High Level Computer Vision

Deep Learning for Computer Vision
Part 3

Bernt Schiele - schiele@mpi-inf.mpg.de
Mario Fritz - fritz@cispa.saarland

https://www.mpi-inf.mpg.de/hlcv
Overview Today

• VGG-network - alternative to AlexNet
  ‣ Very Deep Convolutional Networks for Large-Scale Image Recognition, K. Simonyan, A. Zisserman, ICLR’15

• Deep residual learning for image recognition
  ‣ [He,Zhang,Ren,Sun@cvpr16] - https://arxiv.org/abs/1512.03385

• From detection to segmentation
  ‣ Main Reading: Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs, Chen, Papandreou, Kokkins, Murphy, Yuille, ICLR’15 - https://arxiv.org/abs/1412.7062
  ‣ Also
    - Hypercolumns for object segmentation and fine-grained localization
      Bharath Hariharan, Pablo Arbeláez, Ross Girshick, Jitendra Malik, CVPR’15
      https://arxiv.org/abs/1411.5752
    - Fully Convolutional Networks for Semantic Segmentation
      John Long, Evan Shelhamer, Trevor Darelle, CVPR’15
      https://arxiv.org/abs/1411.4038

• Cityscapes - https://www.cityscapes-dataset.com
Very Deep ConvNets for Large-Scale Image Recognition

Karen Simonyan, Andrew Zisserman
Visual Geometry Group, University of Oxford

ILSVRC Workshop
12 September 2014
Summary of VGG Submission

- Localisation task
  - 1st place, 25.3% error

- Classification task
  - 2nd place, 7.3% error

- Key component: very deep ConvNets
  - up to 19 weight layers
Effect of Depth

• How does ConvNet depth affect the performance?

• Comparison of ConvNets
  • same generic design – fair evaluation
  • increasing depth
  • from 11 to 19 weight layers
Network Design

Key design choices:
- 3x3 conv. kernels – very small
- conv. stride 1 – no loss of information

Other details:
- Rectification (ReLU) non-linearity
- 5 max-pool layers (x2 reduction)
- no normalisation
- 3 fully-connected (FC) layers
Discussion

Why 3x3 layers?

- Stacked conv. layers have a large receptive field
  - two 3x3 layers – 5x5 receptive field
  - three 3x3 layers – 7x7 receptive field
- More non-linearity
- Less parameters to learn
  - ~140M per net
Implementation

• Heavily-modified Caffe C++ toolbox
• Multiple GPU support
  • 4 x NVIDIA Titan, off-the-shelf workstation
  • data parallelism for training and testing
  • ~3.75 times speed-up, 2-3 weeks for training
Comparison – Fixed Training Size

Top-5 Classification Error (Val. Set)

<table>
<thead>
<tr>
<th>Layers</th>
<th>256</th>
<th>384</th>
</tr>
</thead>
<tbody>
<tr>
<td>13 layers</td>
<td>9.4</td>
<td>9.4</td>
</tr>
<tr>
<td>16 layers</td>
<td>8.8</td>
<td>8.7</td>
</tr>
<tr>
<td>19 layers</td>
<td>9.0</td>
<td>8.7</td>
</tr>
</tbody>
</table>

- 16 or 19 layers trained on 384xN images are the best.
Comparison – Random Training Size

Top-5 Classification Error (Val. Set)

- Training scale jittering is better than fixed scales
- Before submission: single net, FC-layers tuning
Comparison – Random Training Size

Top-5 Classification Error (Val. Set)

- Training scale jittering is better than fixed scales
- After submission: three nets, all-layers tuning
Final Results

Top-5 Classification Error (Test Set)

- **2nd place with 7.3% error**
  - combination of 7 models: 6 fixed-scale, 1 multi-scale
- Single model: 8.4% error
Final Results (Post-Competition)

Top-5 Classification Error (Test Set)

- 2\textsuperscript{nd} place with 7.0% error
  - combination of two multi-scale models (16- and 19-layer)
- Single model: 7.3% error
Localisation

Our localisation method

• Builds on very deep classification ConvNets
• Similar to OverFeat
  1. Localisation ConvNet predicts a set of bounding boxes
  2. Bounding boxes are merged
  3. Resulting boxes are scored by a classification ConvNet
Localisation (2)

