Multiscale methods for neural image processing

Sohil Shah, Pallabi Ghosh, Larry S. Davis and Tom Goldstein
Hao Li, Soham De, Zheng Xu, Hanan Samet
A TALK IN TWO ACTS

**Part I: Stacked U-nets**

The globalization problem for segmentation

Simple ways to solve it

**Part 2: Quantized neural nets**

Putting neural nets on portable devices

Binarized training methods, and why they work
PROBLEM SETTING

PASCAL Visual Object Classes (VOC)

Cityscapes
SIMPLE APPROACHES

✓ Designed for classification task
✓ Parameter heavy
✓ Low resolution output
✓ Minimal exchange of context
WHAT MAKES SEGMENTATION HARD?

Field of view problem

High-resolution outputs

Fine scale localization required

Small data
TV IN 2D

\[(\nabla x)_{ij} = (x_{i+1,j} - x_{i,j}, x_{i,j+1} - x_{i,j})\]

Anisotropic \( |(\nabla x)_{ij}| = |x_{i+1,j} - x_{i,j}| + |x_{i,j+1} - x_{i,j}| \)

Isotropic \( \| (\nabla x)_{ij} \| = \sqrt{(x_{i+1,j} - x_{ij})^2 + (x_{i,j+1} - x_{ij})^2} \)
CONVEX SEGMENTATION: FINE SCALE

• Solve speed limited by the **CFL**

\[ \min \mu |\nabla u| + \|u - f\|^2 \]
CONVEX SEGMENTATION: COARSE SCALE

- TV changes the CFL condition

$$\min \mu |\nabla u| + ||u - f||^2$$
SCALE AFFECTS SPEED!

Segmentation using TV

15 iterations  280 iterations  4500 iterations

Solution: use multi-grid!
NEURAL NETS HAVE THE SAME PROBLEMS

far away points talk to each other
CLASSIFICATION NET

Globalize via **pooling**

output contains **global** info
CLASSIFICATION NET

Globalize via convolution

CFL condition becomes field of view problem

High-res output
FIELD OF VIEW BREAKDOWN

Example from: **Zhao et al. “Pyramid Scene Parsing Network”**
GLOBALIZATION METHODS

THINGS GET COMPLICATED…
APPROACHES TO GLOBALIZATION

**Dilated/de-conv modules**
Chen et al. “Semantic image segmentation…” 2014
Yu & Koltun, “Multi-scale context aggregation” 2015

**Multi-scale feature ensembling**
Long et al., “Fully Convolution Semantic Segmentation” 2015
Chen et al. “Attention to scale…” 2016
Xia et al. “Zoom better to see clearer” 2016
Hariharan et al. “Hypercolumns for object segmentation” 2015

**Hand-crafted features**
Lazebnik, Schmid, Ponce, “Beyond bags of features” 2006
Lucchi et al. “Are spatial constraints necessary for segmentation” 2011

Chen et al. “DeepLab”  
Hengshuang et al. “PSPNet”
SEGMENTATION NET

standard net

de-conv units

High resolution
Low localization

LC Chen et al. “DeepLab”
LC Chen et al. “DeepLab”

CRF or graph cuts

segmentation
PSPNET

High memory requirements

Hengshuang Zhao et al. “Pyramid scene parsing network”
BACK TO BASICS…

A “NO-FRILLS” APPROACH
MULTI-GRID SOLVERS

Level 1

Level 2

Level 3
THE U-NET

A V-cycle for neural nets

Ronneberger et al, 2015
MEDICAL IMAGING

resolution is **high**
class complexity is **low**

Ronneberger et al, 2015
NEURAL NET BUILDING BLOCKS

- **Input image**
- Conv
- Relu
- Conv
- Relu
“RESNET” BLOCK
MODIFIED U-NET BLOCK
BIG IDEA: STACKED U-NETS

Combine information globalization of U-nets with power of ResNets

No frills!
TRAINING ON REAL DATASETS
LIMITED TRAINING DATA

ImageNet  1,000,000

MS-COCO  90,000

PASCAL VOC  1464

Solution: pre-train on big datasets, fine tune on small
STANDARD APPROACH

Phase I: Train a classification net
STANDARD APPROACH

Phase 2: Add fancy stuff

standard net

---

fancy stuff

Roughly doubles parameters!
Dilated SUNets

Remove pooling layers
Increase dilation factor of convolutions

Exact same parameters!
RESULTS
METRICS

**Classification**
Top-1 accuracy
Top-5 accuracy

“Cat”

