Integrating Structural Information in Deep Convolutional Neural Networks for Low- and High-Level Vision

Iasonas Kokkinos
Center for Visual Computing
CentraleSupelec

Galen Group
INRIA-Saclay

31 March, 2016
Center for Machine Perception, Prague
Deep Learning and Computer Vision

1980’s
pixels → edge → texton → motif → part → object

2000-2010

2010+

Breakthrough: Imagenet 2012

Imagenet top-5 error rates


K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, Arxiv, 2015 [3.6%]
DCNNs and Vision

2012 onwards: all about DCNNs

if [all] you have [is] a hammer, you treat everything like a nail

- Classification & Detection
- Semantic Segmentation
- Boundary Detection
- Feature Descriptors

Today:

2014 onwards: structured prediction and DCNNs

trust is good, but control is better!

This talk: controlling DCNNs for low- and high- level tasks
Convolutional/Fully Connected DCNN layers

AlexNet

VGG network
Fully convolutional neural networks

"FCNNs" (2015) or "Space Displacement Neural Nets" (1998)

Fully convolutional neural networks
Fully convolutional neural networks
Fully convolutional neural networks
Fully convolutional neural networks
Fully convolutional neural networks

Fast (shared convolutions)
Simple (dense)
This talk: controlling DCNNs for low- and high- level tasks

- Classification & Detection
- Semantic Segmentation
- Boundary Detection
- Feature Descriptors

Scale-Invariant classification

\[ F(x) \rightarrow \{ F(x_{s1}), \ldots, F(x_{sK}) \} \]

\[ F'(x) = \frac{1}{K} \sum_{k=1}^{K} F(x_{sk}) \]

This work: \[ F'(x) = \max_{k} F(x_{sk}) \]

MIL: ‘bag’ of features


T. Dietterich et al. Solving the multiple-instance problem with axis-parallel rectangles. Artificial Intelligence, 1997
Position and Scale evaluation in `batch mode`

Dubout, C., Fleuret, F.: Exact acceleration of linear object detectors. ECCV 2012
Explicit Scale/Position Search + MIL Training

\[ G_c = \max_{(x,y)} F_c(x, y) \]

MIL: Explicit position & scale search during both training and testing

(0) Baseline: max-pooled net
- 13.0%
- ~1% gain

(1) epitomic DCNN
- 11.9%
- ~2% gain

(2) epitomic DCNN + search
- 10.0%

Bonus: Vanilla argmax yields 48% localization error in Imagenet
Towards Object Detection

Search over position and scale: done!

Missing: aspect ratio
Procrustes Alignment: The Greeks did it first!

R. Girschick, Donahue, Darrell, Malik, RCNN, CVPR 2014
Explicit search over aspect ratio, scale & position
Explicit search over aspect ratio, scale & position

See also: Region Proposal Networks (RPN) Faster-RCNN, 2016
## Pascal VOC 2007: Best sliding-window detector

<table>
<thead>
<tr>
<th>Method</th>
<th>Percentage</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN-DPM [1]</td>
<td>43.4%</td>
<td>~10 sec/image</td>
</tr>
<tr>
<td>MP-DPM [2]</td>
<td>46.5%</td>
<td>~10 sec/image</td>
</tr>
<tr>
<td>EE-DPM [3]</td>
<td>46.9%</td>
<td>~10 sec/image</td>
</tr>
<tr>
<td>Ours</td>
<td>58.6%</td>
<td>~10 sec/image</td>
</tr>
</tbody>
</table>

- **sliding windows**
  - CNN-DPM [1]: 43.4%
  - MP-DPM [2]: 46.5%
  - EE-DPM [3]: 46.9%
  - Ours: 58.6%

- **region proposals**
  - RCNN [4]: 62.2%

---

This talk: controlling DCNNs for low- and high-level tasks

- Classification & Detection
- Semantic Segmentation
- Boundary Detection
- Feature Descriptors

S. Chandra, I. Kokkinos, Fast, Exact and Multi-Scale Inference for Semantic Image Segmentation with Deep Gaussian CRFs, arXiv:1603.08358
Semantic segmentation task
Repurposing DCNNs for semantic segmentation

- Accelerate CNN evaluation by ‘hard dropout’ & finetuning
  - In VGG: Subsample first FC layer 7x7 → 3x3