- Last layer predicts a bbox **for each class**
  - Bbox parameterisation: \((x,y,w,h)\)
  - 1000 classes \(\times 4\)-D / class = 4000-D

- Training
  - Euclidean loss
  - initialised with a classification net
  - fine-tuning of **all** layers
**Final Results**

**Top-5 Localisation Error (Test Set)**

- VGG: 25.3%
- GoogleNet: 26.4%
- OverFeat (2013): 29.9%
- SYSU: 31.9%

- **1st place with 25.3% error**
  - combination of 2 localisation models
Summary

• Excellent results using classical ConvNets
  • small receptive fields
  • but very deep → lots of non-linearity
• Depth matters!
• Details in the arXiv pre-print: arxiv.org/pdf/1409.1556/

VGG Team ILSVRC Progress

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the GPUs used for this research.
1. Deep Residual Learning for Image Recognition

- Deep residual learning for image recognition
  He, Zhang, Ren, Sun@cvpr16

- Following slides from first authors of the paper: Kaiming He
Deep Residual Learning for Image Recognition

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun

work done at
Microsoft Research Asia
ResNet @ ILSVRC & COCO 2015 Competitions

**1st places in all five main tracks**

- **ImageNet Classification**: “Ultra-deep” 152-layer nets
- **ImageNet Detection**: 16% better than 2nd
- **ImageNet Localization**: 27% better than 2nd
- **COCO Detection**: 11% better than 2nd
- **COCO Segmentation**: 12% better than 2nd

*improvements are relative numbers

Revolution of Depth

ImageNet Classification top-5 error (%)

ILSVRC'15 ResNet: 3.6
ILSVRC'14 VGG: 7.3
ILSVRC'12 AlexNet: 11.7
ILSVRC'10: 28.2

Shallow

Revolution of Depth

PASCAL VOC 2007 Object Detection mAP (%)

Engines of visual recognition

- HOG, DPM (shallow)
- AlexNet (RCNN) (8 layers) mAP: 58%
- VGG (RCNN) (16 layers) mAP: 66%
- ResNet (Faster RCNN)* (101 layers) mAP: 86%

*with other improvements & more data

Revolution of Depth

AlexNet, 8 layers (ILSVRC 2012)

11x11 conv, 96, /4, poo/2

5x5 conv, 256, poo/2

3x3 conv, 384

3x3 conv, 384

3x3 conv, 256, poo/2

fc, 4096

fc, 4096

fc, 1000

Revolution of Depth

AlexNet, 8 layers (ILSVRC 2012)

VGG, 19 layers (ILSVRC 2014)

GoogleNet, 22 layers (ILSVRC 2014)

Revolution of Depth

AlexNet, 8 layers (ILSVRC 2012)

VGG, 19 layers (ILSVRC 2014)

ResNet, 152 layers (ILSVRC 2015)

Revolution of Depth

ResNet, 152 layers

Is learning better networks as simple as stacking more layers?

Simply stacking layers?

- **Plain** nets: stacking 3x3 conv layers…
- 56-layer net has **higher training error** and test error than 20-layer net

Simply stacking layers?

- “Overly deep” plain nets have higher training error
- A general phenomenon, observed in many datasets

a shallower model (18 layers)

a deeper counterpart (34 layers)

- Richer solution space
- A deeper model should not have higher training error
- A solution by construction:
  - original layers: copied from a learned shallower model
  - extra layers: set as identity
  - at least the same training error
- Optimization difficulties:
solvers cannot find the solution when going deeper…

Deep Residual Learning

- Plain net

\[ H(x) \text{ is any desired mapping, hope the 2 weight layers fit } H(x) \]

Deep Residual Learning

• Residual net

\[ F(x) \]

\[ H(x) = F(x) + x \]

\[ x \]

Weight layer

Relu

Weight layer

Identity

Relu

\[ H(x) \text{ is any desired mapping, hope the 2 weight layers fit } H(x) \]

\[ \text{hope the 2 weight layers fit } F(x) \]

\[ \text{let } H(x) = F(x) + x \]
Deep Residual Learning

- $F(x)$ is a residual mapping w.r.t. identity

If identity were optimal, easy to set weights as 0

If optimal mapping is closer to identity, easier to find small fluctuations

Network “Design”

• Keep it simple

• Our basic design (VGG-style)
  • all 3x3 conv (almost)
  • spatial size /2 => # filters x2
  • Simple design; just deep!

