Dataset
More EncNet Examples on ADE20K Dataset
Failure Examples of EncNet
Highlights

• Introduce a novel CNN architecture:
  • Context Encoding Network, EncNet ("Ink-Net")

• State-of-the-art performance:
  • 85.9% mIoU on PASCAL VOC 2012, 51.7% mIoU on PASCAL Context
  • On ADE20K dataset, out single model surpass the winning entry of COCO Places challenge 2017 (achieving a final score of 0.5567)

• The complete system are publicly available.
  • Source code and pretrained models