**Segmentation**
Intersection over union
### RESULTS

#### Classification Performance

<table>
<thead>
<tr>
<th>Model</th>
<th>Top-1</th>
<th>Top-5</th>
<th>Depth</th>
<th>Params</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-18 †</td>
<td>30.24</td>
<td>10.92</td>
<td>18</td>
<td>11.7M</td>
</tr>
<tr>
<td>ResNet-50 †</td>
<td>23.85</td>
<td>7.13</td>
<td>50</td>
<td>25.6M</td>
</tr>
<tr>
<td>ResNet-101 †</td>
<td>22.63</td>
<td>6.44</td>
<td>101</td>
<td>44.5M</td>
</tr>
<tr>
<td>DenseNet-201 †</td>
<td>22.80</td>
<td>6.43</td>
<td>201</td>
<td>20M</td>
</tr>
<tr>
<td>DenseNet-161 †</td>
<td>22.35</td>
<td>6.20</td>
<td>161</td>
<td>28.5M</td>
</tr>
<tr>
<td>SUNet-64</td>
<td>29.28</td>
<td>10.21</td>
<td>111</td>
<td>6.9M</td>
</tr>
<tr>
<td>SUNet-128</td>
<td>23.64</td>
<td>7.56</td>
<td>111</td>
<td>24.6M</td>
</tr>
<tr>
<td>SUNet-7-128</td>
<td>22.47</td>
<td>6.85</td>
<td>171</td>
<td>37.7M</td>
</tr>
</tbody>
</table>

#### Segmentation Performance

<table>
<thead>
<tr>
<th>Model</th>
<th>mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet-101 [218]</td>
<td>68.39</td>
</tr>
<tr>
<td>SUNet-64</td>
<td>72.85</td>
</tr>
<tr>
<td>SUNet-128</td>
<td>77.16</td>
</tr>
<tr>
<td>SUNet-7-128</td>
<td>78.95</td>
</tr>
</tbody>
</table>

✓ 4.5% mIoU ↑ with 7x fewer parameters  
✓ 10.5% mIoU ↑ w/o sacrificing classification performance

### SEGMENTATION RESULTS

#### Table 7.6: Performance comparison on PASCAL VOC 2012 validation set. All networks were pretrained with MS-COCO.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Validation mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resnet+ASPP</td>
<td>82.70</td>
</tr>
<tr>
<td>Xception+ASPP+Decoder</td>
<td><strong>83.34</strong></td>
</tr>
<tr>
<td>SUNet-7128</td>
<td><strong>83.27</strong></td>
</tr>
<tr>
<td>PSPNet [217]</td>
<td><strong>85.4</strong></td>
</tr>
<tr>
<td>SUNet-7-128</td>
<td>84.3</td>
</tr>
</tbody>
</table>

#### Table 7.7: Performance comparison on PASCAL VOC 2012 test set. For fair comparison, only the methods pre-trained using MS-COCO are displayed.

<table>
<thead>
<tr>
<th>Methods</th>
<th>mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Piecewise (VGG16) [224]</td>
<td>78.0</td>
</tr>
<tr>
<td>LRR+CRF [209]</td>
<td>77.3</td>
</tr>
<tr>
<td>DeepLabv2+CRF [210]</td>
<td>79.7</td>
</tr>
<tr>
<td>Large-Kernel+CRF [220]</td>
<td>82.2</td>
</tr>
<tr>
<td>Deep Layer Cascade* [254]</td>
<td>82.7</td>
</tr>
<tr>
<td>Understanding Conv [221]</td>
<td>83.1</td>
</tr>
<tr>
<td>RefineNet [205]</td>
<td>82.4</td>
</tr>
<tr>
<td>RefineNet-ResNet152 [205]</td>
<td>83.4</td>
</tr>
<tr>
<td>PSPNet [217]</td>
<td><strong>85.4</strong></td>
</tr>
<tr>
<td>SUNet-7-128</td>
<td>84.3</td>
</tr>
</tbody>
</table>

#### Table 7.8: Performance comparison on Cityscapes test set. All methods were trained only using the “fine” set. All nets utilize ResNet-101 as a base network, except if specified or marked with ⇤.