CIFAR-10 experiments

- Deep ResNets can be trained without difficulties
- Deeper ResNets have lower training error, and also lower test error

Deep ResNets can be trained without difficulties
Deeper ResNets have **lower training error**, and also lower test error

ImageNet experiments

- Deeper ResNets have lower error

10-crop testing, top-5 val error (%)

- this model has lower time complexity than VGG-16/19

“Features matter.”
(quote [Girshick et al. 2014], the R-CNN paper)

<table>
<thead>
<tr>
<th>task</th>
<th>2nd-place winner</th>
<th>ResNets</th>
<th>margin (relative)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImageNet Localization</td>
<td>12.0</td>
<td>9.0</td>
<td>27%</td>
</tr>
<tr>
<td>(top-5 error)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ImageNet Detection</td>
<td>53.6</td>
<td>62.1</td>
<td>16%</td>
</tr>
<tr>
<td>(mAP@.5)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COCO Detection</td>
<td>33.5</td>
<td>37.3</td>
<td>11%</td>
</tr>
<tr>
<td>(mAP@.5:.95)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>COCO Segmentation</td>
<td>25.1</td>
<td>28.2</td>
<td>12%</td>
</tr>
<tr>
<td>(mAP@.5:.95)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Our results are all based on **ResNet-101**
- Our features are **well transferrable**

Object Detection (brief)

- Simply “Faster R-CNN + ResNet"

<table>
<thead>
<tr>
<th>Faster R-CNN baseline</th>
<th>mAP@.5</th>
<th>mAP@.5:.95</th>
</tr>
</thead>
<tbody>
<tr>
<td>VGG-16</td>
<td>41.5</td>
<td>21.5</td>
</tr>
<tr>
<td>ResNet-101</td>
<td>48.4</td>
<td>27.2</td>
</tr>
</tbody>
</table>

COCO detection results
(ResNet has 28% relative gain)

Our results on MS COCO

Results on real video. Model trained on MS COCO w/ 80 categories.
(frame-by-frame; no temporal processing)


this video is available online: https://youtu.be/WZmSMkK9VuA
More Visual Recognition Tasks

ResNets lead on these benchmarks (incomplete list):

• **ImageNet** classification, detection, localization
• **MS COCO** detection, segmentation

• **PASCAL VOC** detection, segmentation
• **VQA** challenge 2016

• Human pose estimation [Newell et al 2016]
• Depth estimation [Laina et al 2016]
• Segment proposal [Pinheiro et al 2016]
• …
Potential Applications

ResNets have shown outstanding or promising results on:

- Visual Recognition
- Image Generation (Pixel RNN, Neural Art, etc.)
- Natural Language Processing (Very deep CNN)
- Speech Recognition (preliminary results)
- Advertising, user prediction (preliminary results)

Conclusions

• Deep Residual Learning:
  • Ultra deep networks can be easy to train
  • Ultra deep networks can simply gain accuracy from depth
  • Ultra deep representations are well transferrable

• Follow-up [He et al. arXiv 2016]
  • 200 layers on ImageNet, 1000 layers on CIFAR
Resources

• Models and Code
  • Our ImageNet models in Caffe: https://github.com/KaimingHe/deep-residual-networks

• Many available implementations:
  (list in https://github.com/KaimingHe/deep-residual-networks)
  • Facebook AI Research’s Torch ResNet: https://github.com/facebook/fb.resnet.torch
    • Torch, CIFAR-10, with ResNet-20 to ResNet-110, training code, and curves: code
    • Lasagne, CIFAR-10, with ResNet-32 and ResNet-56 and training code: code
    • Neon, CIFAR-10, with pre-trained ResNet-32 to ResNet-110 models, training code, and curves: code
    • Torch, MNIST, 100 layers: blog, code
    • A winning entry in Kaggle's right whale recognition challenge: blog, code
    • Neon, Place2 (mini), 40 layers: blog, code
    • .......

