

# COCO dataset and Deeper Deep Architectures



# Microsoft COCO

Common Objects in Context



**Tsung-Yi Lin**  
Cornell Tech



**Michael Maire**  
TTI Chicago



**Serge Belongie**  
Cornell Tech



**Lubomir Bourdev**  
Facebook



**James Hays**  
Brown University



**Pietro Perona**  
Caltech



**Deva Ramanan**  
UC Irvine



**Ross Girshick**  
Microsoft Research



**Piotr Dollar**  
Microsoft Research



**Larry Zitnick**  
Microsoft Research



Microsoft COCO  
Common Objects in Context

# Why a new dataset?

Continue our field's momentum:

IMAGENET



**Caltech Pedestrian Detection Benchmark**



<http://mscoco.org>







- ✓ Instance segmentation
- ✓ Non-iconic Images





# Iconic object images



# Iconic scene images





# Non-iconic images

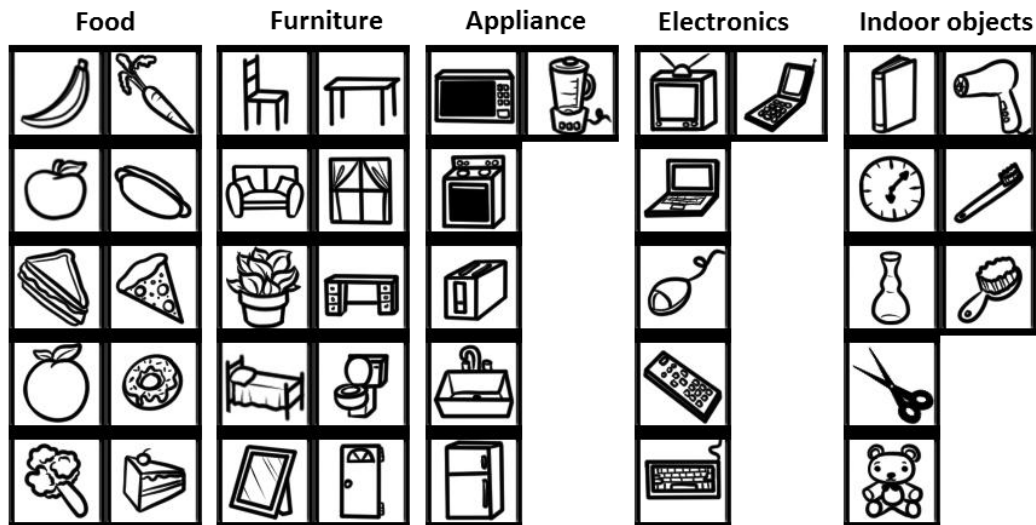
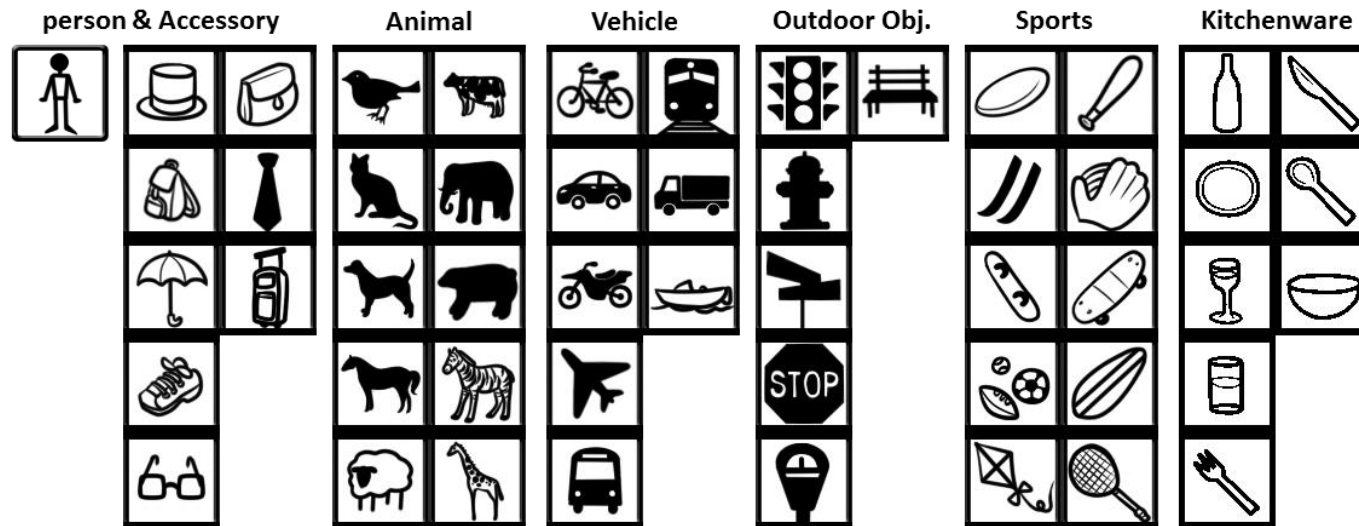