- Decrease score map stride (32->8) with ‘atrous’ (w. holes) algorithm

OK classification-wise, rather poor segmentation-wise

- Large CNN receptive field:
  + good accuracy
  - worse performance near boundaries

J. Long, E. Shelhamer, T. Darrell, FCNNs for Semantic Segmentation, CVPR 15
FCNN-DenseCRF: Accurate & Sharp

P. Krähenbühl and V. Koltun, Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials, NIPS 2011

Markov Random Fields in Vision

\[ P(X, Y) = \frac{1}{Z} \prod_i \Phi(Y_i, X_i) \prod_{(i,j) \in C} \Psi(X_i, X_j) \]

\[ P(X|Y) = ? \]
Mean Field Inference for the Ising Model

Variational Inference: \[ q^* = \arg\min_{q \in \mathcal{Q}} KL(q||p) \]

where: \[ KL(q||p) = \sum_x q(x) \log \frac{q(x)}{p(x)} \]

and \( \mathcal{Q} \) simplifies minimization

Naïve mean field: \[ \mathcal{Q} : \{ q : q(x) = \prod_n q_n(x_n) \} \]

Ising model: \[ p(x) = \frac{1}{Z} \exp (-E(x)) \]

\[ E(x) = \sum_n \sum_{m \in \mathcal{N}_n} J_{m,n} |x_m - x_n| \quad x_n \in \{-1, 1\} \]

Mean Field equations: \[ q_n(1) = \tanh \left( \sum_m J_{n,m} q_m(1) \right) \]
Dense CRF: smart choice of pairwise term

\[
\psi_{i,j}(l, l') = \mu(l, l') \sum_{m=1}^{M} w_m k_m(f_i, f_j)
\]

\[
= [l \neq l'] \left[ w_1 \exp \left( -\frac{||p_i - p_j||^2}{2\sigma^2_a} - \frac{||I_i - I_j||^2}{2\sigma^2_b} \right) + w_2 \exp \left( -\frac{||p_i - p_j||^2}{2\sigma^2_\gamma} \right) \right]
\]

Potts model \quad \text{‘Bilateral kernel’} \quad \text{Spatial proximity}

Mean Field Updates:

\[
Q_i(l) = \frac{1}{Z_i} \exp \left\{ -\psi_i(l) - \sum_{l'} \mu(l, l') \sum_{m=1}^{M} w_m \sum_{j \in N_i} k_m(f_i, f_j) Q_j(l') \right\}
\]

Efficient high-dimensional convolutions using the Permutohedral Lattice

Philipp Krähenbühl and Vladlen Koltun, Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials, NIPS 2011
Qualitative Results

FCNN

FCNN-DCRF
Qualitative Results

<table>
<thead>
<tr>
<th>Image</th>
<th>FCNN</th>
<th>FCNN-DCRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bird in Flight</td>
<td><img src="image" alt="FCNN" /></td>
<td><img src="image" alt="FCNN-DCRF" /></td>
</tr>
<tr>
<td>Steam Engine</td>
<td><img src="image" alt="FCNN" /></td>
<td><img src="image" alt="FCNN-DCRF" /></td>
</tr>
<tr>
<td>Chairs</td>
<td><img src="image" alt="FCNN" /></td>
<td><img src="image" alt="FCNN-DCRF" /></td>
</tr>
<tr>
<td>Fishing Boat</td>
<td><img src="image" alt="FCNN" /></td>
<td><img src="image" alt="FCNN-DCRF" /></td>
</tr>
</tbody>
</table>
Qualitative Results

FCNN

FCNN-DCRF
Indicative Results

![Images of results for FCNN and FCNN-DCRF]

- FCNN
- FCNN-DCRF
Comparison to state-of-the-art (Pascal VOC test)

<table>
<thead>
<tr>
<th>Method</th>
<th>mean IOU (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSRA-CFM</td>
<td>61.8</td>
</tr>
<tr>
<td>FCN-8s</td>
<td>62.2</td>
</tr>
<tr>
<td>TTI-Zoomout-16</td>
<td>64.4</td>
</tr>
<tr>
<td>DeepLab-CRF (our)</td>
<td>66.4</td>
</tr>
<tr>
<td>DeepLab-MSc-CRF (our)</td>
<td>67.1</td>
</tr>
</tbody>
</table>