Thank You!

2. From detection to segmentation

- Main Reading:

- Also
Fully Convolutional Neural Networks for Classification, Detection & Segmentation

or, all your computer wanted to know about horses

Iasonas Kokkinos
Ecole Centrale Paris / INRIA Saclay

& G. Papandreou, P.-A. Savalle, S. Tsogkas, L-C Chen, K. Murphy, A. Yuille, A. Vedaldi
Fully convolutional neural networks
Fully convolutional neural networks

**Fully connected layers: 1x1 spatial convolution kernels**

Allows network to process images of arbitrary size

P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus and Y. LeCun, OverFeat, ICLR, 2014

M. Oquab, L. Bottou, I. Laptev, J. Sivic, Weakly Supervised Object Recognition with CNNs, TR2014

J. Long, E. Shelhamer, T. Darrell, Fully Convolutional Networks for Semantic Segmentation, CVPR 15
Sliding Window with ConvNet

Input Window

Feature Extractor

Classifier

Clasess
Sliding Window with ConvNet

No need to compute two separate windows
Just one big input window, computed in a single pass
Fully convolutional neural networks

Fast  (shared convolutions)
Simple (dense)
Part 2: FCNNs for semantic segmentation

L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. Yuille
Semantic segmentation task
System outline

Input → Deep Convolutional Neural Network → Aeroplane Coarse Score map → Bi-linear Interpolation

Final Output → Fully Connected CRF

J. Long, E. Shelhamer, T. Darrell, FCNNs for Semantic Segmentation, CVPR 15
P. Krähenbühl and V. Koltun, Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials, NIPS 2011
Repurposing DCNNs for semantic segmentation

- Accelerate CNN evaluation by ‘hard dropout’ & finetuning
  - In VGG: Subsample first FC layer $7 \times 7 \rightarrow 3 \times 3$

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

“Hole” algorithm

- “Normal” Resolution
  - Black: Filter width = 3, Stride = 2
- Increase Resolution by Factor of 2:
  - Magenta: same Filter with width 3, Stride = 1
“Hole” algorithm

- skip subsampling
  - in their case for VGG-net: after the last two max-pooling layers
- for the next layer filter: sparsely sample the feature map with “input stride” 2 (or 4 respectively)

Figure 1: Illustration of the hole algorithm in 1-D, when $\text{kernel\_size} = 3$, $\text{input\_stride} = 2$, and $\text{output\_stride} = 1$. 
FCNN-DCRF: Full & densely connected

- Large CNN receptive field:
  + good accuracy
  - worse performance near boundaries
- Dense CRF: sharpen boundaries using image-based info

P. Krähenbühl and V. Koltun, Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials, NIPS 2011
CRF - Conditional Random Field

- Energy function to be minimized

\[ E(x) = \sum_i \theta_i(x_i) + \sum_{ij} \theta_{ij}(x_i, x_j) \]

- with unary terms obtained from the CNN:

\[ \theta_i(x_i) = -\log P(x_i) \]

- and pairwise terms (Potts model)

\[ \theta_{ij}(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{K} w_m \cdot k^m(f_i, f_j) \]

- with

\[ \mu(x_i, x_j) = 1 \text{ if } x_i \neq x_j. \]
Indicative Results

Raw score maps

After dense CRF
Indicative Results

Raw score maps  After dense CRF
Indicative Results

Raw score maps

After dense CRF
Indicative Results

Raw score maps
After dense CRF
### Improvements due to fully-connected CRF

<table>
<thead>
<tr>
<th>Method</th>
<th>mean IOU (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeepLab</td>
<td>59.80</td>
</tr>
<tr>
<td>DeepLab-CRF</td>
<td>63.74</td>
</tr>
<tr>
<td>DeepLab-MSc</td>
<td>61.30</td>
</tr>
<tr>
<td>DeepLab-MSc-CRF</td>
<td>65.21</td>
</tr>
</tbody>
</table>

Krahenbuhl et. al. (TextonBoost unaries)
27.6 -> 29.1 (+1.5)

