# High Level Computer Vision

# Deep Learning for Computer Vision
# Part 3

Bernt Schiele - schiele@mpi-inf.mpg.de
Mario Fritz - mfritz@mpi-inf.mpg.de

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](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

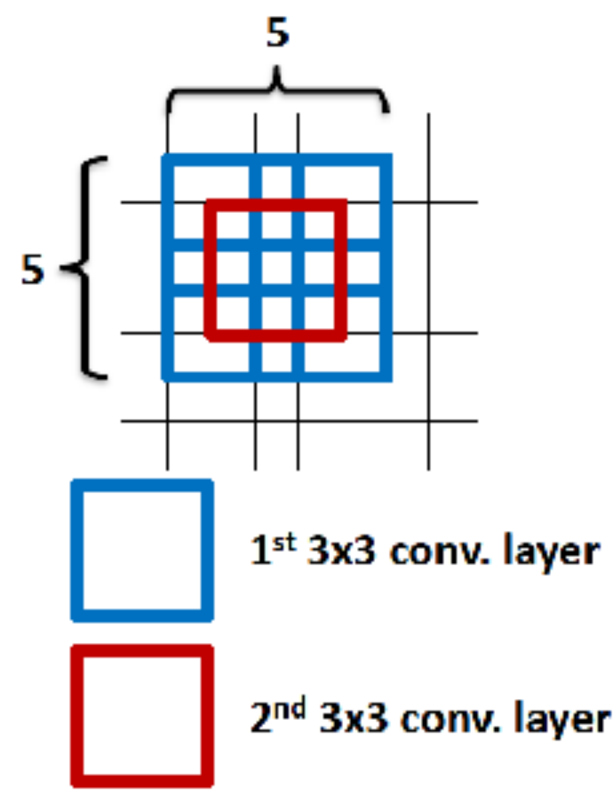
Other details:

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

# Discussion

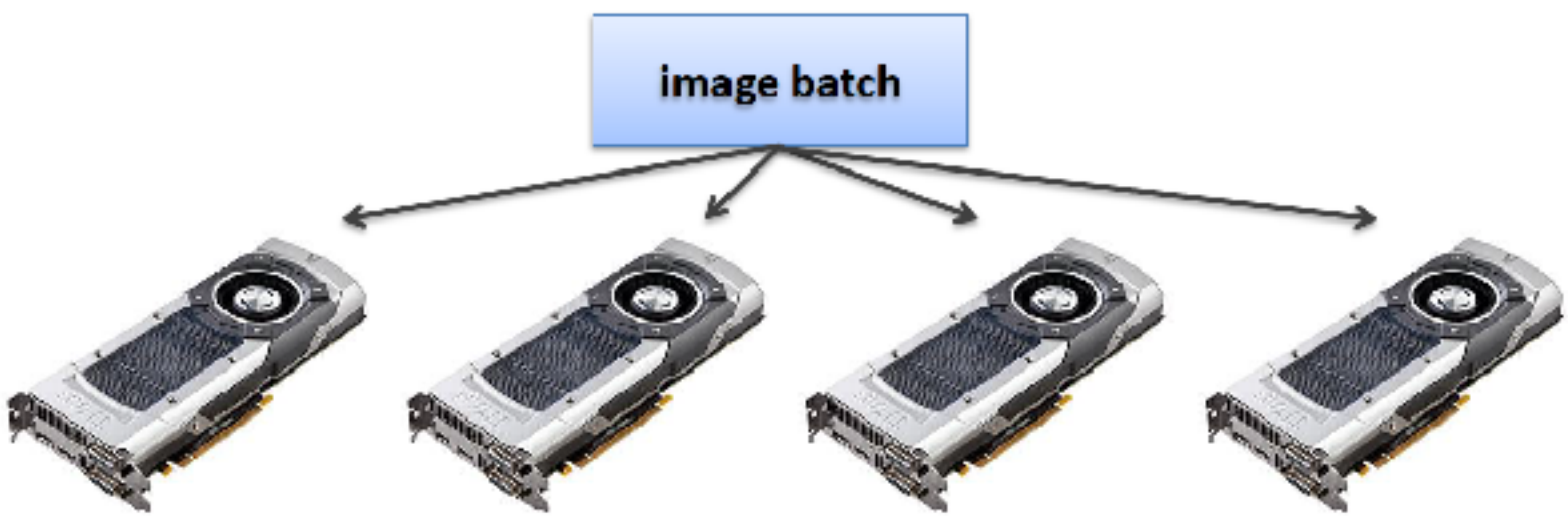
Why 3x3 layers?

- 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

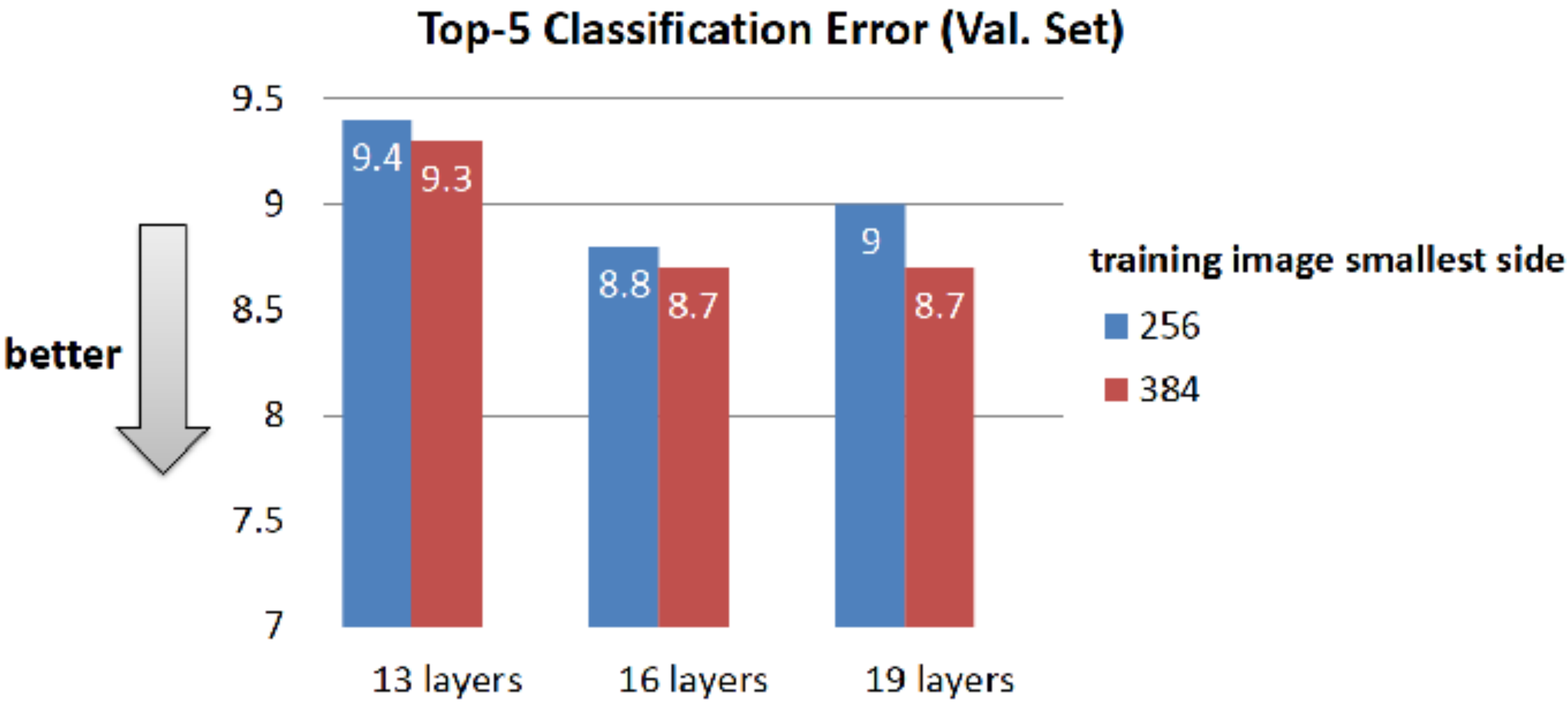


# Implementation

- 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

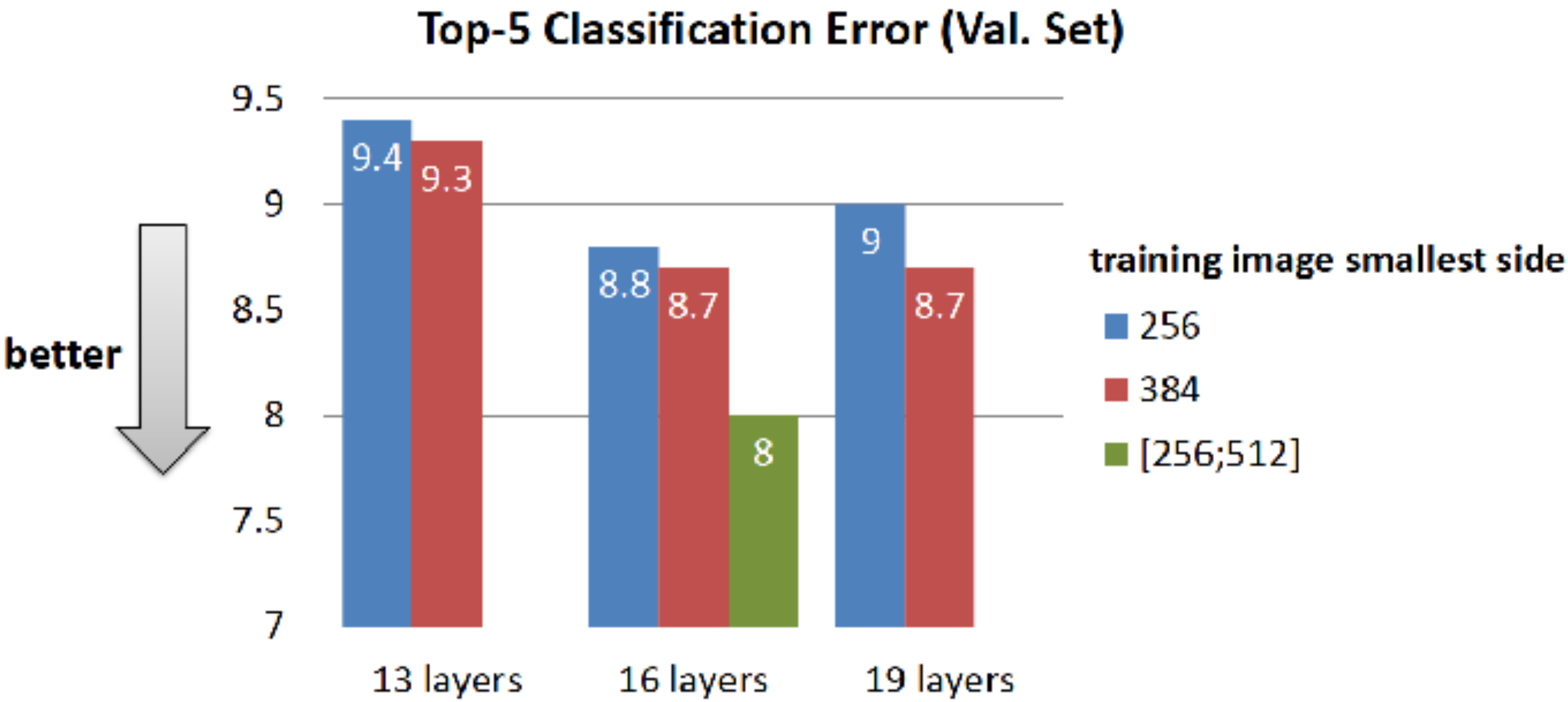


# Comparison – Fixed Training Size



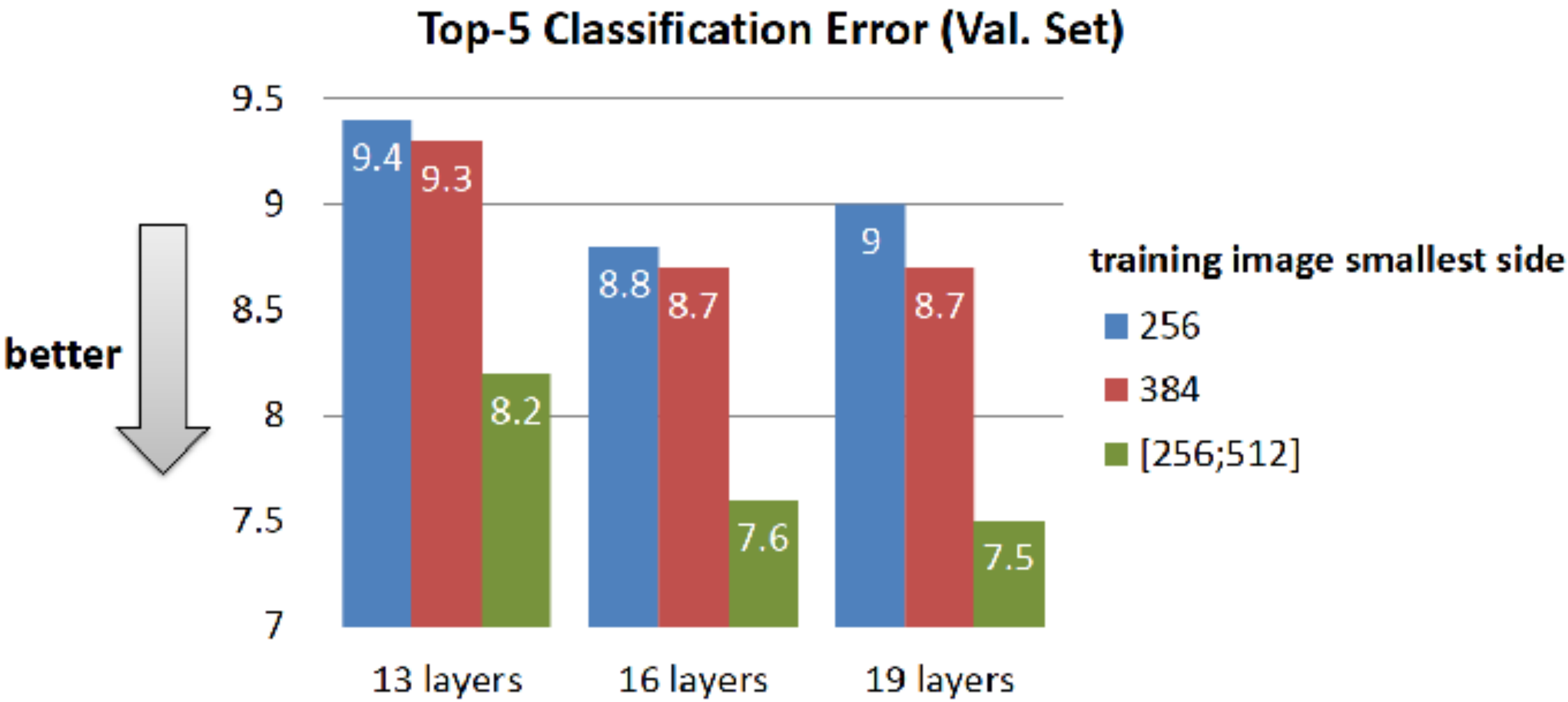

- 16 or 19 layers trained on 384xN images are the best

# Comparison – Random Training Size



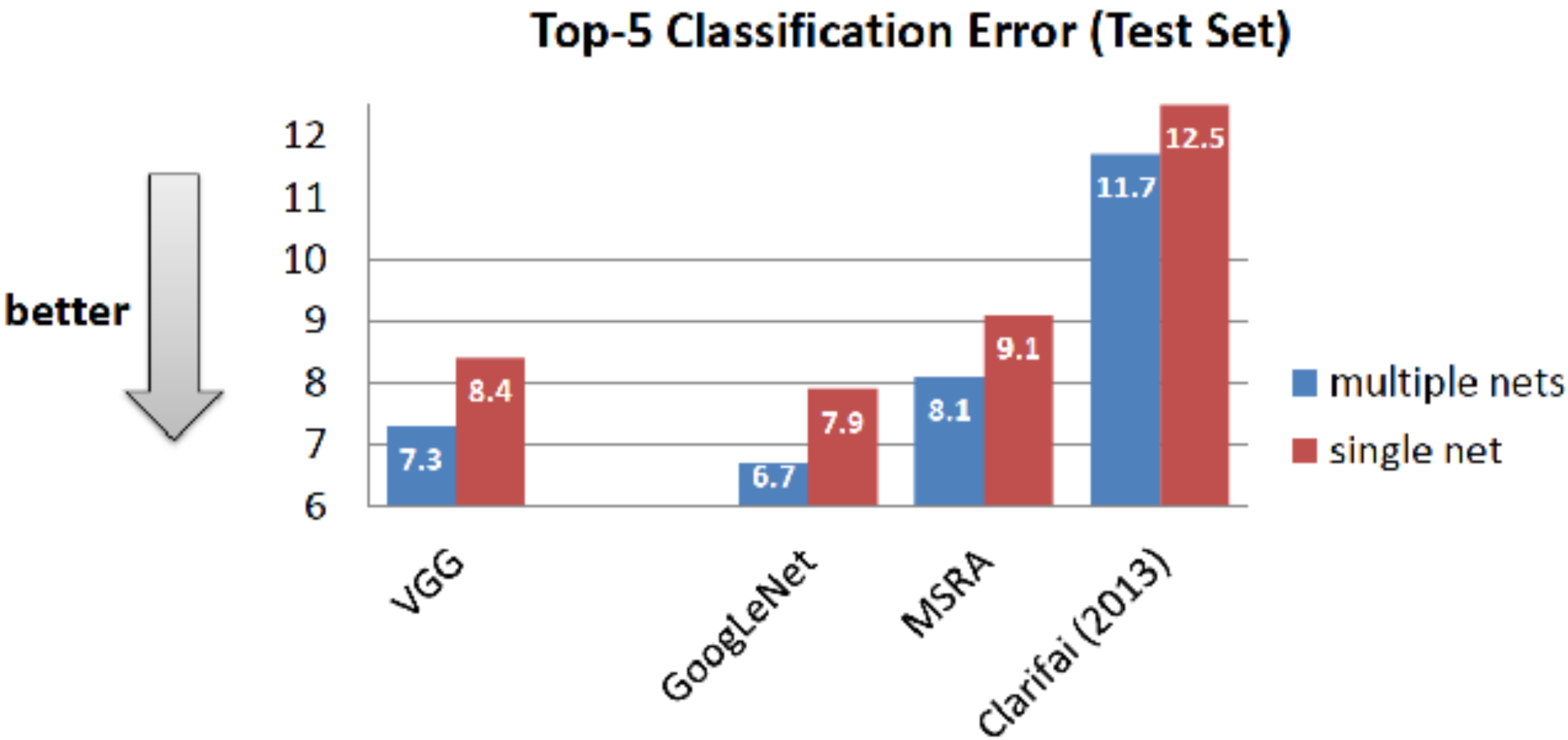

- Training scale jittering is better than fixed scales
- Before submission: single net, FC-layers tuning