Pre-CNN: Up to 50%

CNN: 60-64%

CNN + CRF: >67%

Pascal Train: 67%

Coco + Pascal: 71%


Semantic Part Segmentation

Fast, Exact, and Multi-Scale Inference for FCNN-CRF

S. Chandra, I. Kokkinos, Fast, Exact and Multi-Scale Inference for Semantic Image Segmentation with Deep Gaussian CRFs, arXiv:1603.08358
Gaussian Random Fields: Random Fields for dummies

\[
\pi(x) = \frac{1}{Z} \exp \left( -x^T \Theta x + \theta^T x \right)
\]

\[
\Theta x^* = \theta
\]

Maximum-A-Posteriori inference =
Minimum Mean-Squared Error inference =
solution of linear system

Gaussian MRF: blurry samples (hard to have outliers)
Gaussian CRF: image-based pairwise terms (e.g. discontinuity-preserving)

Jancsary, Nowozin, Sharp & Rother, Regression Tree Fields, CVPR12
Tappen, Liu, Adelson & Freeman, Learning Gaussian CRFs for low-level vision, CVPR07
Deep Gaussian Conditional Random Field

Pairwise terms

Dog-Background (Vertical)

Dog-Background (Horizontal)

Outputs

Unary terms

Posterior
Deep Gaussian Conditional Random Field vs. DenseCRF

Variables
- Deep Gaussian CRF: continuous
- Dense CRF: discrete

Inference
- Deep Gaussian CRF: exact (linear system)
- Dense CRF: approximate (mean-field)

Learning
- Deep Gaussian CRF: exact (linear system)
- Dense CRF: BackProp on mean-field

Unary terms
- Deep Gaussian CRF: CNN-based
- Dense CRF: CNN-based

Pairwise terms
- Deep Gaussian CRF: CNN-based
- Dense CRF: parametric (Gaussian form)
Linear systems & Gaussian CRFs

\[ Ax = B \]

Gauss-Seidel:
\[ x_i^{(k+1)} \leftarrow \frac{1}{a_{ii}} \left\{ b_i - \sum_{j<i} a_{ij} x_j^{(k+1)} - \sum_{j>i} a_{ij} x_j^{(k)} \right\} \]

Jacobi:
\[ x_i^{(k+1)} \leftarrow \frac{1}{a_{ii}} \left\{ b_i - \sum_{j \neq i} a_{ij} x_j^{(k)} \right\} \]

\[ \Theta x^* = \theta \]

Parallel Mean-Field

Sequential Mean-Field

Conjugate gradients: 2x faster!
Naïve Multi-Resolution Semantic Segmentation


I. Kokkinos, Pushing the Boundaries of Boundary Detection using Deep Learning, ICLR 2016
Learn to enforce coupling of different results
Consistently better results than decoupled learning!
Improvements/Complementarity with DenseCRF
## Quantitative Results

<table>
<thead>
<tr>
<th>Method</th>
<th>IoU</th>
<th>IoU after dense CRF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basenet</td>
<td>72.72</td>
<td>73.78</td>
</tr>
<tr>
<td>QO₄</td>
<td>73.41</td>
<td>75.13</td>
</tr>
<tr>
<td>QO₄&lt;sup&gt;mres&lt;/sup&gt;</td>
<td>73.86</td>
<td>75.46</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>mean IoU (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepLab-CRF (Chen et al., 2014)</td>
<td>66.4</td>
</tr>
<tr>
<td>DeepLab-MSc-CRF (Chen et al., 2014)</td>
<td>67.1</td>
</tr>
<tr>
<td>DeepLab-CRF-7x7 (Chen et al., 2014)</td>
<td>70.3</td>
</tr>
<tr>
<td>DeepLab-CRF-LargeFOV (Chen et al., 2014)</td>
<td>70.3</td>
</tr>
<tr>
<td>DeepLab-MSc-CRF-LargeFOV (Chen et al., 2014)</td>
<td>71.6</td>
</tr>
<tr>
<td>Deeplab-Cross-Joint (Chen et al., 2015a)</td>
<td>73.9</td>
</tr>
<tr>
<td>CRFRNN (Zheng et al., 2015)</td>
<td>74.7</td>
</tr>
<tr>
<td>Adelaide Context (Lin et al., 2016)</td>
<td>77.8</td>
</tr>
<tr>
<td>Deep Parsing Network (Liu et al., 2015)</td>
<td>77.4</td>
</tr>
<tr>
<td>Ours (QO₄&lt;sup&gt;mres&lt;/sup&gt;)</td>
<td>75.5</td>
</tr>
</tbody>
</table>
This talk: controlling DCNNs for low- and high- level tasks

- Classification & Detection
- Semantic Segmentation
- Boundary Detection
- Feature Descriptors

I. Kokkinos, Pushing the Boundaries of Boundary Detection using Deep Learning, ICLR 2016 (earlier title: ‘Surpassing Humans in Boundary Detection’)
Can humans do it?