### Improvements due to Dense CRF

Our work (FCNN unaries)
61.3 -> 65.21 (+3.9)
Another fully convolutional network for semantic segmentation (without CRF)

Comparisons to Fully Convolutional Net

Ground-truth

FCN-8

Our work

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%
3. Cityscapes Dataset

• Dataset for semantic labeling and “understanding”
  ‣ Cordts, Omaran, Ramos, Rehfeld, Enzweiler, Benenson, Franke, Roth, Schiele @ cvpr16
  ‣ https://www.cityscapes-dataset.net
  ‣ http://arxiv.org/abs/1604.01685
The Cityscapes Dataset
for Semantic Scene Labeling and Understanding

https://www.cityscapes-dataset.net

Marius Cordts\(^1,3\)
Timo Rehfeld\(^1,3\)
Uwe Franke\(^1\)

Mohamed Omran\(^2\)
Markus Enzweiler\(^1\)
Stefan Roth\(^3\)

Sebastian Ramos\(^1\)
Rodrigo Benenson\(^2\)
Bernt Schiele\(^2\)

1
DAIMLER

2
MPH

3
TECHNISCHE UNIVERSITÄT DARMSTADT
Previous Work

**KITTI** [Geiger et al. ‘12]
- stereo video
- no official semantic labeling or instance labeling challenge

**CamVid** [Brostow et al., to appear]
- monocular video

**Daimler Urban Scenes** [Scharwächter et al. ‘14]
- stereo video
- limited number of classes / annotation density
Overview

https://www.cityscapes-dataset.net

- 2 MP automotive-grade CMOS cameras (OnSemi AR0331)
- 1/3” sensor, 17Hz, rolling shutter
- 16 bit linear intensity HDR
- + 8-bit tonemapped LDR
- stereo setup (22cm baseline)
  - 30 frame video snippets (~2/3 of the dataset)
  - + long videos (remaining ~1/3)
Example video snippet

https://www.cityscapes-dataset.net
Example video snippet

https://www.cityscapes-dataset.net
Overview

https://www.cityscapes-dataset.net

- precomputed disparity
Overview

https://www.cityscapes-dataset.net

- in-vehicle odometry
- outside temperature
- GPS tracks
Labels

https://www.cityscapes-dataset.net

- 8 categories – 30 classes
  - instance-level annotations for all vehicles & humans
- 19 classes evaluated
  - rare cases excluded
Dense Labeling: 5,000 images

https://www.cityscapes-dataset.net

- 2975 training images
- 500 validation images
- 1525 test images (for benchmark)
- annotated 20th frame from every video snippet
- instance labels for dynamic classes
Coarse Labeling: 20,000 images

https://www.cityscapes-dataset.net

- all for weakly-supervised training
- annotated every 20th frame from long video
Objective: Complexity

https://www.cityscapes-dataset.net

- Complex, real-world scenes

most instances, most people
most cars
most bicyles
fewest instances
Objective: Diversity

- **50 cities**
  - across all of Germany
  - + Zürich + Strasbourg
  - KITTI, CamVid & DUS: 1 city only

- **3 seasons**
  - spring, summer, fall
  - winter purposely excluded

- **fair weather**
  - rain & snow are excluded
  - daytime only
Comparison to Previous Datasets

<table>
<thead>
<tr>
<th>dataset size &amp; density</th>
<th>#pixels [10^9]</th>
<th>annot. density [%]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours (fine)</td>
<td>9.41</td>
<td>97.0</td>
</tr>
<tr>
<td>Ours (coarse)</td>
<td>26.0</td>
<td>67.5</td>
</tr>
<tr>
<td>CamVid</td>
<td>0.62</td>
<td>96.2</td>
</tr>
<tr>
<td>DUS</td>
<td>0.14</td>
<td>63.0</td>
</tr>
<tr>
<td>KITTI</td>
<td>0.23</td>
<td>88.9</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>instance statistics</th>
<th>#humans [10^3]</th>
<th>#vehicles [10^3]</th>
<th>#h/image</th>
<th>#v/image</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ours (fine)</td>
<td>24.2</td>
<td>49.1</td>
<td>7.0</td>
<td>14.1</td>
</tr>
<tr>
<td>KITTI</td>
<td>6.1</td>
<td>30.3</td>
<td>0.8</td>
<td>4.1</td>
</tr>
<tr>
<td>Caltech</td>
<td>192[^1]</td>
<td>-</td>
<td>1.5</td>
<td>-</td>
</tr>
</tbody>
</table>