<table>
<thead>
<tr>
<th>Methods</th>
<th>mIoU</th>
</tr>
</thead>
<tbody>
<tr>
<td>LRR (VGG16) [209]</td>
<td>69.7</td>
</tr>
<tr>
<td>DeepLabv2+CRF [210]</td>
<td>70.4</td>
</tr>
<tr>
<td>Deep Layer Cascade* [254]</td>
<td>71.1</td>
</tr>
<tr>
<td>Piecewise (VGG16) [224]</td>
<td>71.6</td>
</tr>
<tr>
<td>RefineNet [205]</td>
<td>73.6</td>
</tr>
<tr>
<td>Understanding Conv [221]</td>
<td>77.6</td>
</tr>
<tr>
<td>PSPNet [217]</td>
<td>78.4</td>
</tr>
<tr>
<td>SUNet-7-128</td>
<td>75.3</td>
</tr>
</tbody>
</table>

#### 7.6.6 Results on Test set

**PASCAL VOC 2012:**

Before submitting test set output to an evaluation server, the above model was further fine-tuned on the “trainval” set with batch-norm parameters frozen and at 10⇥ smaller initial learning rate. Table 7.7 compares the test set results against other state-of-the-art methods. PSPNet performs slightly better than SUNet, but at the cost of 30M more parameters while training at an output stride = 8. Figure 7.3, 7.4 displays some qualitative results on validation and test sets.

**Cityscapes:**

A similar training strategy as in PASCAL is adopted except that the multi-scale inference is performed on additional scales \{1.5, 1.75, 2.0, 2.25, 2.5\}. Only the finer \(\times 2\) inference was used for training on train+val (2x data).

#### 7.6.7 Activation Maps

Figure 7.5 shows the activation map recorded at the end of each level (as indicated in figure 7.2) for an example input image of an “Aeroplane.” As noted earlier, the inclusion of strided convolutions instead of multigrid dilations leads to noisy feature maps (see col 3; rows 4-6). The addition of de-gridding layers serves to produce a coherent prediction map at the output (see col 2; row 6).

#### 7.7 Conclusion

The fundamental structure of conventional bottom-up classification networks limits their efficacy on secondary tasks involving pixel-level localization or classification. To...
SAMPLE RESULTS
SAMPLE RESULTS
SAMPLE RESULTS
SAMPLE RESULTS
FAILURE CASE

input | target | output
FAILURE CASE
SUMMARY OF PART I

CFL condition for convex segmentation  FOV problem for neural nets

Unet blocks globalize information better

We can compete with SOTA using simple architectures

SUNets use fewer parameters and train fast
QUESTIONS?
And now for something completely different...

QUANTIZED NETS
DEEP NETS ARE BIG

Low power devices

SOLUTION: QUANTIZED/LOW-PRECISION NETWORKS

X1 ± 1
X2 ± 1
X3 ± 1

\[ f(\sum_{i=1}^{n} W_i X_i) \]

Advantages

• FAST & hardware friendly: no multiplications
• Low storage costs
• Low power consumption

BinaryConnect [Courbariaux NIPS’15]
BinaryNet [Hubara NIPS’16]
XNOR-Net [Rastegar, ECCV’16]
DoReFA-Net [Zhou, arXiv’16]
DeepCompression [Han, ICLR’16]
……
HOW TO TRAIN QUANTIZED NETS?

minimize \( f(w) \)

Non-quantized: Stochastic Gradient Descent

\[
    w^{k+1} = w^k - \alpha \nabla f(w^k)
\]

**Fully-quantized**: Stochastic rounding [Gupta ICML’15]

\[
    w^{k+1} = w^k - Q[\alpha \nabla f(w^k)]
\]
EXPERIMENT

Train CNNs (VGG-Net, ResNets, Wide-RseNet) with binary weight on CIFAR-10
HOW TO TRAIN QUANTIZED NETS?

minimize $f(w)$

**Fully-quantized**: Stochastic rounding [Gupta ICML'15]

$$w^{k+1} = w^k - Q[\alpha \nabla f(w^k)]$$

**Semi-quantized**: BinaryConnect [Courbariaux NIPS'15]

$$w^{k+1} = w^k - \alpha \nabla f(Q[w^k])$$

Floating point
EXPERIMENT

Train CNNs (VGG-Net, ResNets, Wide-RseNet) with binary weight on CIFAR-10

The SR method cannot beat BC, why?
BINARIZED NETS GO MAINSTREAM

Hubara et al. “Binarized neural nets…”
Rastegari et al. “XNOR Net…”
Lin et al. “Nets with few multiplications…”
Lin, Zhao, Pan. “Towards accurate binary…”
Shayar et al. “Learning discrete weights…”
Park, Ahn, Yoo. “Weighted Entropy quantization…”
Mishra et al. “WRPN: Wide reduced precision nets”
Kim Zhu et al. “Trained ternary quantization…”
Umuroglu et al. “Finn: A framework…”
Kim & Smaragdis. “Bitwise Nets…”