Segmentation: task-agnostic, ill-posed
Can humans do it?

Segmentation: task-agnostic, ill-posed
Can humans do it?

Segmentation: task-agnostic, ill-posed
30 years of boundary detection

S. Xie and Z. Tu, Holistically-Nested Edge Detection, ICCV 2015
I. Kokkinos, Pushing the boundaries of boundary detection using deep learning, ICLR 2016
This work

Starting point:
Holistically-Nested Edge Detection,
S. Xie and Z. Tu, ICCV 2015

Learning Techniques:
Multiple Instance Learning for Boundary Detection
Graduated Deep Supervised Networks

Network Architecture:
Tied Multi-Scale Networks
Grouping in DCNNs
Holistically-Nested Edge Detection network
HED network

VGG convolutional layers (1-5)

Outputs: $f^m$, $m = 1, \ldots, 5$
HED network

Parameters: $\mathbf{W}^m$

Inputs: $\mathbf{f}^m$

Outputs: $\mathbf{S}^m = \langle \mathbf{W}^m, \mathbf{f}^m \rangle$  $m = 1, \ldots, 5$
HED network

Parameters: \((\alpha_1, \ldots, \alpha_5)\)

Inputs: \(s^1, \ldots, s^5\)

Outputs: \(f = \sum_{m=1}^{5} \alpha_m s^m\)
HED network

\[ l^m(W, w^{(m)}) = \sum_{j \in Y} w_{\hat{y}_j} S(\hat{y}_j, s_j^m) \quad s_j^m = \langle w^{(m)}, f_j \rangle \]

\[ \mathcal{L}_{side}(W, w) = \sum_{m=1}^{M} \alpha_m l^m(W, w^{(m)}) \]

\[ \mathcal{L}_{HED}(W, w, h) = \mathcal{L}_{side}(W, w) + \mathcal{L}_{fuse}(W, w, h) \]
This work in a nutshell

Starting point:
Holistically-Nested Edge Detection,
S. Xie and Z. Tu, ICCV 2015

Learning Techniques:
Multiple Instance Learning for Boundary Detection
Graduated Deep Supervised Networks

Network Architecture:
Tied Multi-Scale Networks
Grouping in DCNNs
Ambiguity in boundary annotations

Common interpretation, but different position information!
Ambiguity in boundary annotations

Solution: take into account annotator inaccuracies
Ambiguity in boundary annotations

For every positive point, gather set of locations that can `support` it.

False negative if no such point leads to a positive decision.

<table>
<thead>
<tr>
<th>Method</th>
<th>Baseline</th>
<th>MIL</th>
<th>G-DSN</th>
<th>M-Scale</th>
<th>VOC</th>
<th>Grouping</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODS</td>
<td>0.7781</td>
<td>0.7863</td>
<td>0.7892</td>
<td>0.8033</td>
<td>0.8086</td>
<td>0.8134</td>
</tr>
<tr>
<td>OIS</td>
<td>0.7961</td>
<td>0.8083</td>
<td>0.8106</td>
<td>0.8196</td>
<td>0.8268</td>
<td>0.8308</td>
</tr>
<tr>
<td>AP</td>
<td>0.804</td>
<td>0.802</td>
<td>0.789</td>
<td>0.8483</td>
<td>0.861</td>
<td>0.866</td>
</tr>
</tbody>
</table>
This work in a nutshell

Starting point:
Holistically-Nested Edge Detection,
S. Xie and Z. Tu, ICCV 2015

Learning Techniques:
Multiple Instance Learning for Boundary Detection
Graduated Deep Supervised Networks

Network Architecture:
Tied Multi-Scale Networks
Spectral Clustering in DCNNs
Holistically-Nested Edge Detection Training

\[ \mathcal{L}(W, w, h) = \mathcal{L}_{side}(W, w) + \mathcal{L}_{fuse}(W, w, h) \]