# Comparison – Random Training Size



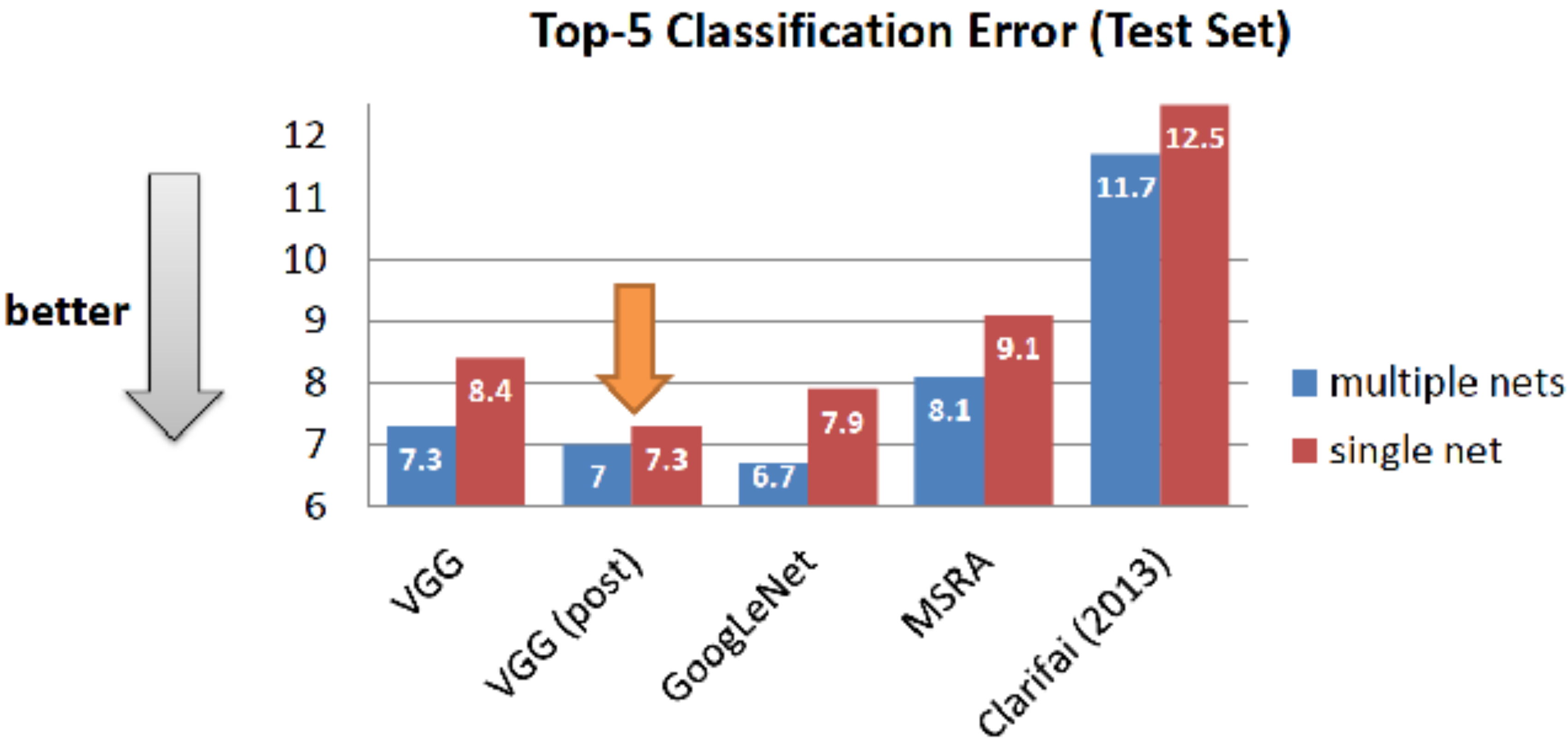

- Training scale jittering is better than fixed scales
- After submission: three nets, all-layers tuning

# Final Results




- 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)




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

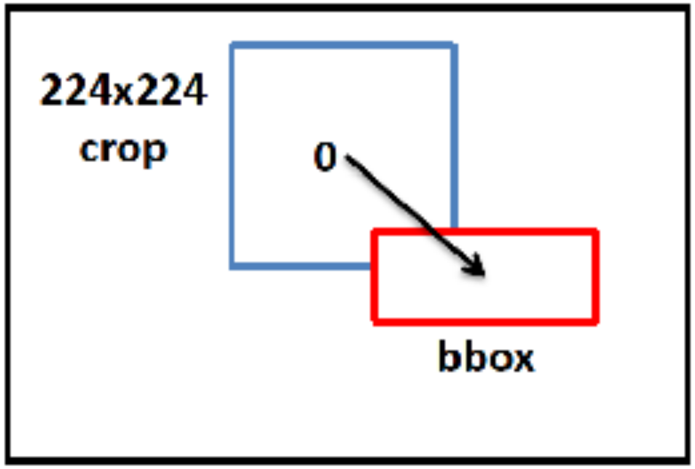
# Localisation

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

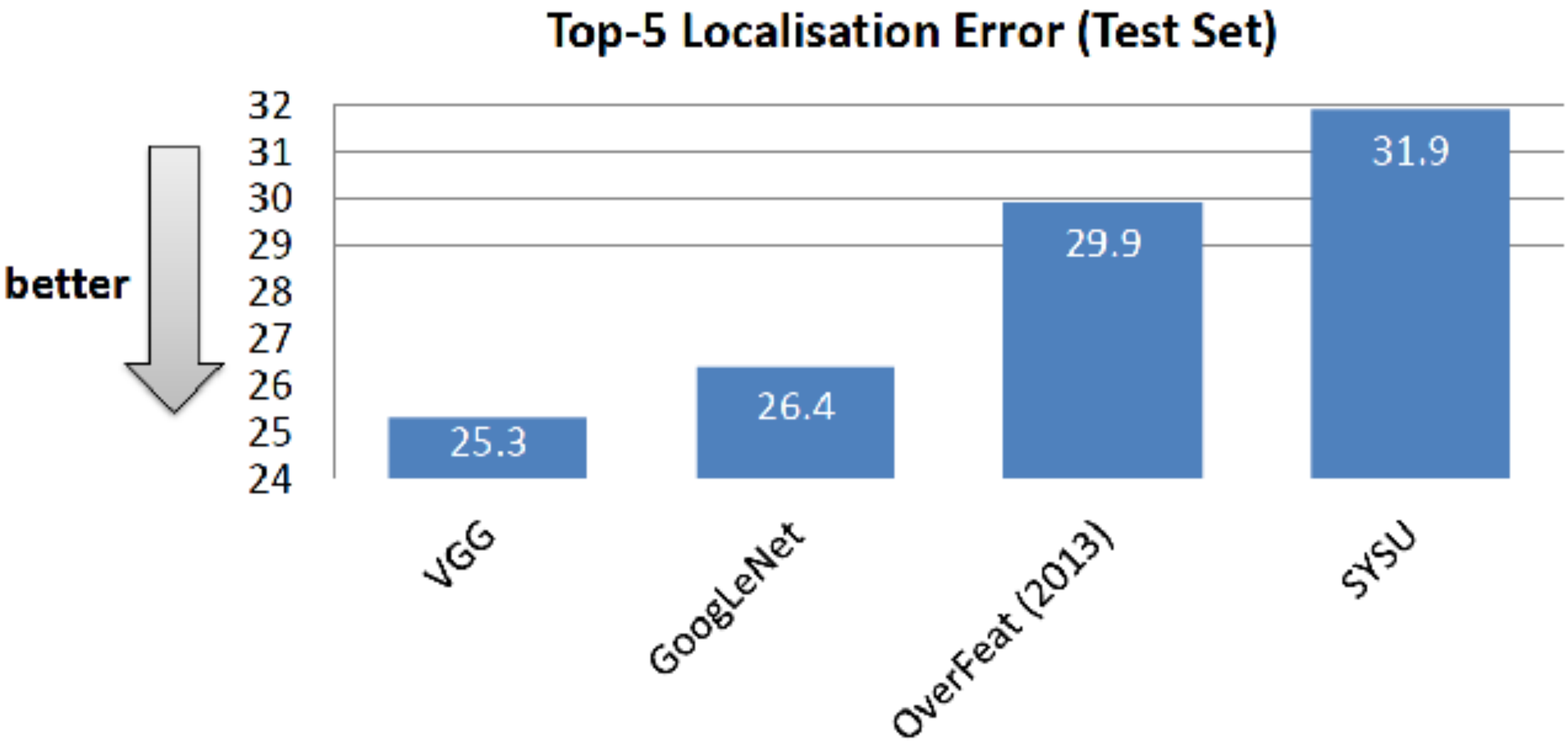
# Localisation (2)

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




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

# Final Results




- 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
  https://arxiv.org/abs/1512.03385

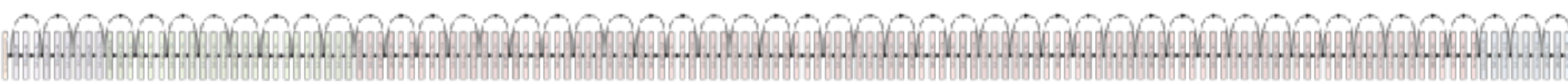
- Following slides from first authors of the paper: **Kaiming He**

# 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

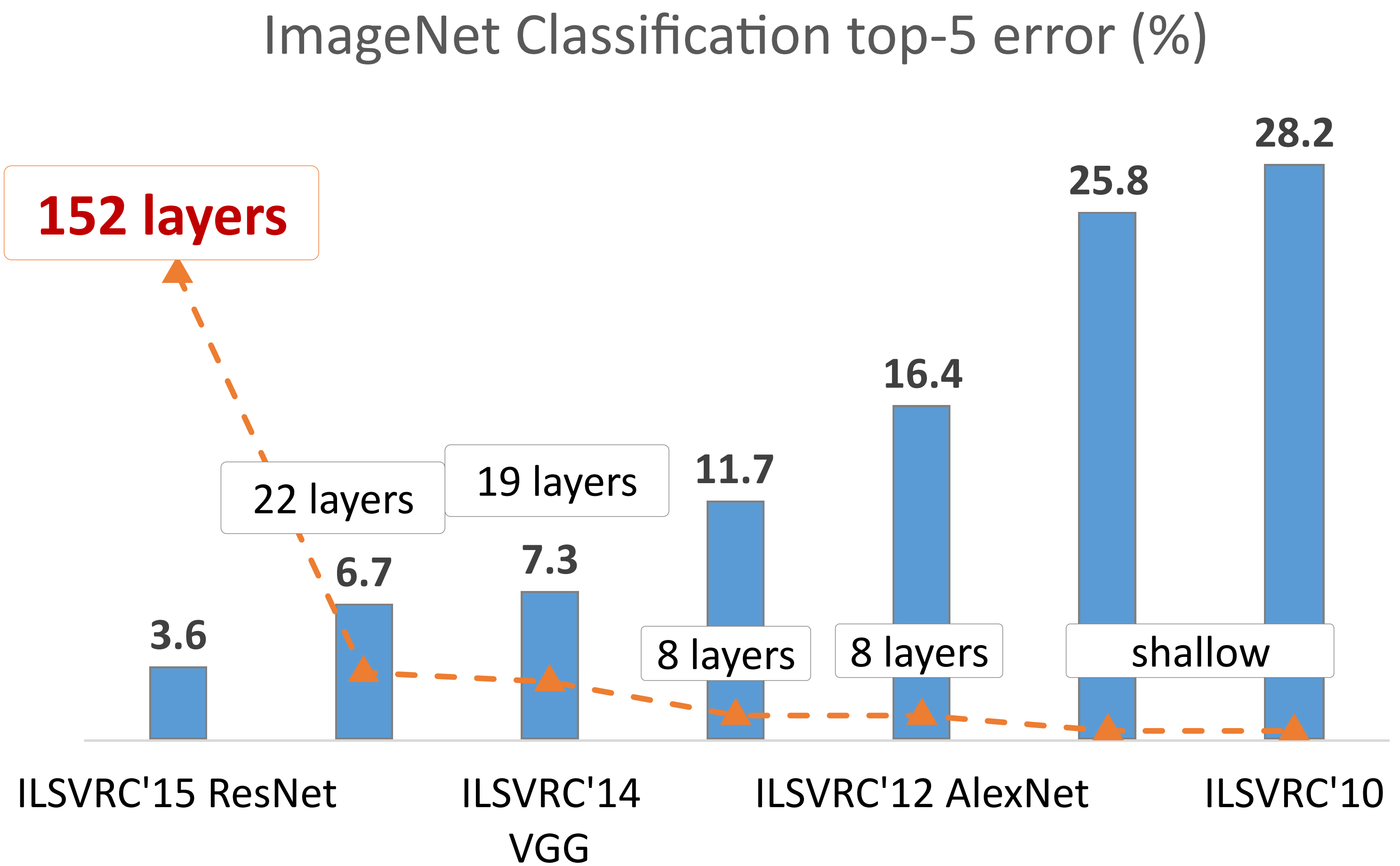
**1st places in all five main tracks**

- 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

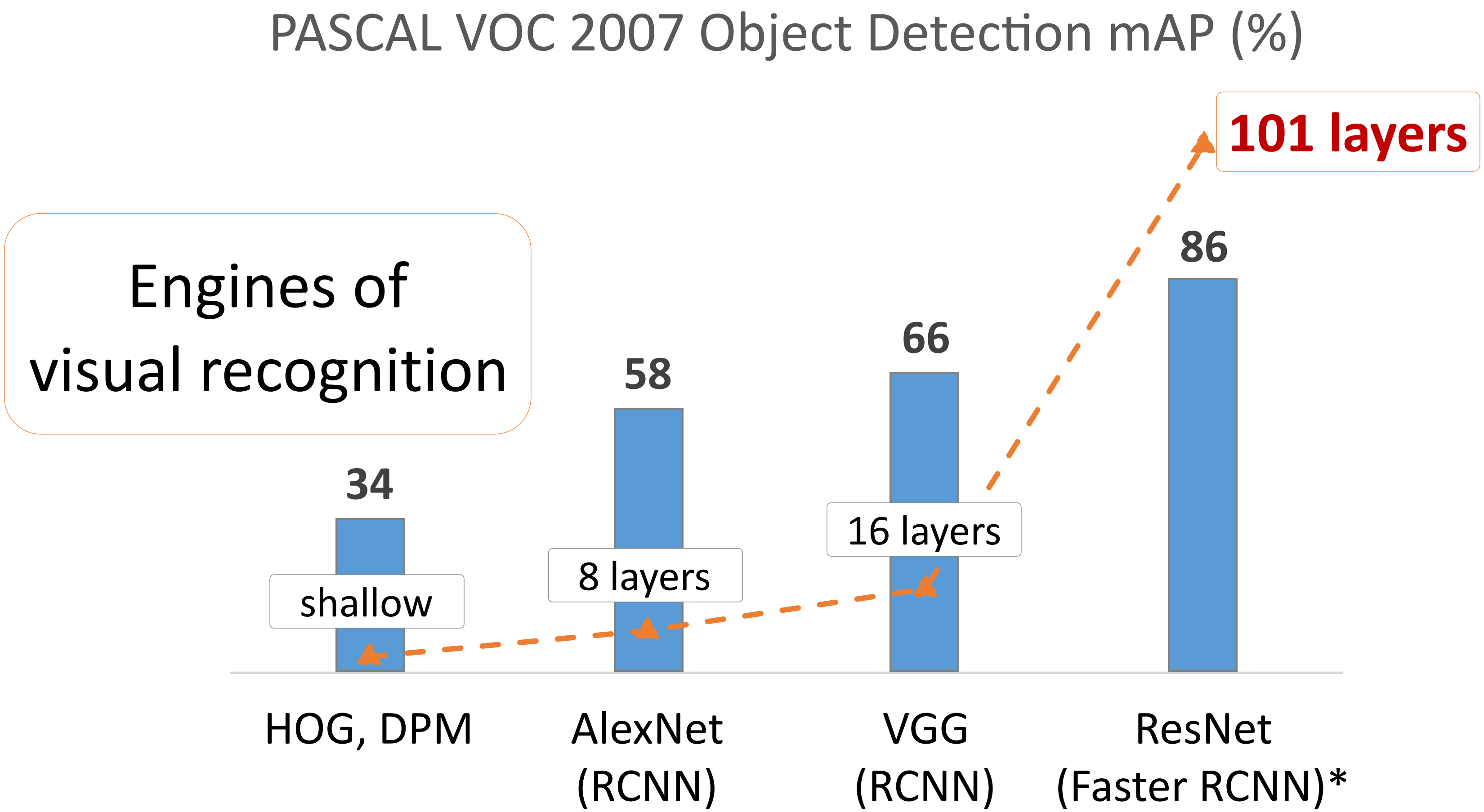
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