# Object categories (things not stuff)



# flickr

**(All creative commons)**

330,000 images

“Dog”





# “Dog + Car”



*Im2Text: Describing Images Using 1 Million Captioned Photographs, V. Ordonez, G. Kulkarni, T. L. Berg NIPS'11*



Microsoft COCO  
Common Objects in Context

# Annotation pipeline

<http://mscoco.org>







# Divide and Conquer

## 1. Category Labeling



dog, bottle

## 2. Instance spotting



## 3. Instance segmentation





# 1. Category Labeling

Image 4 :



Image  
contains:

Task: select **person** and **accessory** items shown in the image (if any):

	person	hat	back pack	umbrella	shoe	eye glasses	handbag	tie	suitcase	

1/11



## 2. Instance Spotting



car

0 car(s) found in this image.



# 3. Instance Segmentation



<http://opensurfaces.cs.cornell.edu/>

Sean Bell, Paul Upchurch, Noah Snaveley, Kavita Bala,  
Cornell University.

<http://mscoco.org>





FLIP

T.A.Z.E.E





After training







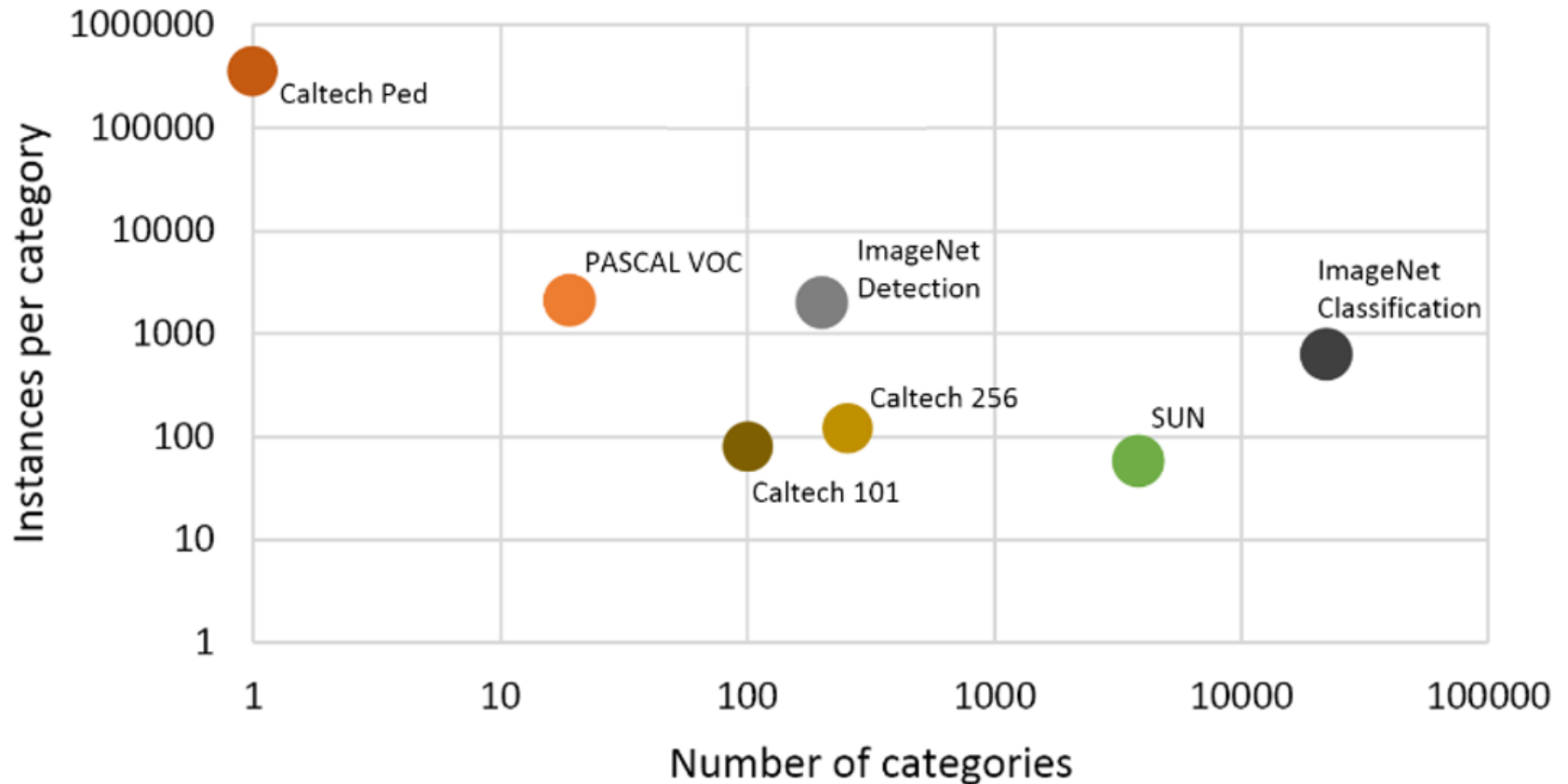




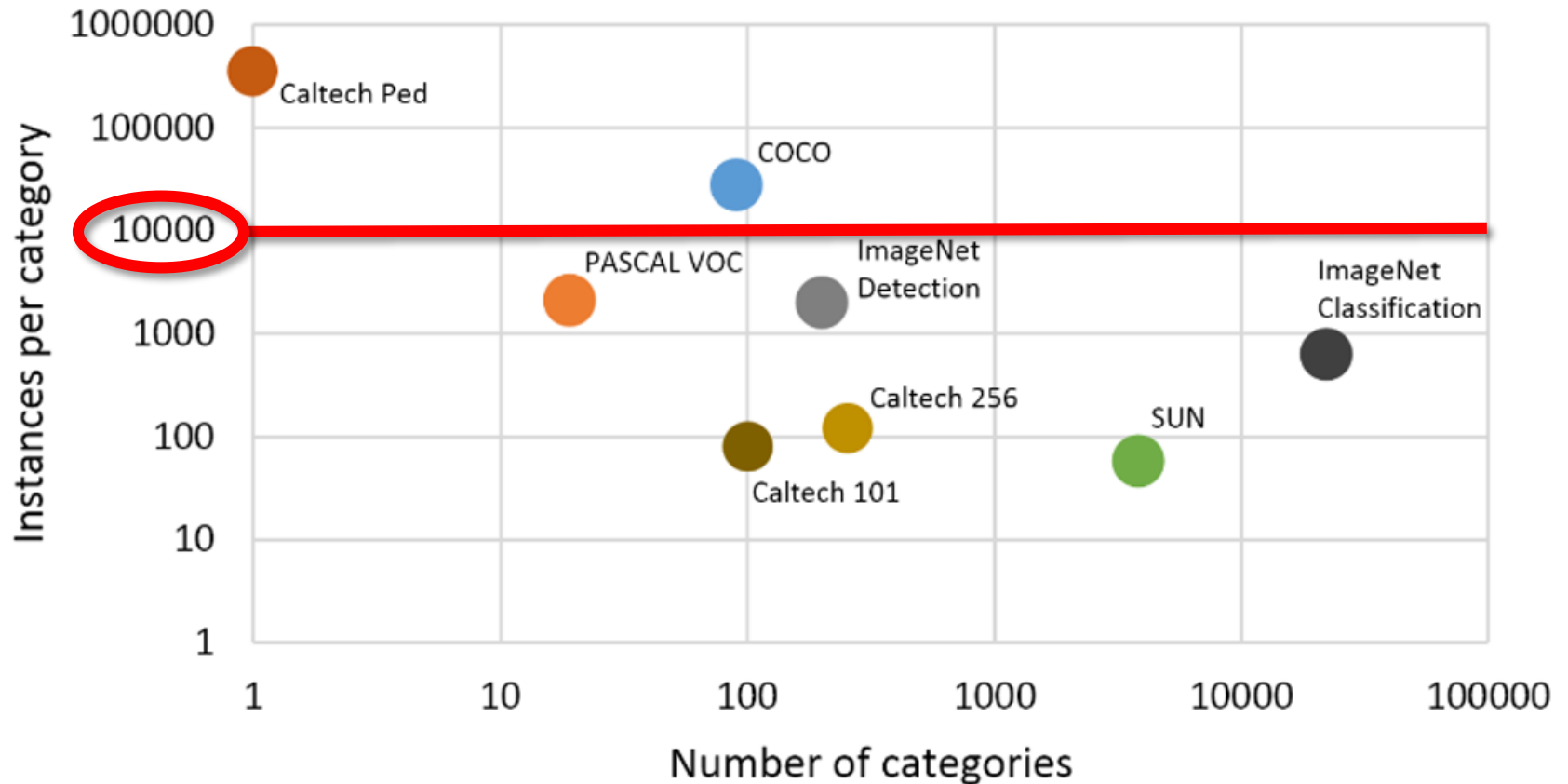
Microsoft COCO  
Common Objects in Context