DSN’s side losses: steer network parameters to correct values

\[ \mathcal{L}^{(t)}(W, w, h) = (1 - \frac{t}{T})\mathcal{L}_{side}(W, w) + \mathcal{L}_{fuse}(W, w, h) \]

Graduated DSN: remove side losses as training progresses

<table>
<thead>
<tr>
<th>Method</th>
<th>Baseline</th>
<th>MIL</th>
<th>G-DSN</th>
<th>M-Scale</th>
<th>VOC</th>
<th>Grouping</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODS</td>
<td>0.7781</td>
<td>0.7863</td>
<td>0.7892</td>
<td>0.8033</td>
<td>0.8086</td>
<td>0.8134</td>
</tr>
<tr>
<td>OIS</td>
<td>0.7961</td>
<td>0.8083</td>
<td>0.8106</td>
<td>0.8196</td>
<td>0.8268</td>
<td>0.8308</td>
</tr>
<tr>
<td>AP</td>
<td>0.804</td>
<td>0.802</td>
<td>0.789</td>
<td>0.8483</td>
<td>0.861</td>
<td>0.866</td>
</tr>
</tbody>
</table>
This work in a nutshell

Starting point:
Holistically-Nested Edge Detection,
S. Xie and Z. Tu, ICCV 2015

Learning Techniques:
Multiple Instance Learning for Boundary Detection
Graduated Deep Supervised Networks

Network Architecture:
Tied Multi-Scale Networks
Grouping in DCNNs
Boundary CNN scale-space
Boundary CNN scale-space

\[ \frac{1}{2} \]
Boundary CNN scale-space
Boundary CNN scale-space
Boundary CNN scale-space
Multi-Scale DSN

Image Pyramid  Tied CNN outputs  Scale fusion
Multi-Scale DSN

-tied weights  -end-to-end training

<table>
<thead>
<tr>
<th>Method</th>
<th>Baseline</th>
<th>MIL</th>
<th>G-DSN</th>
<th>M-Scale</th>
<th>VOC</th>
<th>Grouping</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODS</td>
<td>0.7781</td>
<td>0.7863</td>
<td>0.7892</td>
<td>0.8033</td>
<td>0.8086</td>
<td>0.8134</td>
</tr>
<tr>
<td>OIS</td>
<td>0.7961</td>
<td>0.8083</td>
<td>0.8106</td>
<td>0.8196</td>
<td>0.8268</td>
<td>0.8308</td>
</tr>
<tr>
<td>AP</td>
<td>0.804</td>
<td>0.802</td>
<td>0.789</td>
<td>0.8483</td>
<td>0.861</td>
<td>0.866</td>
</tr>
</tbody>
</table>
## Pascal Context Dataset


<table>
<thead>
<tr>
<th>Method</th>
<th>Baseline</th>
<th>MIL</th>
<th>G-DSN</th>
<th>M-Scale</th>
<th>VOC</th>
<th>Grouping</th>
</tr>
</thead>
<tbody>
<tr>
<td>ODS</td>
<td>0.7781</td>
<td>0.7863</td>
<td>0.7892</td>
<td>0.8033</td>
<td>0.8086</td>
<td>0.8134</td>
</tr>
<tr>
<td>OIS</td>
<td>0.7961</td>
<td>0.8083</td>
<td>0.8106</td>
<td>0.8196</td>
<td>0.8268</td>
<td>0.8308</td>
</tr>
<tr>
<td>AP</td>
<td>0.804</td>
<td>0.802</td>
<td>0.789</td>
<td>0.8483</td>
<td>0.861</td>
<td>0.866</td>
</tr>
</tbody>
</table>

- tied weights  
- end-to-end training  
- more data 😊
This work in a nutshell

Starting point:
Holistically-Nested Edge Detection,
S. Xie and Z. Tu, ICCV 2015

Learning Techniques:
Multiple Instance Learning for Boundary Detection
Graduated Deep Supervised Networks

Network Architecture:
Tied Multi-Scale Networks
Grouping in DCNNs
This work in a nutshell

Starting point:
Holistically-Nested Edge Detection,
S. Xie and Z. Tu, ICCV 2015

Learning Techniques:
Multiple Instance Learning for Boundary Detection
Graduated Deep Supervised Networks