# Revolution of Depth




Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

# Revolution of Depth

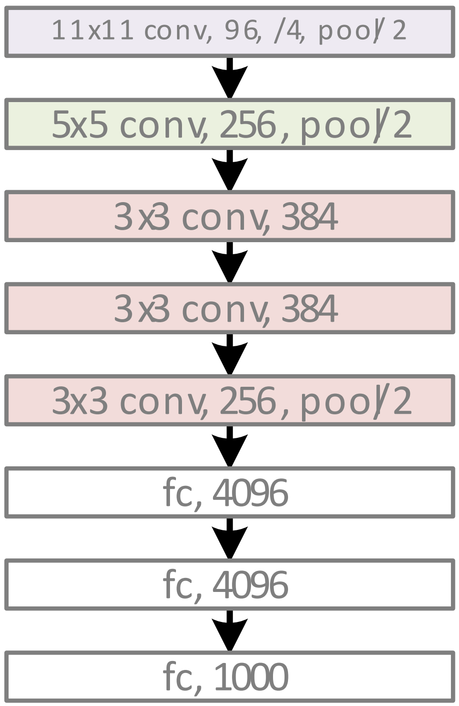



*w/ other improvements & more data

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

# Revolution of Depth

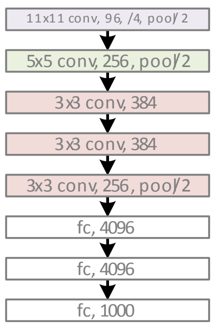
AlexNet, 8 layers (ILSVRC 2012)



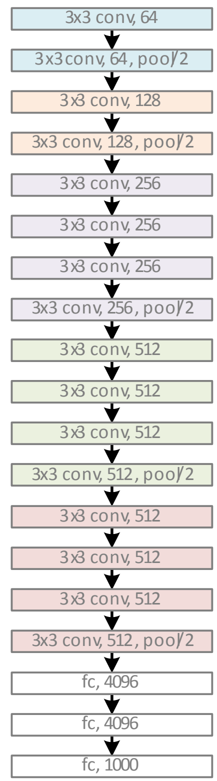

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

# Revolution of Depth

AlexNet, 8 layers (ILSVRC 2012)




VGG, 19 layers (ILSVRC 2014)

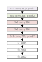



GoogleNet, 22 layers (ILSVRC 2014)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

# Revolution of Depth

AlexNet, 8 layers (ILSVRC 2012)



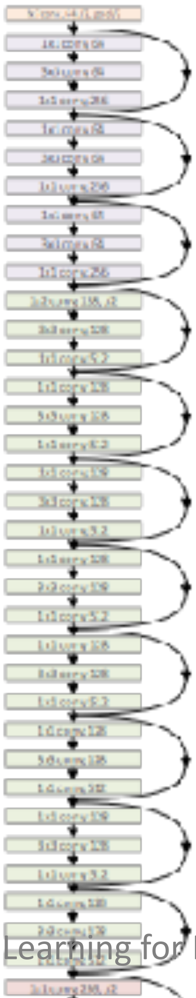
VGG, 19 layers (ILSVRC 2014)

ResNet, 152 layers (ILSVRC 2015)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

# Revolution of Depth

ResNet, 152 layers

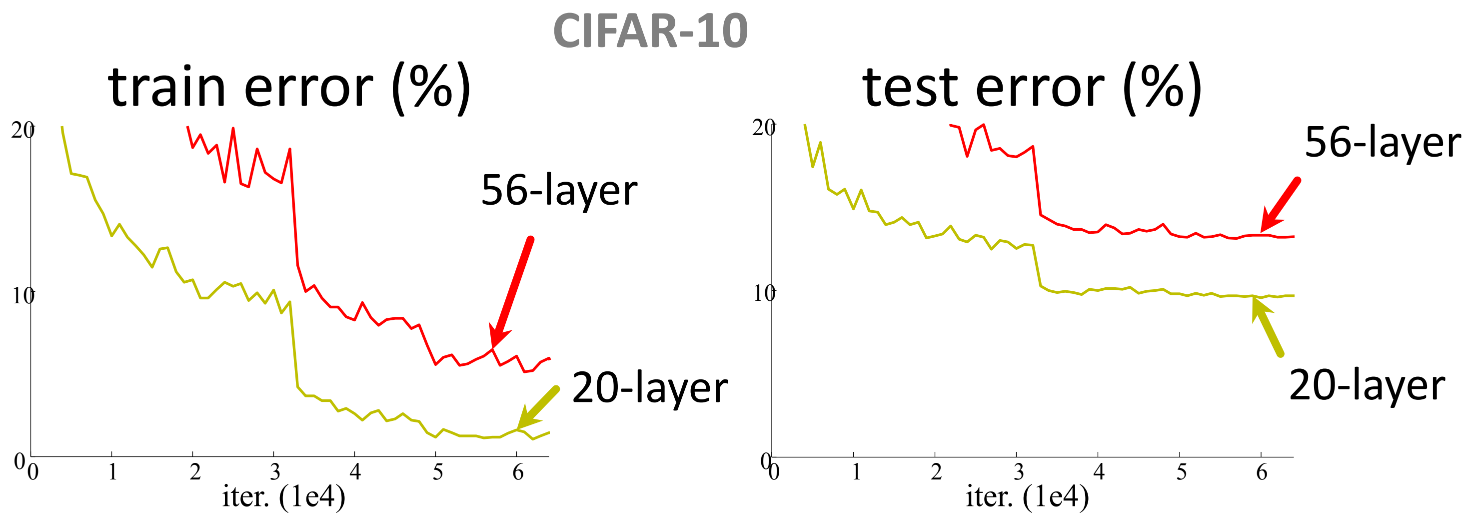


Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

# Is learning better networks as simple as stacking more layers?

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

# Simply stacking layers?

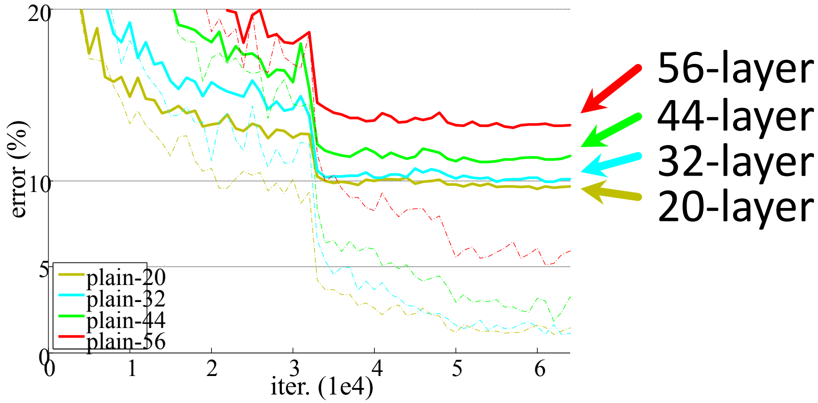



- *Plain* nets: stacking 3x3 conv layers...
- 56-layer net has **higher training error** and test error than 20-layer net

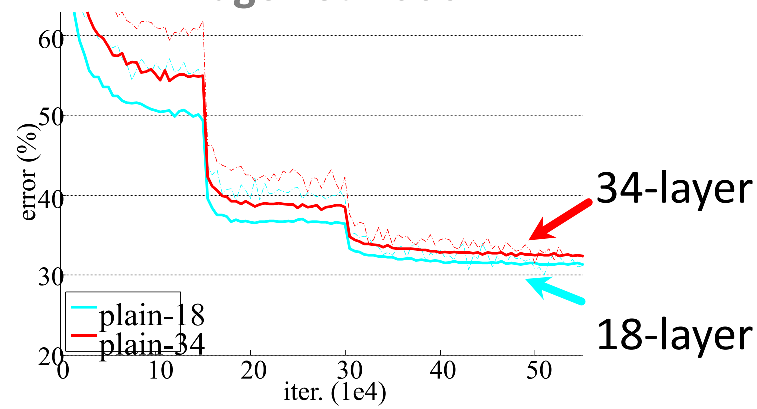
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

# Simply stacking layers?

CIFAR-10




ImageNet-1000



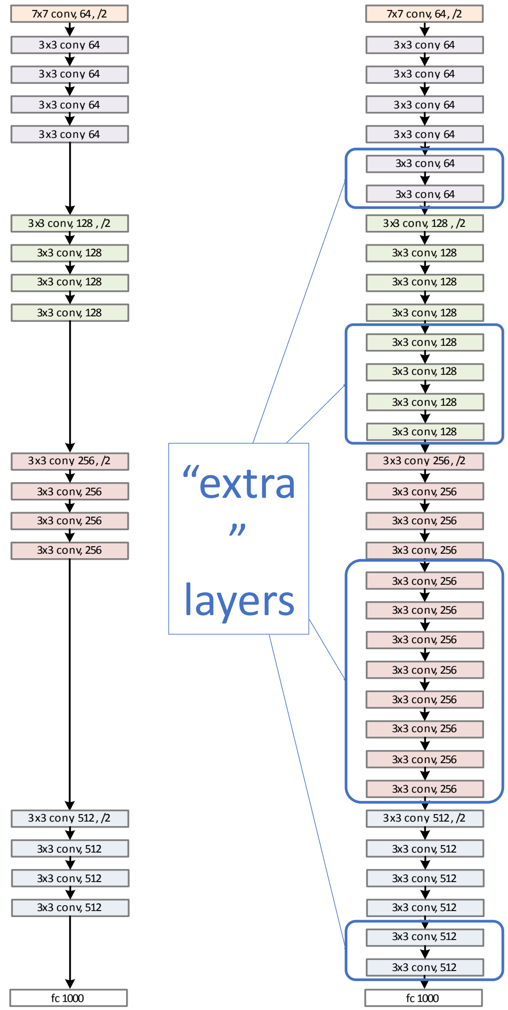

solid: test/val
dashed: train

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

a shallower model
(18 layers)

a deeper counterpart
(34 layers)

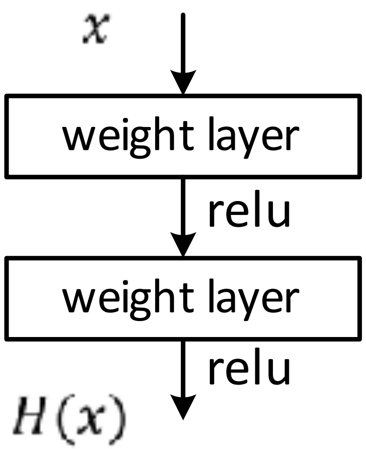



- Richer solution space
- A deeper model should not have **higher training error**
- A solution *by construction*:
  - original layers: copied from a learned shallower model
  - extra layers: set as identity
  - at least the same training error
- Optimization difficulties: solvers cannot find the solution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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

# Deep Residual Learning

- Plaint net




any two stacked layers

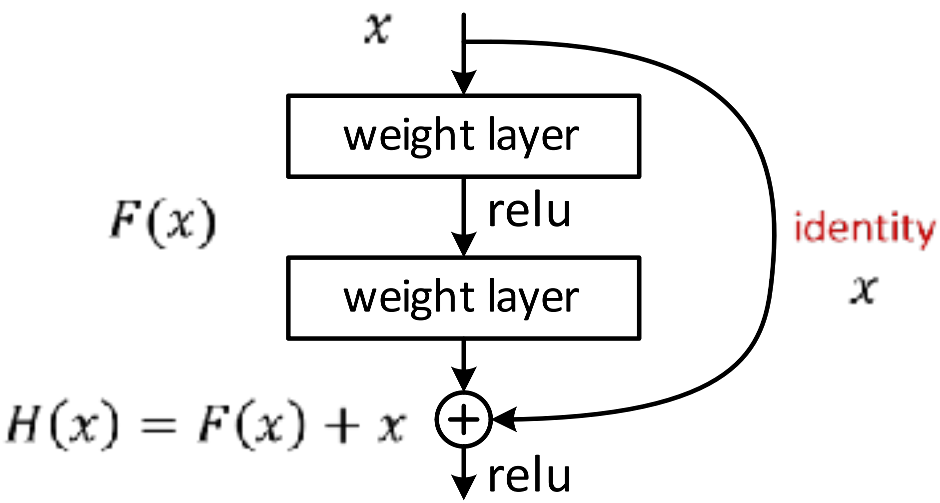
$H(x)$ is any desired mapping,

hope the 2 weight layers fit $H(x)$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

# Deep Residual Learning

- Residual net




$H(x)$ is any desired mapping,

~~hope the 2 weight layers fit $H(x)$~~

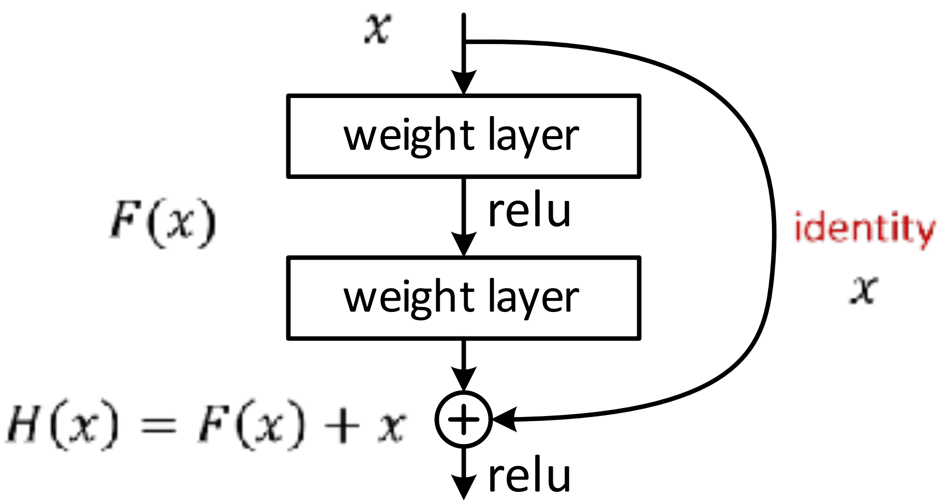
hope the 2 weight layers fit $F(x)$

let $H(x) = F(x) + x$

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

# 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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

# Network “Design”

- Keep it simple

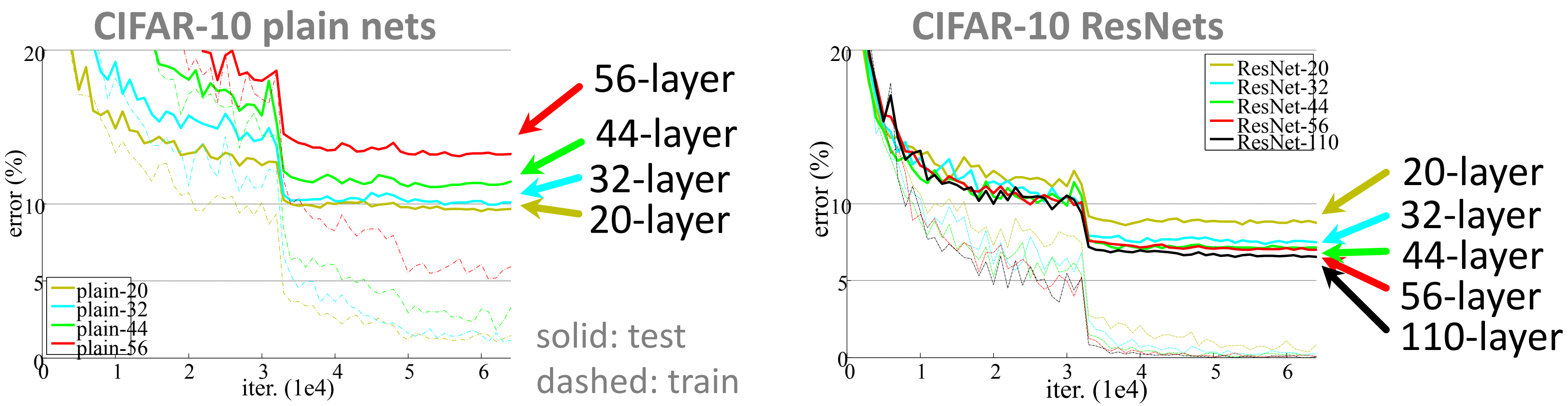
- Our basic design (VGG-style)
  - all 3x3 conv (almost)
  - spatial size /2 => # filters x2
  - Simple design; just deep!