…and tons more
CONVERGENCE UNDER CONVEXITY ASSUMPTIONS
Theorem 2 Assume that $F$ is $\mu$-strongly convex and the learning rates are given by $\alpha_t = \frac{1}{\mu(t+1)}$. Let $G$ bound the gradient magnitude. Then

$$\mathbb{E}[F(\bar{w}^T) - F(w^*)] \leq \frac{(1 + \log(T + 1))G^2(1 + \Delta^{-1})}{2\mu T} + \frac{\sqrt{d} \Delta G}{2}$$
Theorem 2 Assume that $F$ is $\mu$-strongly convex and the learning rates are given by $\alpha_t = \frac{1}{\mu(t+1)}$. Let $G$ bound the gradient magnitude. Then

$$\mathbb{E}[F(\bar{w}^T) - F(w^*)] \leq \frac{(1 + \log(T+1))G^2(1 + \Delta^{-1})}{2\mu T} + \frac{\sqrt{d\Delta G}}{2}$$

Fully quantized noise floor
Theorem 1 Assume that $F$ is $\mu$ -strongly convex and the learning rates are given by $\alpha_t = \frac{1}{\mu(t+1)}$. Let $G$ bound the gradient magnitude. Then

$$\mathbb{E}[F(\bar{w}^T) - F(w^*)] \leq \frac{(1 + \log(T + 1))G^2}{2\mu T} + \frac{DL_2\sqrt{d}\Delta}{2}$$

$L_2$ is a Lipschitz constant for the Hessian

semi-quantized noise floor
But this can’t be the whole story.

What can we say about these methods on NON-CONVEX problems?

The answer has to do with exploration vs exploitation.
FLOATING POINT

Learning rate = 1

Quantized scalar weight $\Delta = 0.5$
FLOATING POINT

Learning rate = 1

Quantized scalar weight $\Delta = 0.5$
FLOATING POINT

Learning rate = 1

Quantized scalar weight \( \Delta = 0.5 \)
FLOATING POINT

Learning rate = 1

Quantized scalar weight $\Delta = 0.5$
FLOATING POINT

Learning rate = 0.1

Histogram
FLOATING POINT

Learning rate = 0.01

Histogram
FLOATING POINT

Learning rate = 0.001

Histogram
FULLY QUANTIZED

Learning rate = 0.1

Histogram
FULLY QUANTIZED
Learning rate = 0.01

Histogram
FULLY QUANTIZED

Learning rate = 0.001

Histogram
WHAT'S WRONG?

Floating Point / Binary Connect

Exploration \rightarrow \text{Shrink Learning rate} \rightarrow \text{Exploitation}

Stochastic Rounding

Exploration \rightarrow \text{Shrink Learning rate} \rightarrow \text{Exploitation}
MARKOV CHAIN INTERPRETATION

“Weight space’’

\[ w_{k+1} = w_k - Q[\alpha \nabla f(w_k)] \]
MARKOV CHAIN INTERPRETATION

Long term dynamics governed by the equilibrium distribution $\pi_{\alpha}$

$w^{k+1} = w^k - Q[\alpha \nabla f(w^k)]$
MARKOV CHAIN INTERPRETATION

$\pi_{\alpha}$

equilibrium distribution
What about stochastic rounding?

Fully discrete stochastic rounding does not concentrate on stationary points.

**Theorem:** If the noise satisfies

- Tails aren’t too fat: $\int_\nu^\infty p_{x,k}(z) \, dz < C/\nu^2$
- Not concentrated at 0
- SGD iterates stay bounded

Then...

- Iterates of fully-quantized SGD don’t concentrate on minimizers.
- But BinaryConnect does!

Assumptions are weak enough for neural nets!
ANNEALING PROPERTIES MAKE A DIFFERENCE

Train CNNs (VGG-Net, ResNets, Wide-ResNet) with binary weight on CIFAR-10/100

concentration/annealing

BinaryConnect

Full-Precision
SUMMARY OF PART II

Quantized nets reduce net size and energy

Training requires specialized rounding methods

We still don’t understand the non-convexities of neural nets!
THANKS!

Stacked U-Nets: A No-Frills Approach to Natural Image Segmentation

Sohil Shah, Pallabi Ghosh, Larry S. Davis and Tom Goldstein

Towards a deeper Understanding of Training Quantized Networks

Hao Li, Soham De, Zheng Xu, Christoph Studer, Hanan Samet, Tom Goldstein