Network Architecture:
Tied Multi-Scale Networks
Grouping in DCNNs

Shi & Malik, Normalized Cuts and Image Segmentation. PAMI 2000
Catanzaro et. al.: Efficient, high-quality image contour detection. ICCV 2009
FCNNs + Spectral Clustering
FCNNs + Spectral Clustering
FCNNs + Spectral Clustering

\[(D - W)y = \lambda Dy\]
FCNNs + Spectral Clustering

\[(D - W)y = \lambda Dy\]
FCNNs + Spectral Clustering

\[(D - W)y = \lambda Dy\]
FCNNs + Spectral Clustering

\[(D - W)y = \lambda Dy\]
FCNNs + Spectral Clustering

\[(D - W)y = \lambda Dy\]

Catanzaro et al.: Efficient, high-quality image contour detection. ICCV 2009
-Global Pb: ~60 seconds (CPU)  -spectralPb layer: 0.2 seconds (GPU)
All-in-one caffe network, ~1 second per frame
Progress in edge detection

I. Kokkinos, Pushing the boundaries of boundary detection using deep learning, ICLR 2016

Recall vs Precision plot showing different methods:
- Human: F = 0.8027
- Grouping: F = 0.8134
- VOC data: F = 0.8086
- Multi-res: F = 0.8033
- G-DSN: F = 0.7893
- MIL: F = 0.7875
- Baseline: F = 0.7781

This work is indicated by a red arrow pointing to a point on the curve, suggesting an improvement over the baseline.
One last trick!


I. Kokkinos, Pushing the boundaries of boundary detection using deep learning, ICLR 2016
2015: Deeplab: FCNNs + DenseCRF

2016: Combine with spectral embedding
2016: Combine with spectral embedding

Boundaries  Top-3 eigenvectors  unaries  posterior
## Spectral embedding + DenseCRF

<table>
<thead>
<tr>
<th>Method</th>
<th>mAP %</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adelaide-Context-CNN-CRF-COCO (Lin et al., 2015)</td>
<td>77.8</td>
</tr>
<tr>
<td>CUHK-DPN-COCO (Liu et al., 2015)</td>
<td>77.5</td>
</tr>
<tr>
<td>Adelaide-Context-CNN-CRF-COCO (Lin et al., 2015)</td>
<td>77.2</td>
</tr>
<tr>
<td>MSRA-BoxSup (Dai et al., 2015)</td>
<td>75.2</td>
</tr>
<tr>
<td>Oxford-TVG-CRF-RNN-COCO (Zheng et al., 2015)</td>
<td>74.7</td>
</tr>
<tr>
<td>DeepLab-MSc-CRF-LF-COCO-CJ (Chen et al., 2015)</td>
<td>73.9</td>
</tr>
<tr>
<td>DeepLab-CRF-COCO-LF(Chen et al., 2015)</td>
<td>72.7</td>
</tr>
<tr>
<td>Multi-Scale DeepLab</td>
<td>72.1</td>
</tr>
<tr>
<td>Multi-Scale DeepLab-CRF</td>
<td>74.8</td>
</tr>
<tr>
<td>Multi-Scale DeepLab-CRF-Embeddings</td>
<td>75.4</td>
</tr>
<tr>
<td>Multi-Scale DeepLab-CRF-Embeddings-GraphCuts</td>
<td>75.7</td>
</tr>
</tbody>
</table>

I. Kokkinos, Pushing the boundaries of boundary detection using deep learning, ICLR 2016
Bottom-up alternative: metric learning

This talk: controlling DCNNs for low- and high-level tasks

- Classification & Detection
- Semantic Segmentation
- Boundary Detection
- Feature Descriptors

E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, F. Moreno-Noguer,
Discriminative Learning of Deep Convolutional Descriptors, ICCV15
Discriminative learning of Deep Convolutional Feature Point Descriptors

Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, Francesc Moreno-Noguer

https://github.com/cvlab-epfl/deepdesc-release
Advertisement #2

CVIU Special Issue on Deep Learning for CV

Submission deadline: April 16, 2016
Conclusion

2012 onwards: all about DCNNs

if [all] you have [is] a hammer, you treat everything like a nail

- Classification & Detection
- Semantic Segmentation
- Boundary Detection
- Feature Descriptors

2014 onwards: incorporating structure in DCNNs

trust is good, but control is better!
even better are results!

Thanks!