plain net

ResNet

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

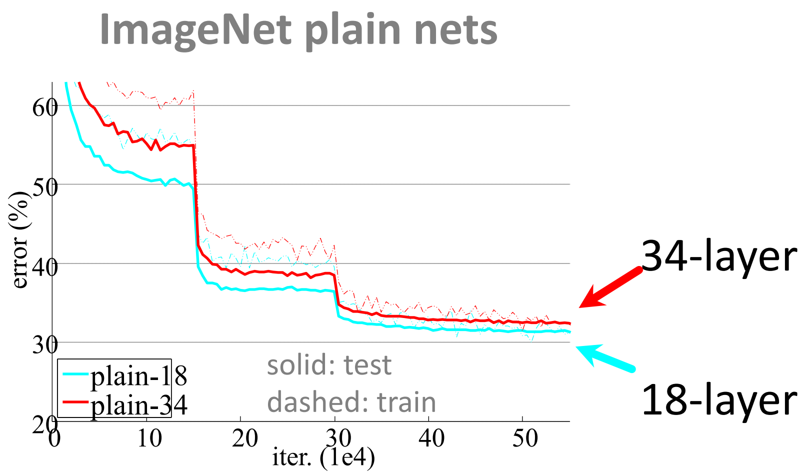
# CIFAR-10 experiments

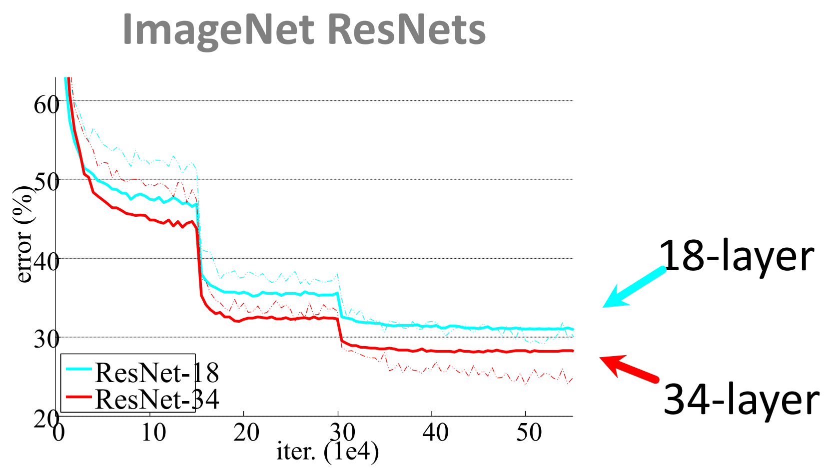



- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

# ImageNet experiments



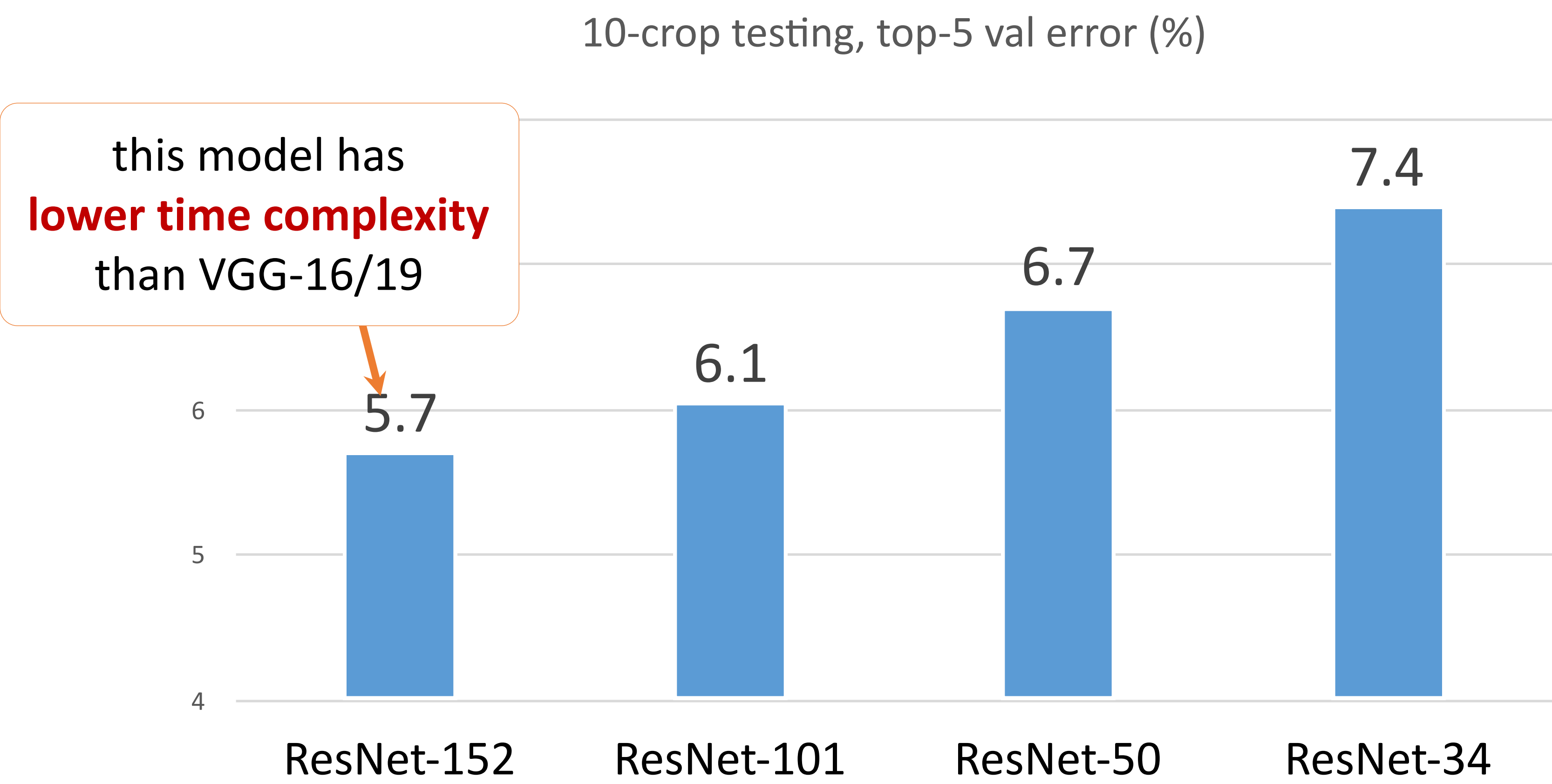




- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

# ImageNet experiments

- Deeper ResNets have lower error




Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

# "Features matter."

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

| task | 2nd-place winner | ResNets | margin (relative) |
|---|---|---|---|
| ImageNet Localization (top-5 error) | 12.0 | 9.0 | **27%** |
| ImageNet Detection (mAP@.5) | 53.6 | 62.1 | **16%** |
| COCO Detection (mAP@.5:.95) | 33.5 | 37.3 | **11%** |
| COCO Segmentation (mAP@.5:.95) | 25.1 | 28.2 | **12%** |

**absolute 8.5% better!**

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

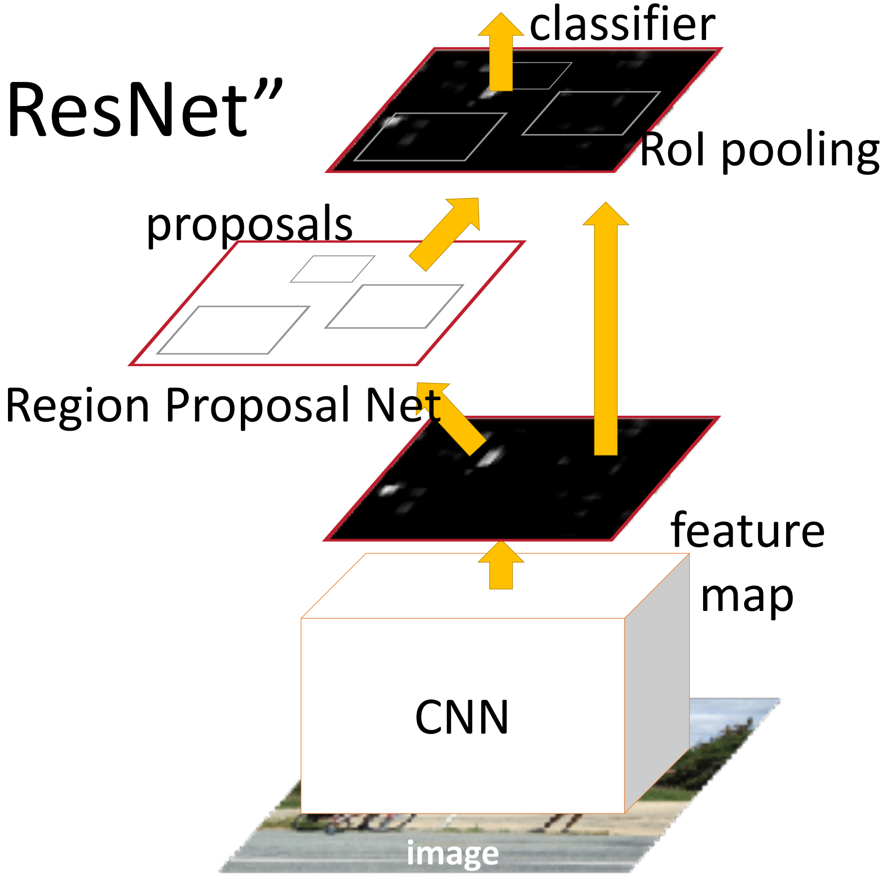
# Object Detection (brief)

- Simply "Faster R-CNN + ResNet"

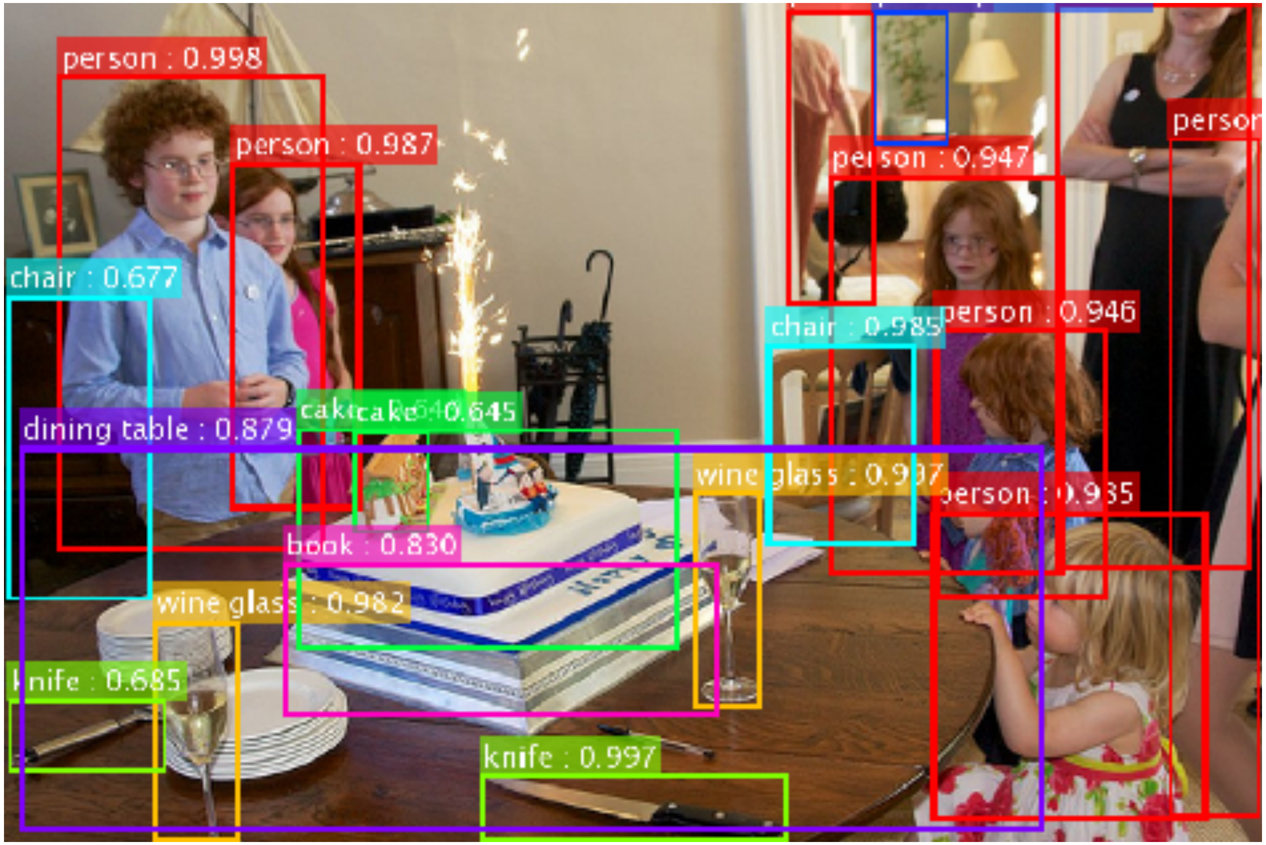
| Faster R-CNN baseline | mAP@.5 | mAP@.5:.95 |
|---|---|---|
| VGG-16 | 41.5 | 21.5 |
| ResNet-101 | **48.4** | **27.2** |

COCO detection results

(ResNet has 28% relative gain)




Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.
Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". NIPS 2015.




# Our results on MS COCO

*the original image is from the COCO dataset

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.
Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". NIPS 2015.

this video is available online: https://youtu.be/WZmSMkK9VuA


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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.
Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". NIPS 2015.

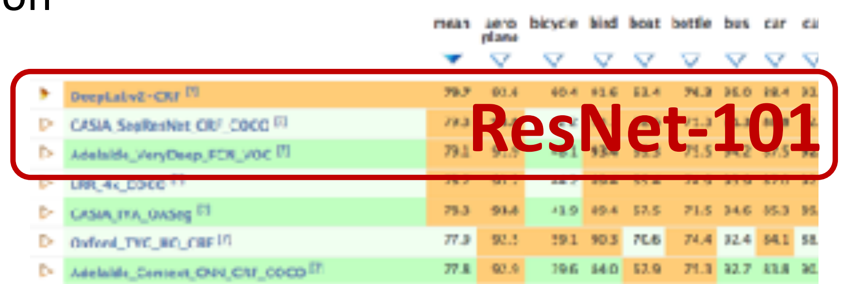
# More Visual Recognition Tasks

ResNets lead on these benchmarks (incomplete list):

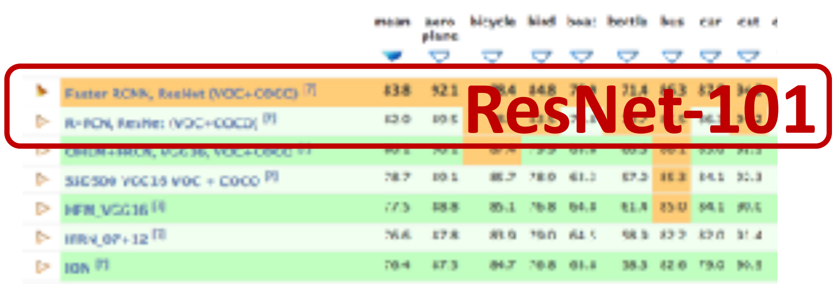
- **ImageNet** classification, detection, localization
- **MS COCO** detection, segmentation

- **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]
- …




PASCAL **segmentation** leaderboard




PASCAL **detection** leaderboard

# 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)

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

# 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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Identity Mappings in Deep Residual Networks”. arXiv 2016.
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.