- CamVid & DUS: no instance annotations
- KITTI: only bboxes

histogram of vehicle distances

[^1]: S. Long et al. (2015)
Control experiments

<table>
<thead>
<tr>
<th>Static Prediction</th>
<th>IoU (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coarse Labels</td>
<td>10.3%</td>
</tr>
<tr>
<td>Fine Labels</td>
<td>10.1%</td>
</tr>
<tr>
<td>GT Segm. + Coarse</td>
<td>10.9%</td>
</tr>
<tr>
<td>GT Segm. + Fine</td>
<td>10.1%</td>
</tr>
</tbody>
</table>

Bernt Schiele
Baselines

- fully convolutional network [Long et al. CVPR’15]
- CRF as RNN [Zheng et al. ICCV’15]
- “Adelaide” [Lin et al. CVPR’16]
- deepLab [Papandreou et al. ICCV’15]
- deep parsing network [Liu et al. ICCV’15]
- SegNet [Badrinarayanan et al. arXiv]
FCN Results
FCN Results
## Baselines - Quantitative Results

### Table

<table>
<thead>
<tr>
<th>Model</th>
<th>Classes IoU</th>
<th>Categories IoU</th>
<th>Train</th>
<th>Val</th>
<th>Coarse</th>
<th>Sub</th>
<th>Extended</th>
<th>4</th>
<th>56.1</th>
<th>34.2</th>
<th>79.8</th>
<th>66.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>FCN-8s</td>
<td>65.3</td>
<td>85.7</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>4</td>
<td>56.1</td>
<td>34.2</td>
<td>79.8</td>
<td>66.4</td>
</tr>
<tr>
<td>FCN-8s</td>
<td>61.9</td>
<td>81.6</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>2</td>
<td></td>
<td>4</td>
<td>57.0</td>
<td>32.0</td>
<td>79.1</td>
<td>61.9</td>
</tr>
<tr>
<td>FCN-8s</td>
<td>58.3</td>
<td>83.4</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>57.0</td>
<td>32.0</td>
<td>79.1</td>
<td>61.9</td>
</tr>
<tr>
<td>FCN-8s</td>
<td>58.0</td>
<td>78.2</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>4</td>
<td>57.0</td>
<td>32.0</td>
<td>79.1</td>
<td>61.9</td>
</tr>
</tbody>
</table>

### Notes
- Approximately 3,500 **finely** annotated images
- 500 **finely** annotated images
- 20,000 **coarsely** annotated images

### Literature
- Badrinarayanan et al. @ arXiv’15
- Badrinarayanan et al. @ arXiv’15
- Liu et al @ ICCV’15
- Zheng et al @ ICCV’15
- Chen et al. @ ICLR’15
- Papandreou et al @ ICCV’15
- Lin et al. @ CVPR’16
- Yu & Koltun @ ICLR’16
Cross-Dataset Generalization

https://www.cityscapes-dataset.net

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Best reported result</th>
<th>Our result</th>
</tr>
</thead>
<tbody>
<tr>
<td>KITTI [53]</td>
<td>61.6 [3]</td>
<td>70.9</td>
</tr>
<tr>
<td>KITTI [59]</td>
<td>82.2 [65]</td>
<td>81.2</td>
</tr>
</tbody>
</table>

FCN [Long et al. CVPR’15] trained on Cityscapes
Cityscapes: Conclusions

https://www.cityscapes-dataset.net

- Cityscapes is the largest and most diverse datasets for semantic segmentation of urban street scenes
  - aim is to become the standard dataset for
    - scene labeling (urban scenarios)
    - instance segmentation (people, cars, etc)
  - planned as dynamic entity which will be expanded & adapted

- Recent CNNs approaches:
  - already achieve very good results
  - impressive cross-dataset generalization
  - using coarse annotations only leads to reduced performance