# Resources

# Thank You!

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

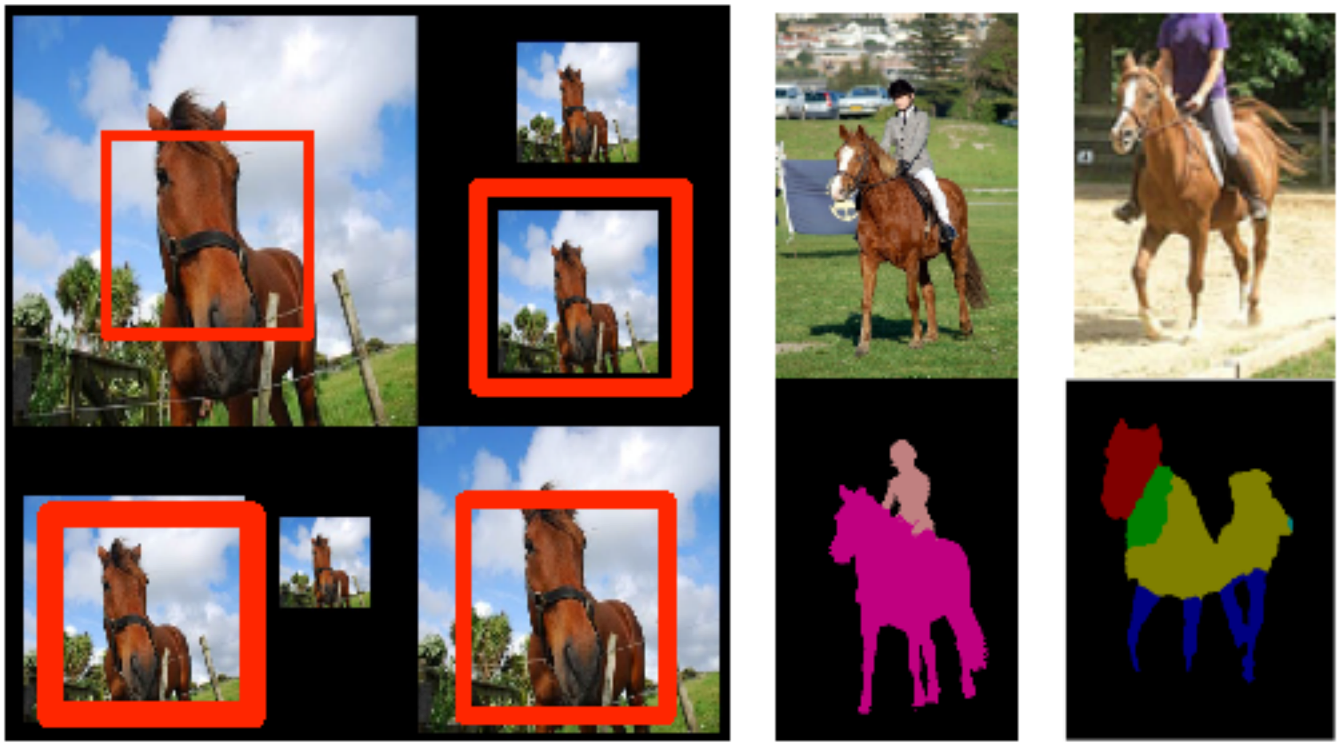
- Many available implementations:
  (list in https://github.com/KaimingHe/deep-residual-networks)
  - Facebook AI Research's Torch ResNet: https://github.com/facebook/fb.resnet.torch
  - Torch, CIFAR-10, with ResNet-20 to ResNet-110, training code, and curves: code
  - Lasagne, CIFAR-10, with ResNet-32 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
  - …….

# 2. 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

# Fully Convolutional Neural Networks for Classification, Detection & Segmentation

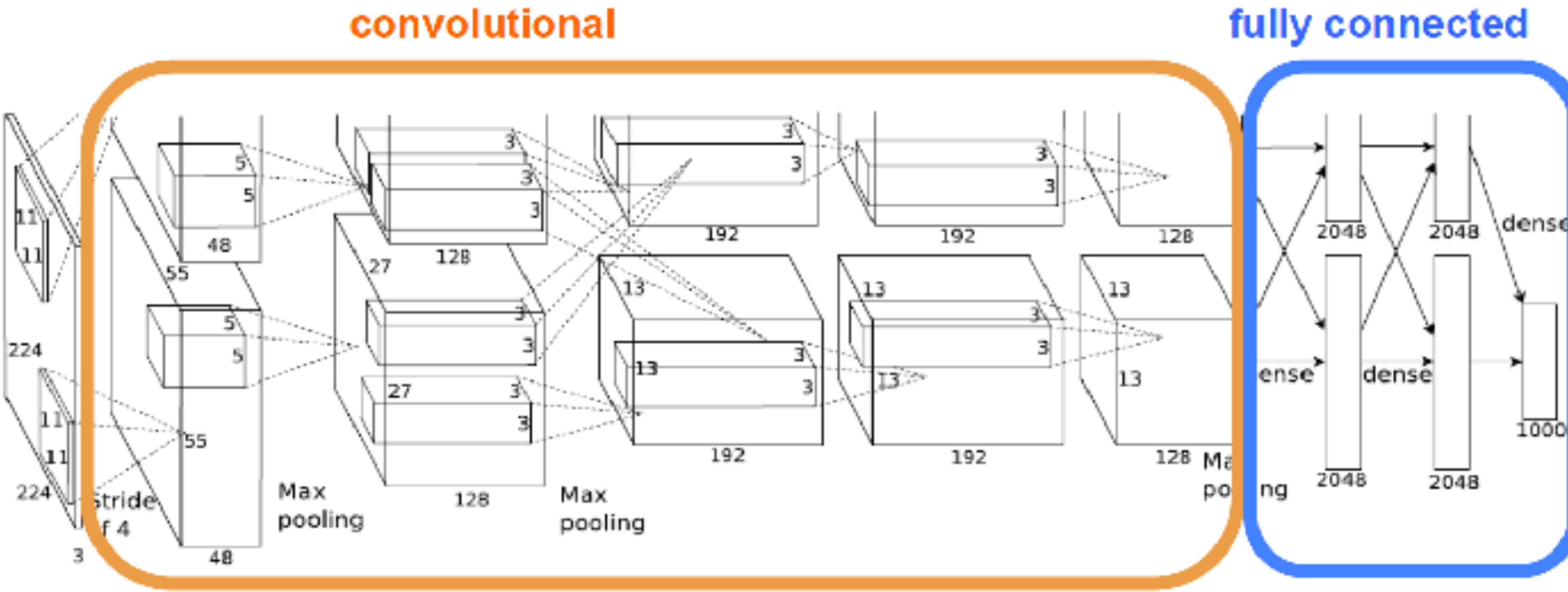


*or, all your computer wanted to know about horses*

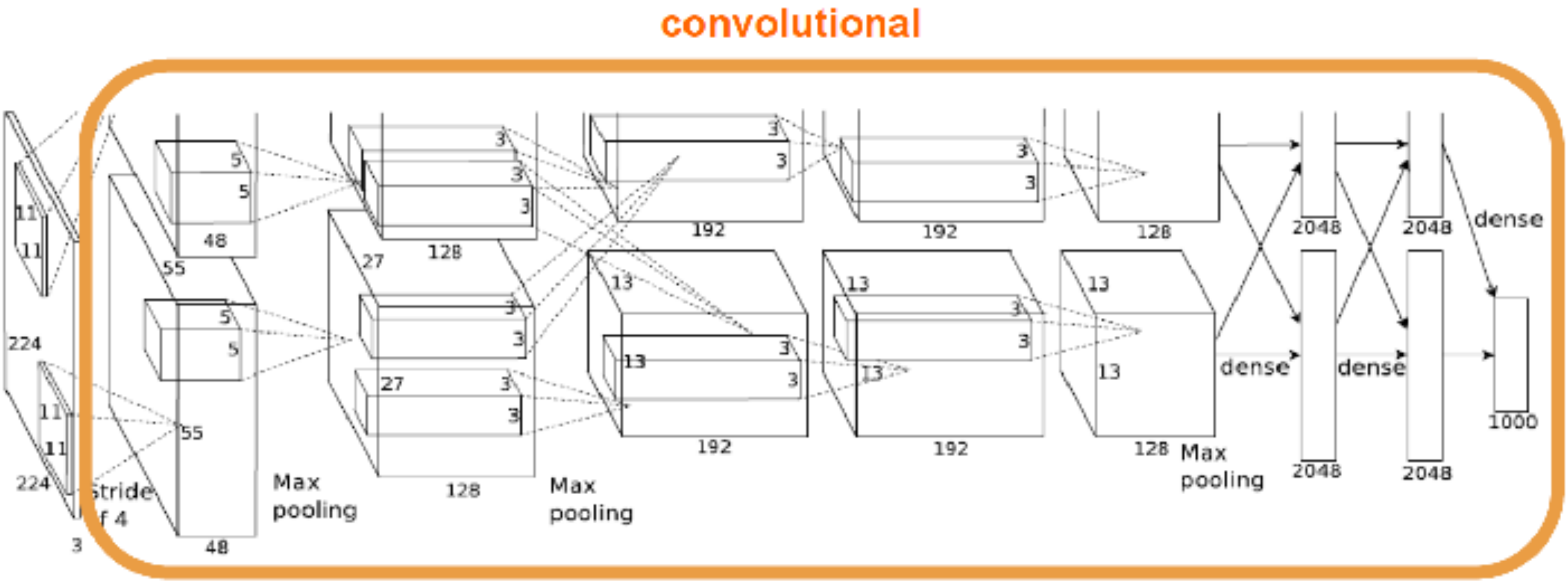
Iasonas Kokkinos
Ecole Centrale Paris / INRIA Saclay

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

# Fully convolutional neural networks



# Fully convolutional neural networks

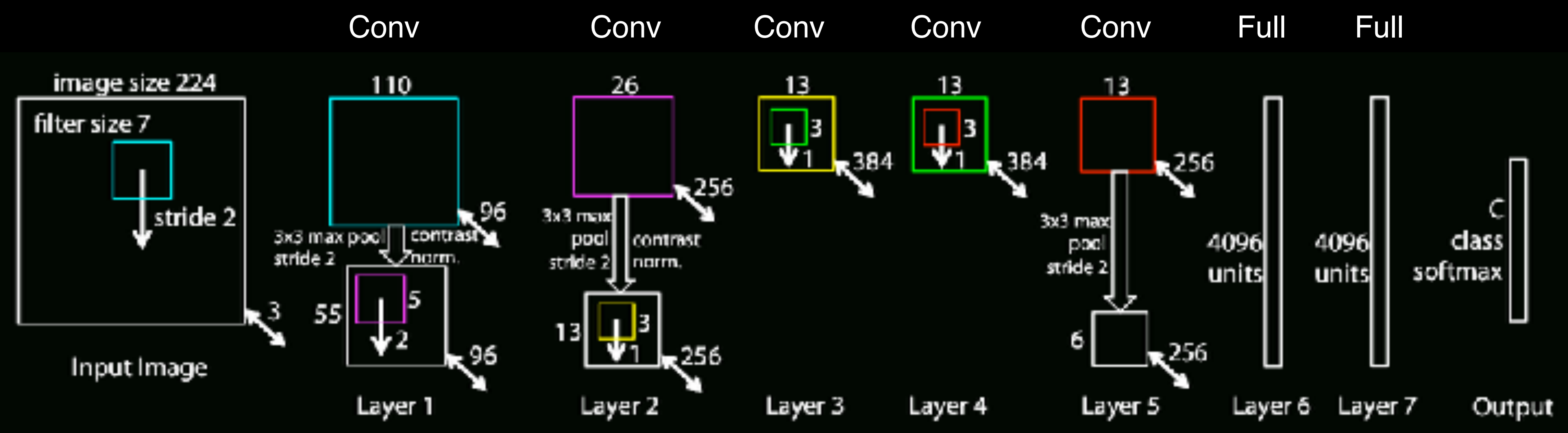



Fully connected layers: 1x1 spatial convolution kernels

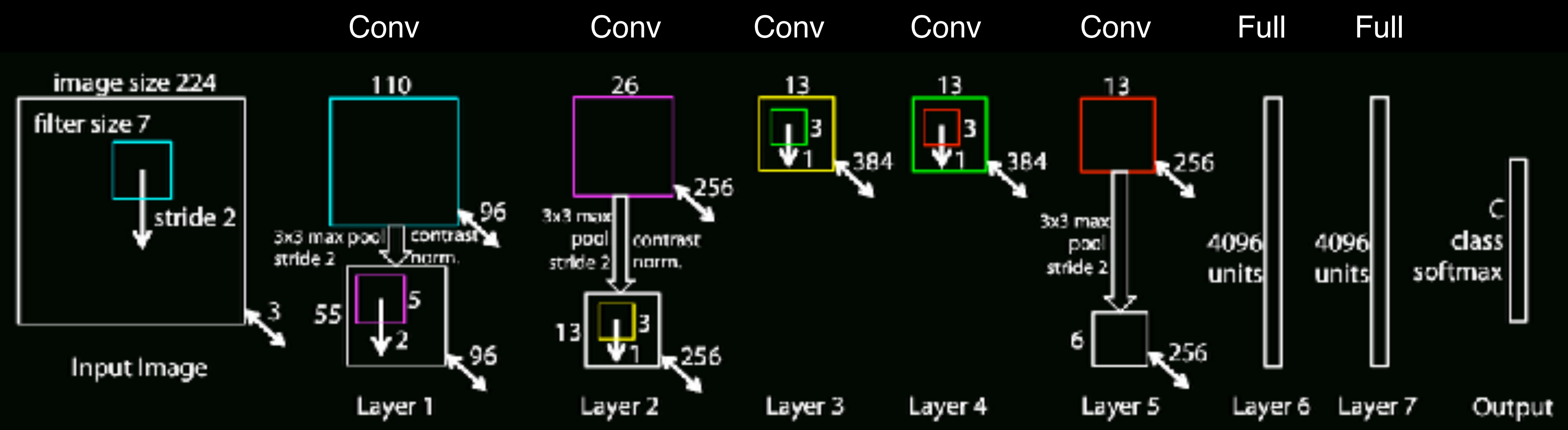
Allows network to process images of arbitrary size


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




# 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

G. Papandreou

L-C. Chen, UCLA

K. Murphy, Google

A. Yuille, UCLA

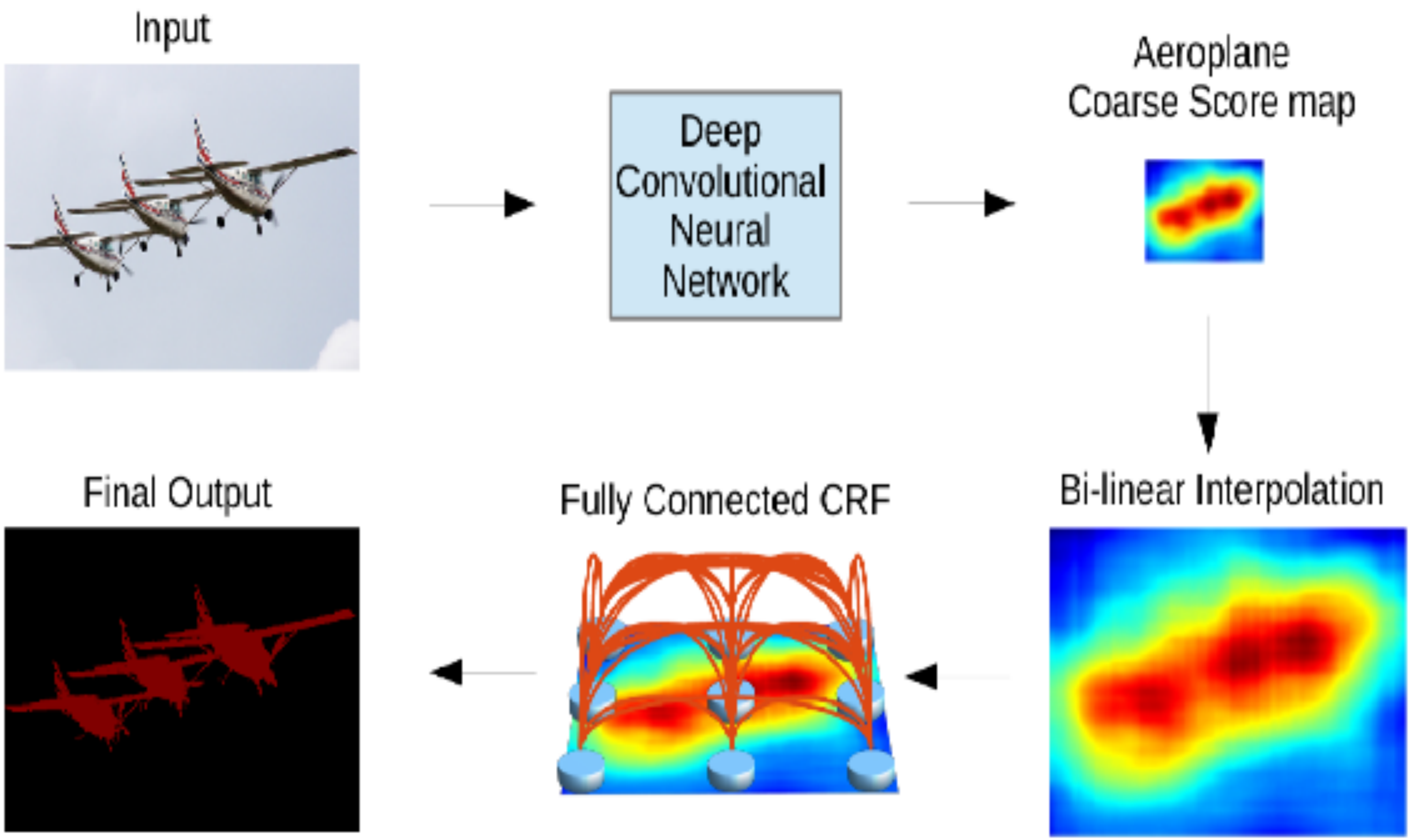
**L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy and A. Yuille**
**Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs, http://arxiv.org/abs/1412.7062**

# Semantic segmentation task

PASCAL2
Pattern Analysis, Statistical Modelling and Computational Learning

Microsoft COCO
Common Objects in Context

# System outline

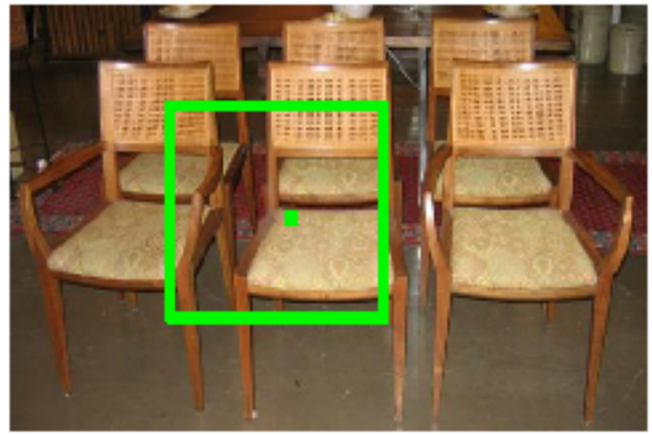



J. Long, E. Shelhamer, T. Darrell, FCNNs for Semantic Segmentation, CVPR 15

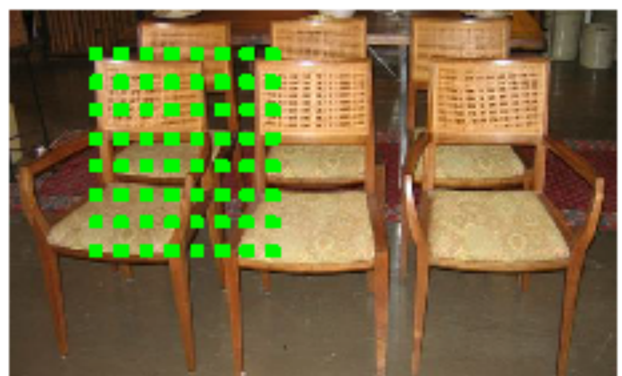
P. Krähenbühl and V. Koltun, Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials, NIPS 2011

# Repurposing DCNNs for semantic segmentation

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



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

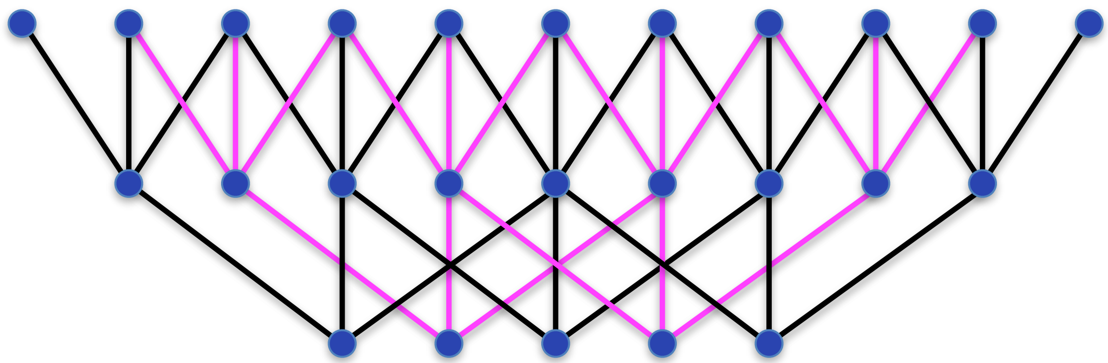


**8 FPS**

M. Holschneider, et al, A real-time algorithm for signal analysis with the help of the wavelet transform, *Wavelets, Time-Frequency Methods and Phase Space*, 1989.

# "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)

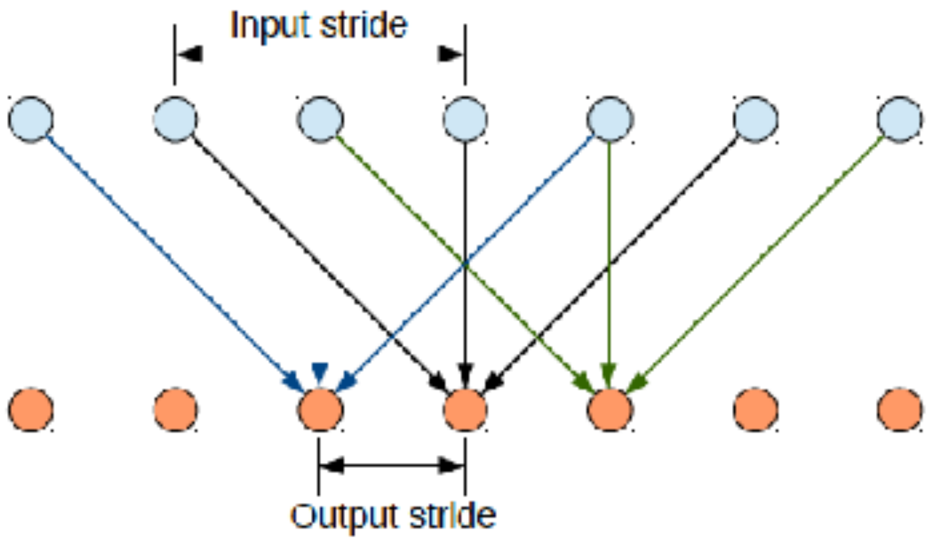
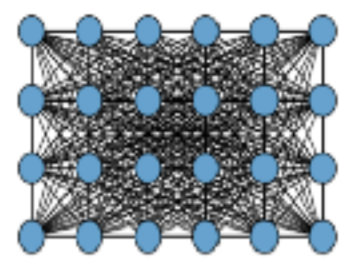



Figure 1: Illustration of the hole algorithm in 1-D, when *kernel_size* = 3, *input_stride* = 2, and *output_stride* = 1.

# FCNN-DCRF: Full & densely connected

FCNN-based labelling

from densely-connected CRF



- 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

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

  - with unary terms obtained from the CNN:

$$\theta_i(x_i) = -\log P(x_i)$$

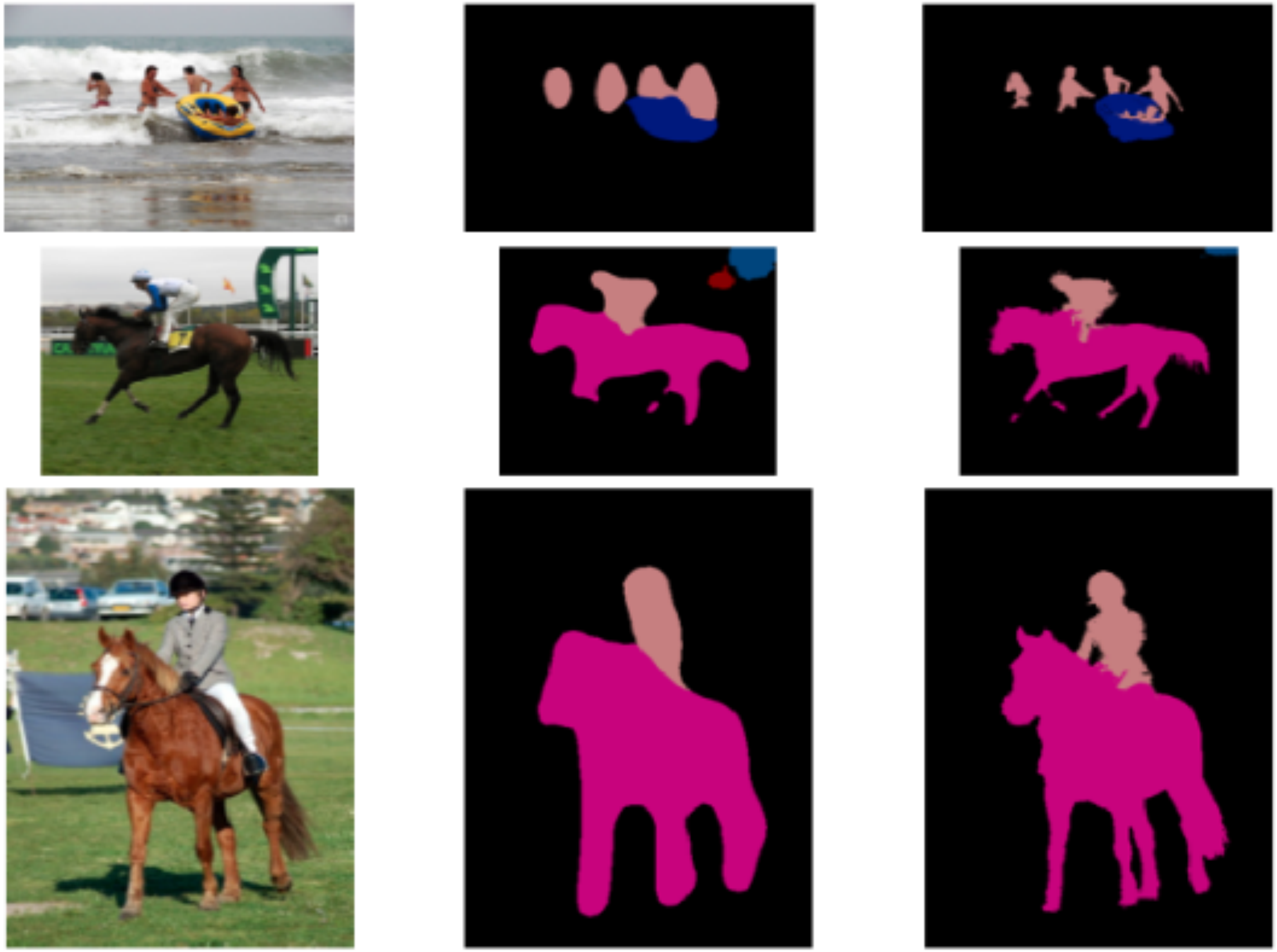
  - and pairwise terms (Potts model)

$$\theta_{ij}(x_i, x_j) = \mu(x_i, x_j) \sum_{m=1}^{K} w_m \cdot k^m(\boldsymbol{f}_i, \boldsymbol{f}_j)$$

    - with $\mu(x_i, x_j) = 1 \text{ if } x_i \neq x_j$

$$\sum_{m=1}^{K} w_m \cdot k^m(\boldsymbol{f}_i, \boldsymbol{f}_j) = w_1 \exp\Big(-\frac{||p_i - p_j||^2}{2\sigma_\alpha^2} - \frac{||I_i - I_j||^2}{2\sigma_\beta^2}\Big) + w_2 \exp\Big(-\frac{||p_i - p_j||^2}{2\sigma_\gamma^2}\Big)$$

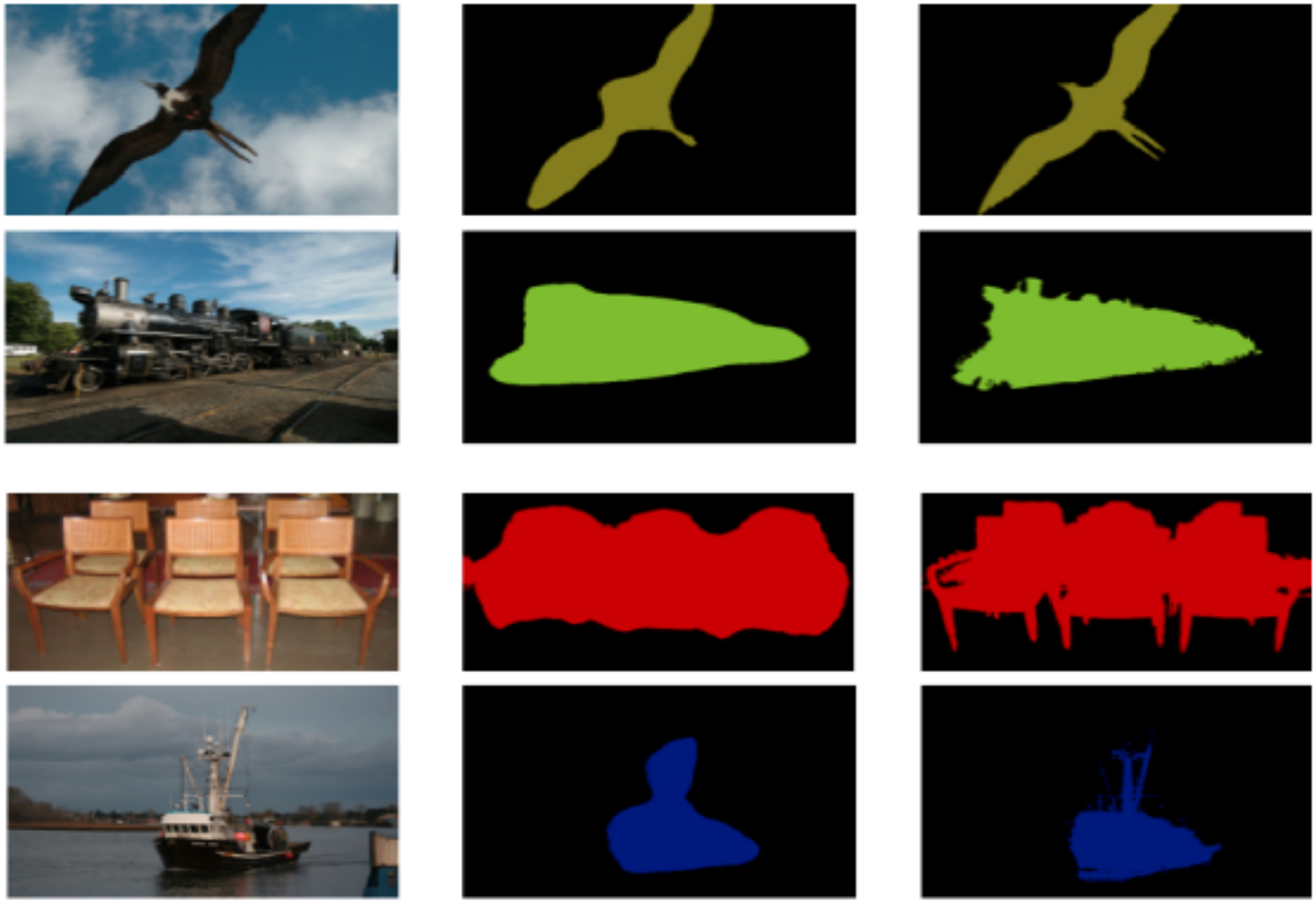
# Indicative Results



Raw score maps

After dense CRF

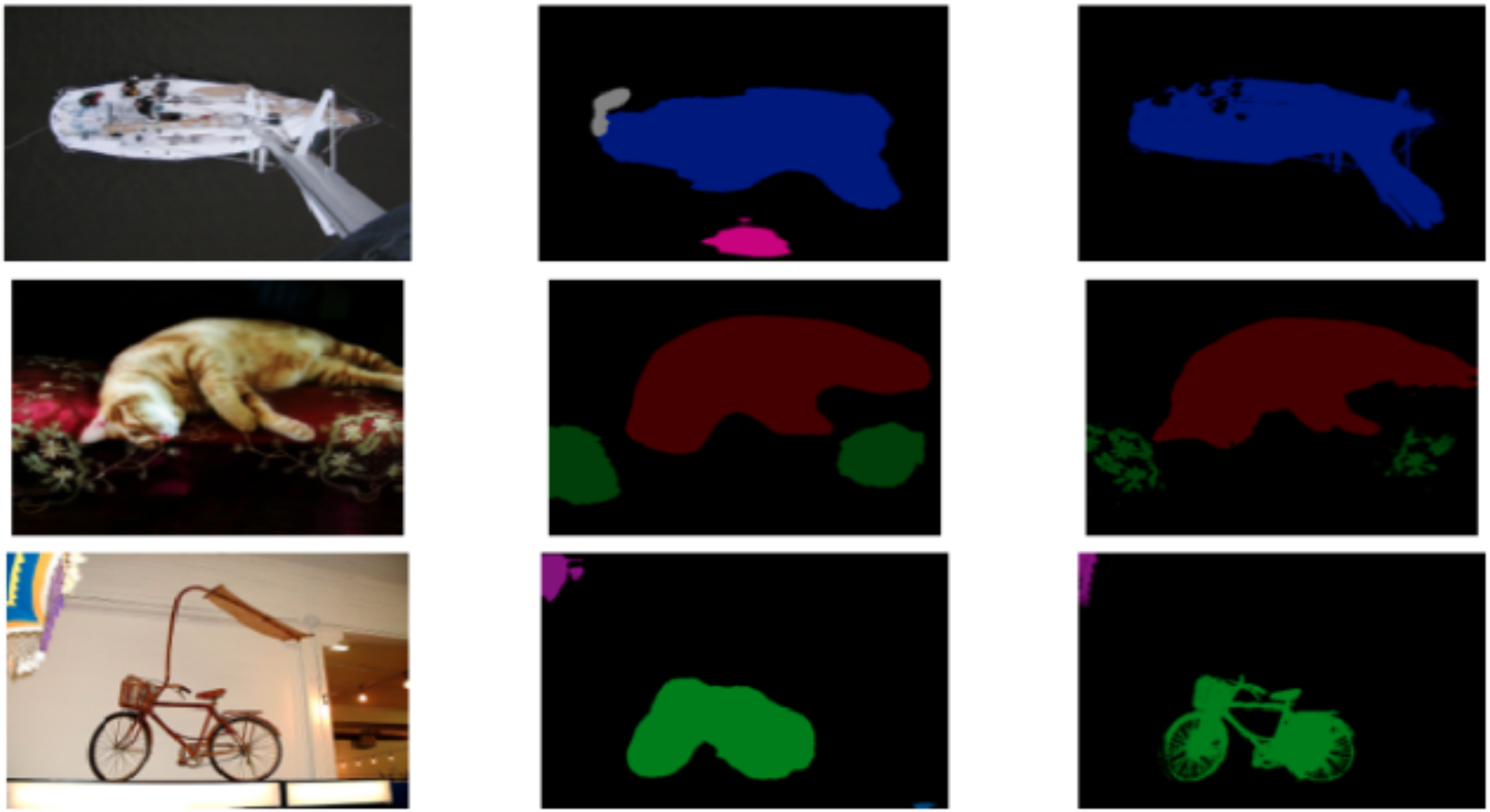
# Indicative Results



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

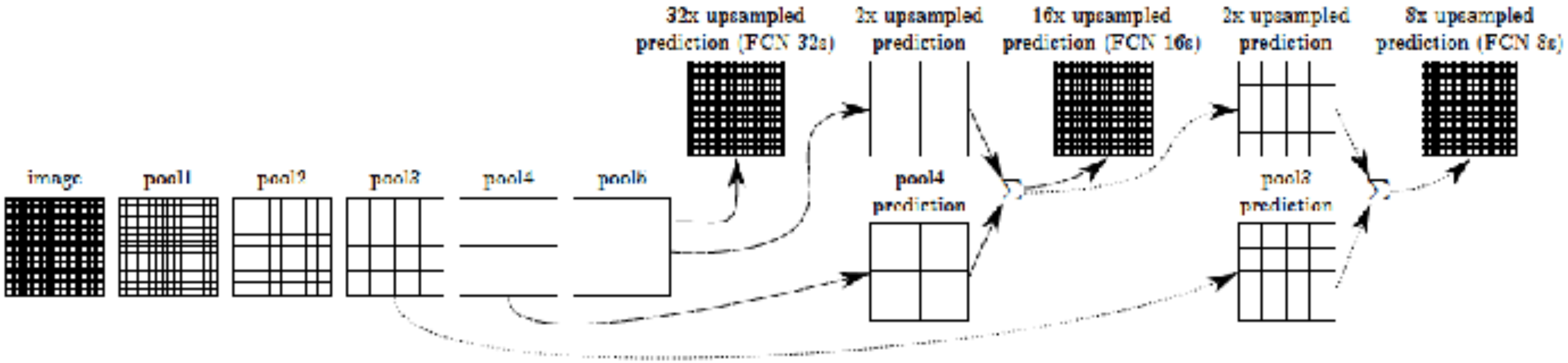
| Method | mean IOU (%) |
|---|---|
| DeepLab | 59.80 |
| DeepLab-CRF | 63.74 |
| DeepLab-MSc | 61.30 |
| DeepLab-MSc-CRF | 65.21 |

Improvements due to Dense CRF

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

Our work (FCNN unaries)
61.3 -> 65.21 (+3.9)

# Another fully convolutional network for semantic segmentation (without CRF)




J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *arXiv:1411.4038*, 2014.

# Comparisons to Fully Convolutional Net

Ground-truth

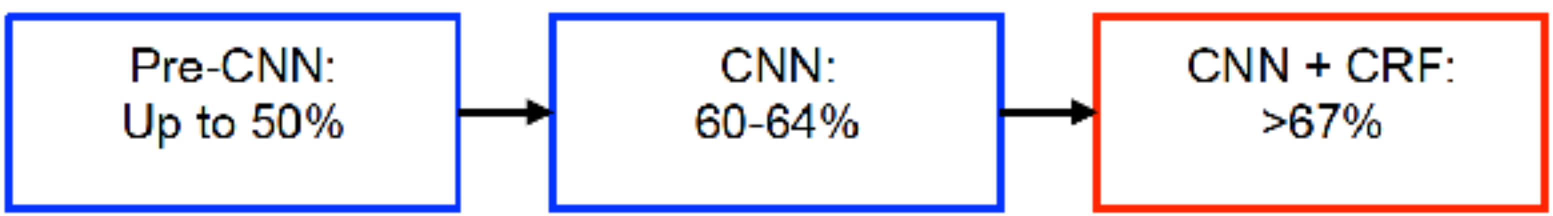
FCN-8

Our work

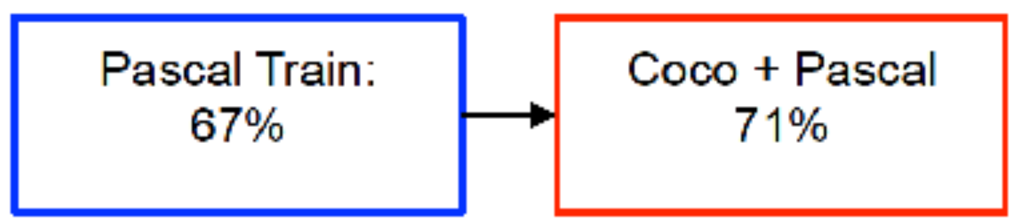
J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. *arXiv:1411.4038*, 2014.

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

| Method | mean IOU (%) |
| --- | --- |
| MSRA-CFM | 61.8 |
| FCN-8s | 62.2 |
| TTI-Zoomout-16 | 64.4 |
| DeepLab-CRF (our) | 66.4 |
| DeepLab-MSc-CRF (our) | 67.1 |




G. Papandreou, et al, Weakly- and Semi-Supervised Learning of a DCNN for Semantic Image Segmentation, arxiv 2015



# 3. Cityscapes Dataset

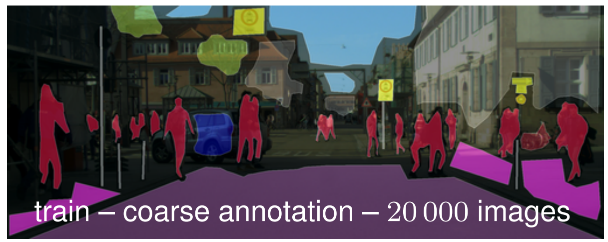
- Dataset for semantic labeling and "understanding"
  - Cordts, Omaran, Ramos, Rehfeld, Enzweiler, Benenson, Franke, Roth, Schiele @ cvpr16
  - https://www.cityscapes-dataset.net
  - http://arxiv.org/abs/1604.01685

# The Cityscapes Dataset

## for Semantic Scene Labeling and Understanding

CITYSCAPES DATASET

https://www.cityscapes-dataset.net




Marius Cordts[1,3] Mohamed Omran[2] Sebastian Ramos[1]

Timo Rehfeld[1,3] Markus Enzweiler[1] Rodrigo Benenson[2]

Uwe Franke[1] Stefan Roth[3] Bernt Schiele[2]

1 DAIMLER

2 max planck institut informatik

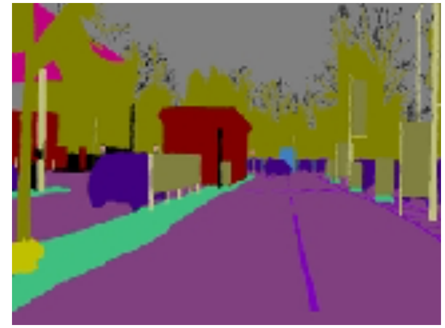
3 TECHNISCHE UNIVERSITÄT DARMSTADT

# Previous Work

CITYSCAPES DATASET

**KITTI** [Geiger et al. '12]

- stereo video
- no official semantic labeling or instance labeling challen



**CamVid** [Brostow et al., to appear]

- monocular video

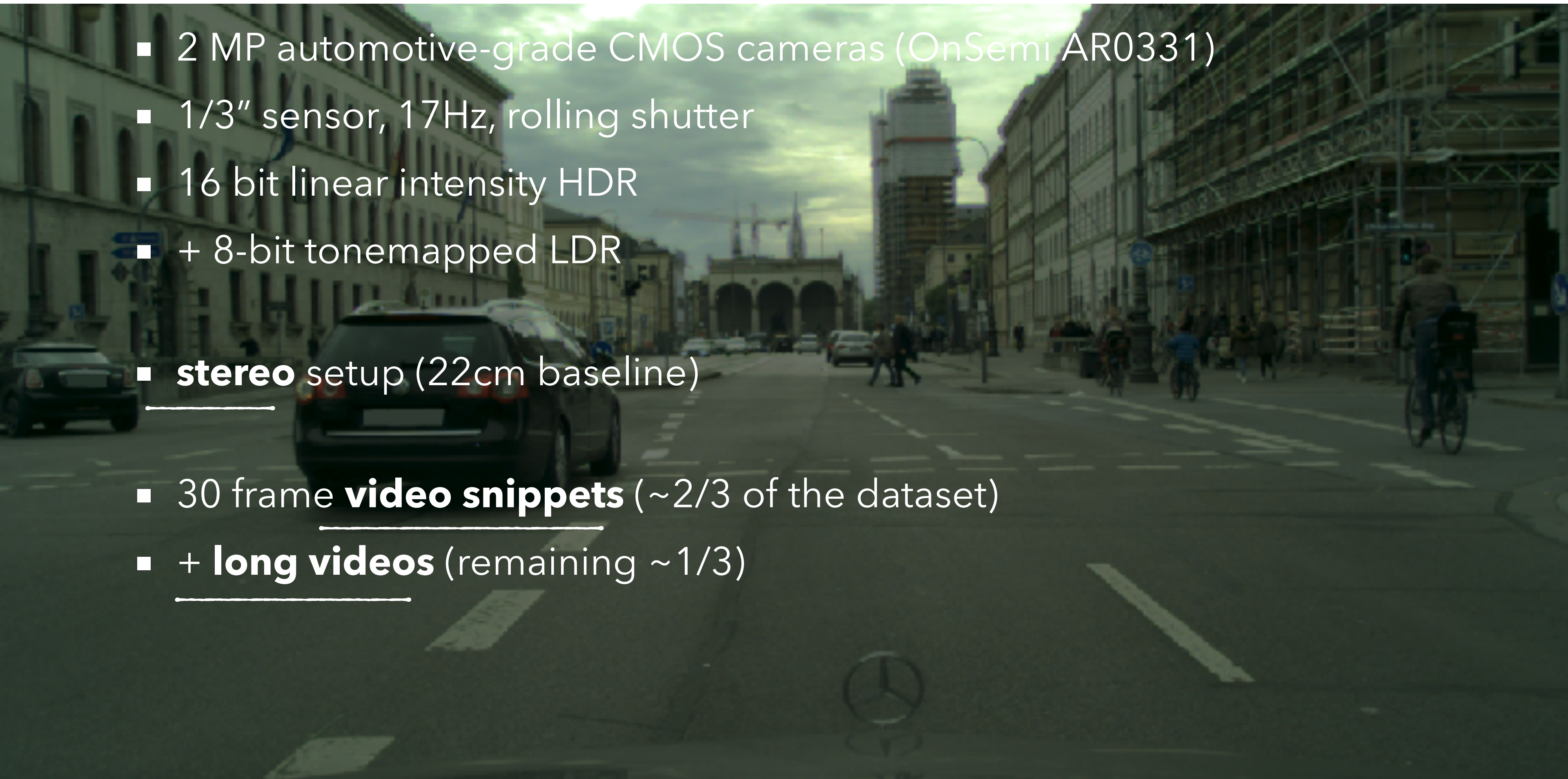

**Daimler Urban Scenes** [Scharwächter et al. '14]

- stereo video
- limited number of classes / annotation density

# Overview

CITYSCAPES DATASET

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

CITYSCAPES DATASET

https://www.cityscapes-dataset.net

# Overview

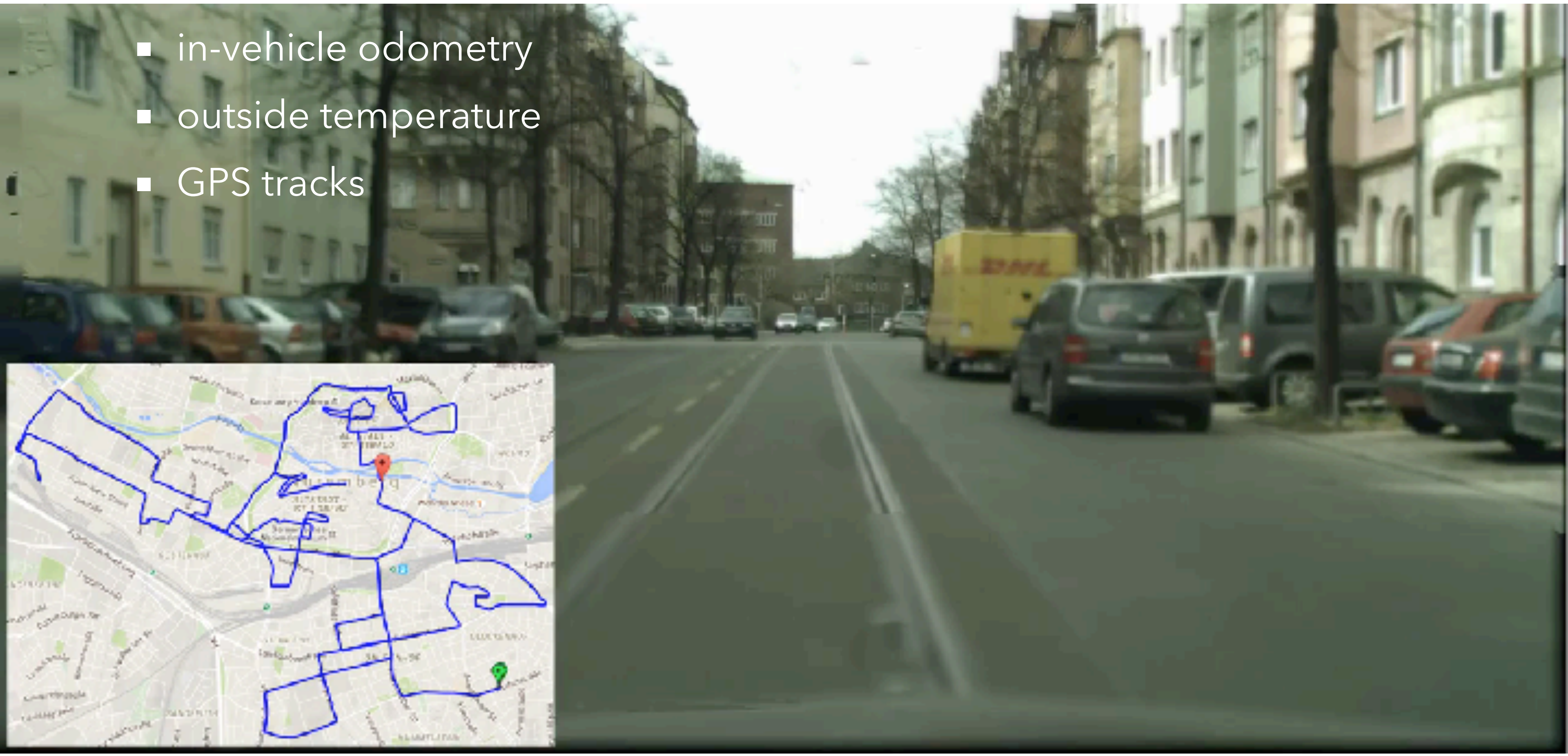
https://www.cityscapes-dataset.net

- precomputed disparity

# Overview

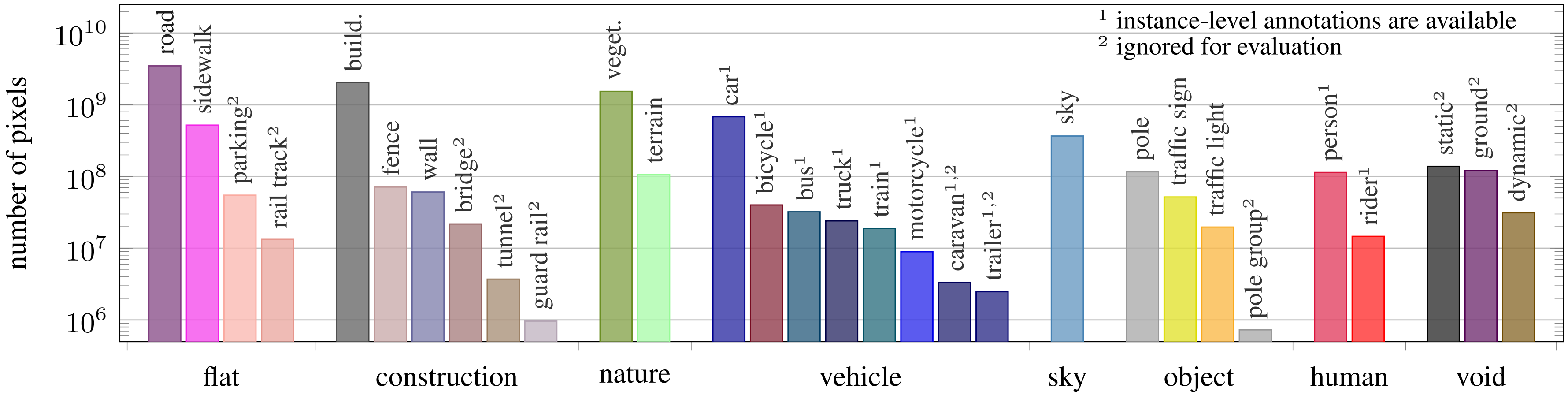
https://www.cityscapes-dataset.net

- 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

CITYSCAPES DATASET

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

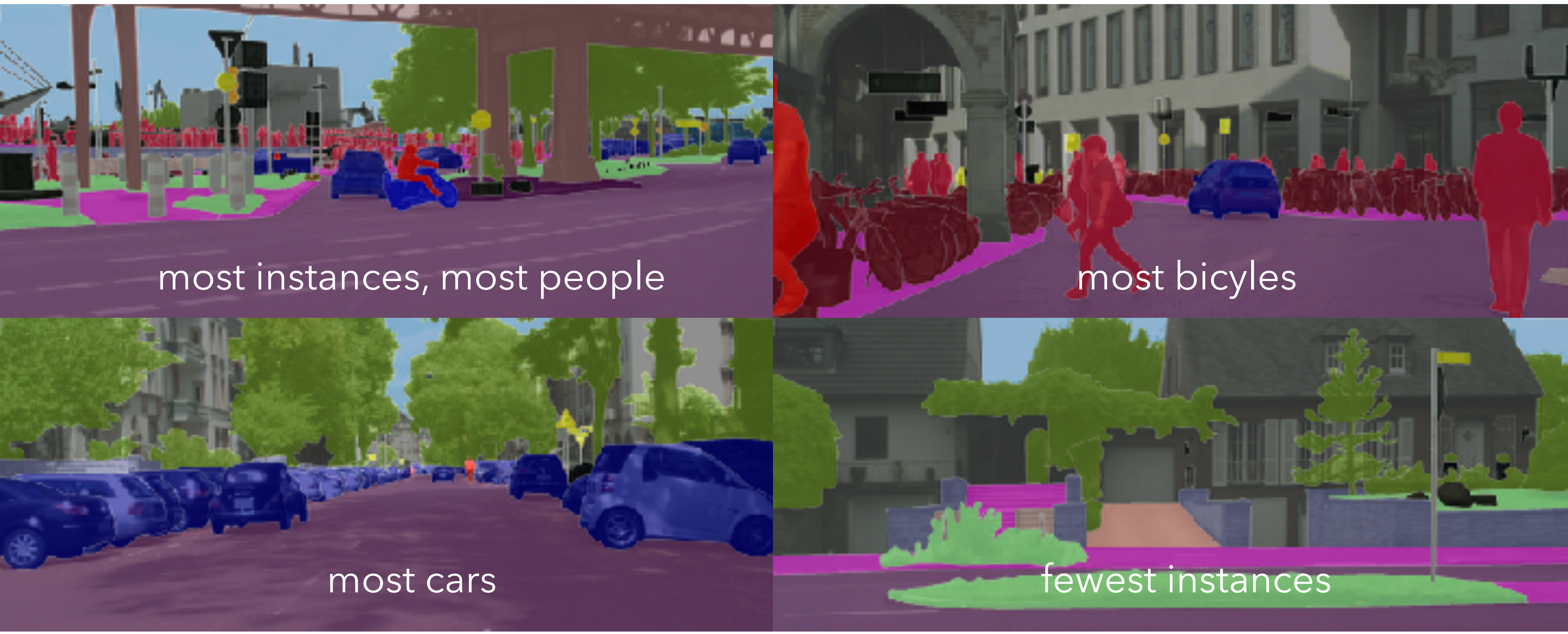
https://www.cityscapes-dataset.net

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

# Objective: Complexity

https://www.cityscapes-dataset.net

- **Complex, real-world scenes**



# Objective: Diversity

CITYSCAPES DATASET

- **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

CITYSCAPES DATASET

| | #pixels [$10^9$] | annot. density [%] |
|---|---|---|
| Ours (fine) | 9.41 | **97.0** |
| Ours (coarse) | **26.0** | 67.5 |
| CamVid | 0.62 | 96.2 |
| DUS | 0.14 | 63.0 |
| KITTI | 0.23 | 88.9 |

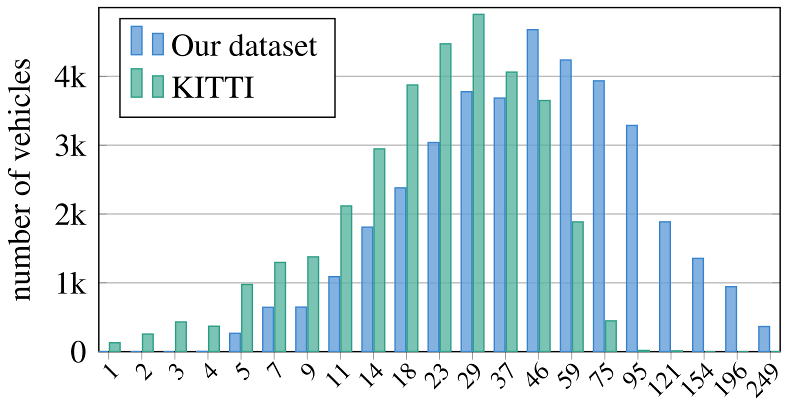
dataset size & density

| | #humans [$10^3$] | #vehicles [$10^3$] | #h/image | #v/image |
|---|---|---|---|---|
| Ours (fine) | 24.2 | **49.1** | **7.0** | **14.1** |
| KITTI | 6.1 | 30.3 | 0.8 | 4.1 |
| Caltech | **192**[1] | - | 1.5 | - |

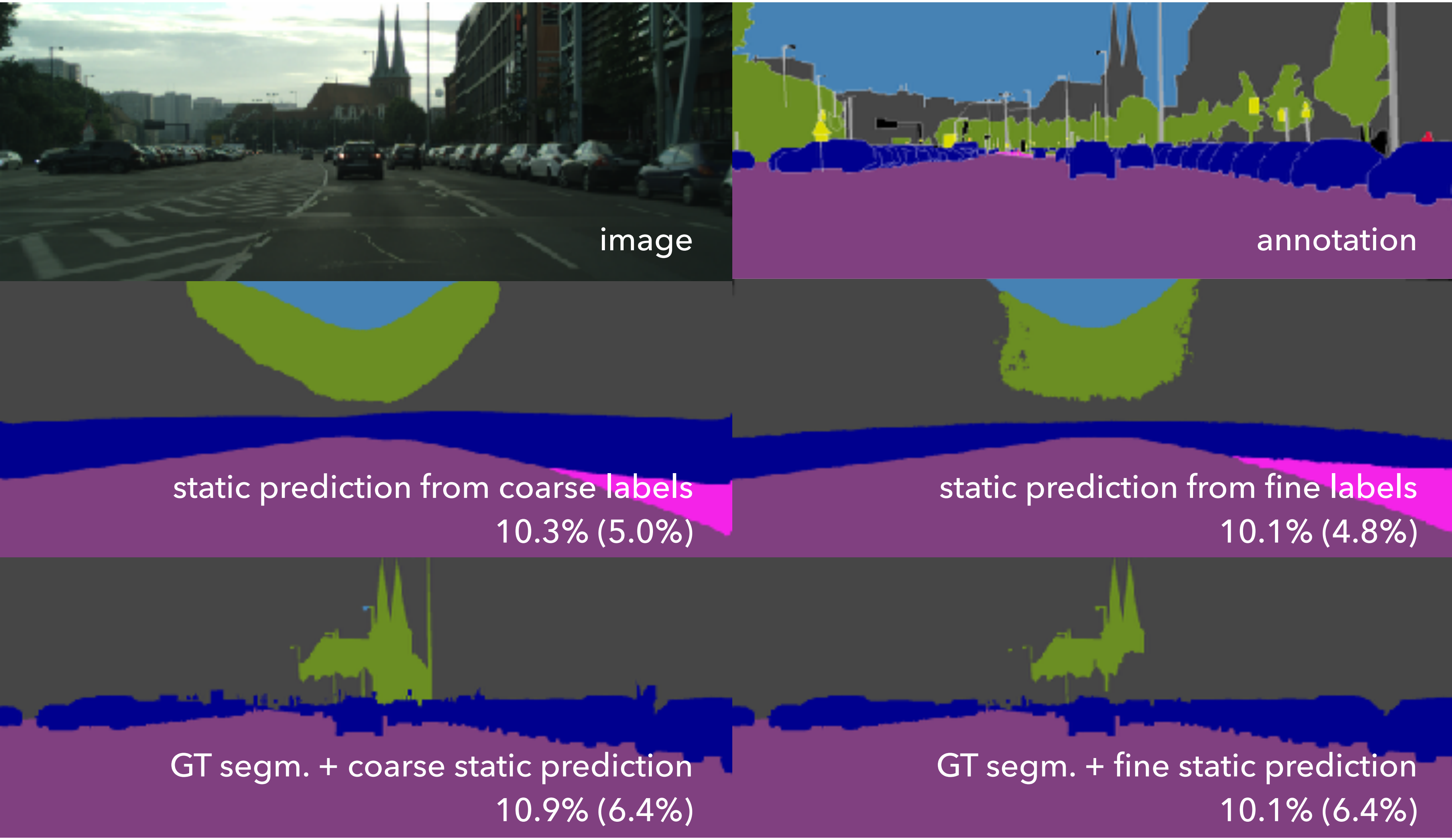
instance statistics

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

histogram of vehicle distances

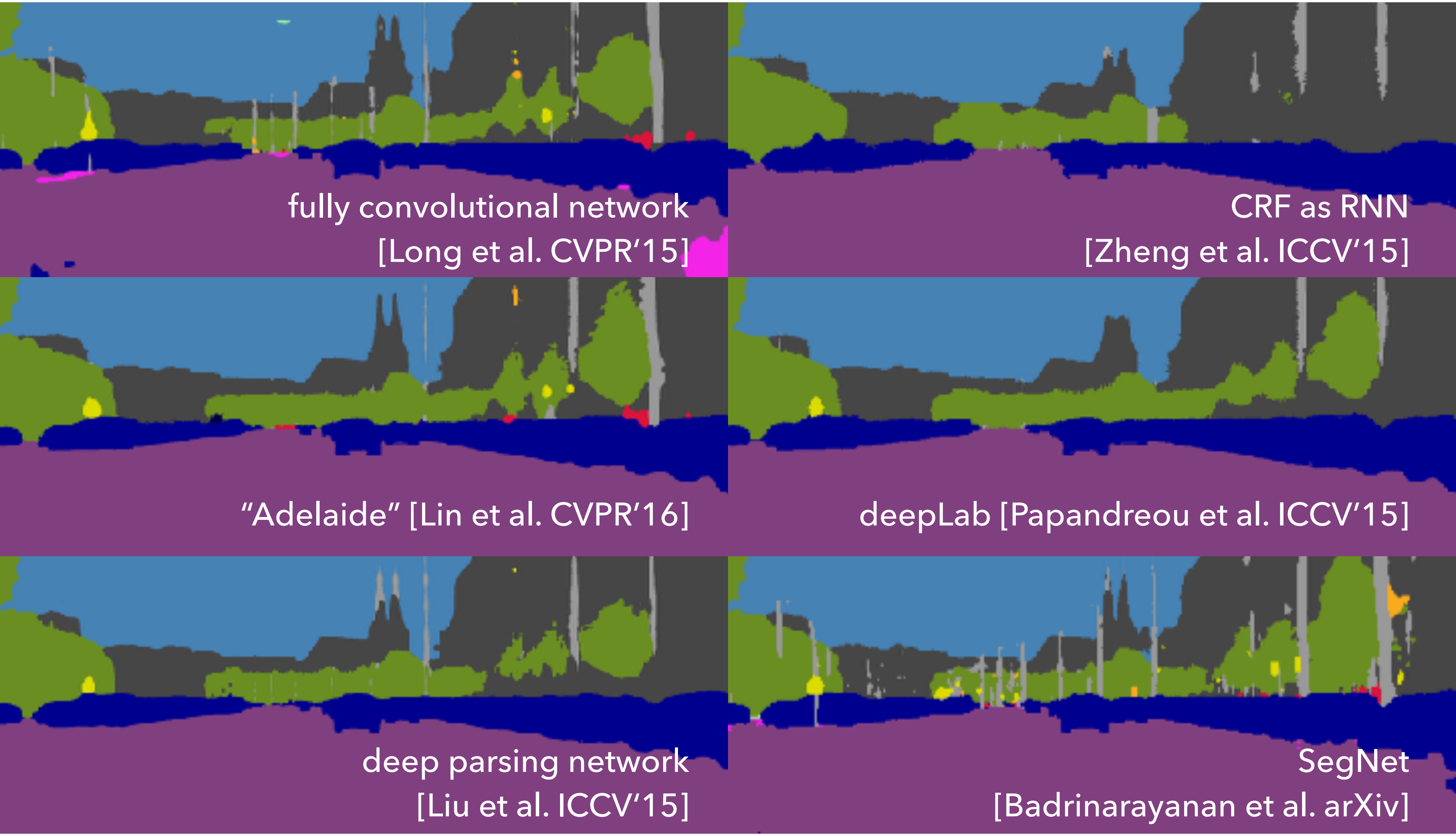


# Control experiments




IoU (iIoU)

# 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

CITYSCAPES DATASET

https://www.cityscapes-dataset.net

| | train | val | coarse | sub | Classes IoU | Classes iIoU | Categories IoU | Categories iIoU | |
|---|---|---|---|---|---|---|---|---|---|
| FCN-8s | ✓ | ✓ | | | $\boxed{65.3}$ | 41.7 | $\boxed{85.7}$ | 70.1 | ~3,500 **finely** annotated images |
| FCN-8s | ✓ | ✓ | | 2 | 61.9 | 33.6 | 81.6 | 60.9 | |
| FCN-8s | | ✓ | | | $\boxed{58.3}$ | 37.4 | $\boxed{83.4}$ | 67.2 | 500 **finely** annotated images |
| FCN-8s | | | ✓ | | $\boxed{58.0}$ | 31.8 | $\boxed{78.2}$ | 58.4 | 20,000 **coarsely** annotated images |
| [4] extended | ✓ | | | 4 | 56.1 | 34.2 | 79.8 | 66.4 | Badrinarayanan et al. @ arXiv'15 |
| [4] basic | ✓ | | | 4 | 57.0 | 32.0 | 79.1 | 61.9 | Badrinarayanan et al. @ arXiv'15 |
| [40] | ✓ | ✓ | ✓ | 3 | 59.1 | 28.1 | 79.5 | 57.9 | Liu et al @ ICCV'15 |
| [81] | ✓ | | | 2 | 62.5 | 34.4 | 82.7 | 66.0 | Zheng et al @ ICCV'15 |
| [9] | ✓ | ✓ | | 2 | 63.1 | 34.5 | 81.2 | 58.7 | Chen et al. @ ICLR'15 |
| [48] | ✓ | ✓ | ✓ | 2 | 64.8 | 34.9 | 81.3 | 58.7 | Papandreou et al @ ICCV'15 |
| [37] | ✓ | | | | 66.4 | **46.7** | 82.8 | 67.4 | Lin et al. @ CVPR'16 |
| [79] | ✓ | | | | **67.1** | 42.0 | **86.5** | **71.1** | Yu & Koltun @ ICLR'16 |

# Cross-Dataset Generalization

CITYSCAPES DATASET

https://www.cityscapes-dataset.net

| Dataset | Best reported result | Our result |
|---|---|---|
| Camvid [6] | 62.9 [3] | 72.6 |
| KITTI [53] | 61.6 [3] | 70.9 |
| KITTI [59] | 82.2 [65] | 81.2 |

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