# Properties

# Number of categories vs. number of instances

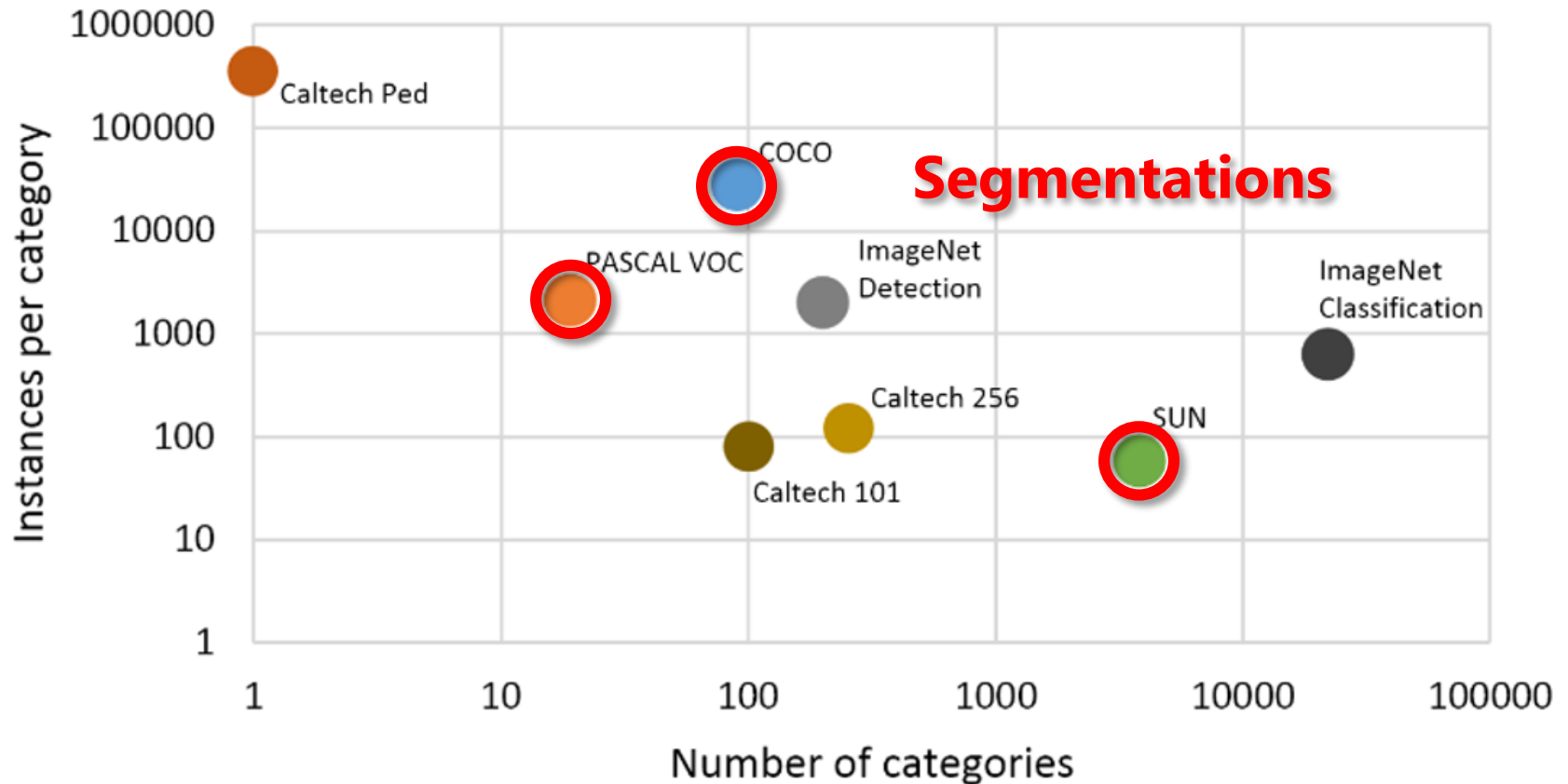


# Number of categories vs. number of instances





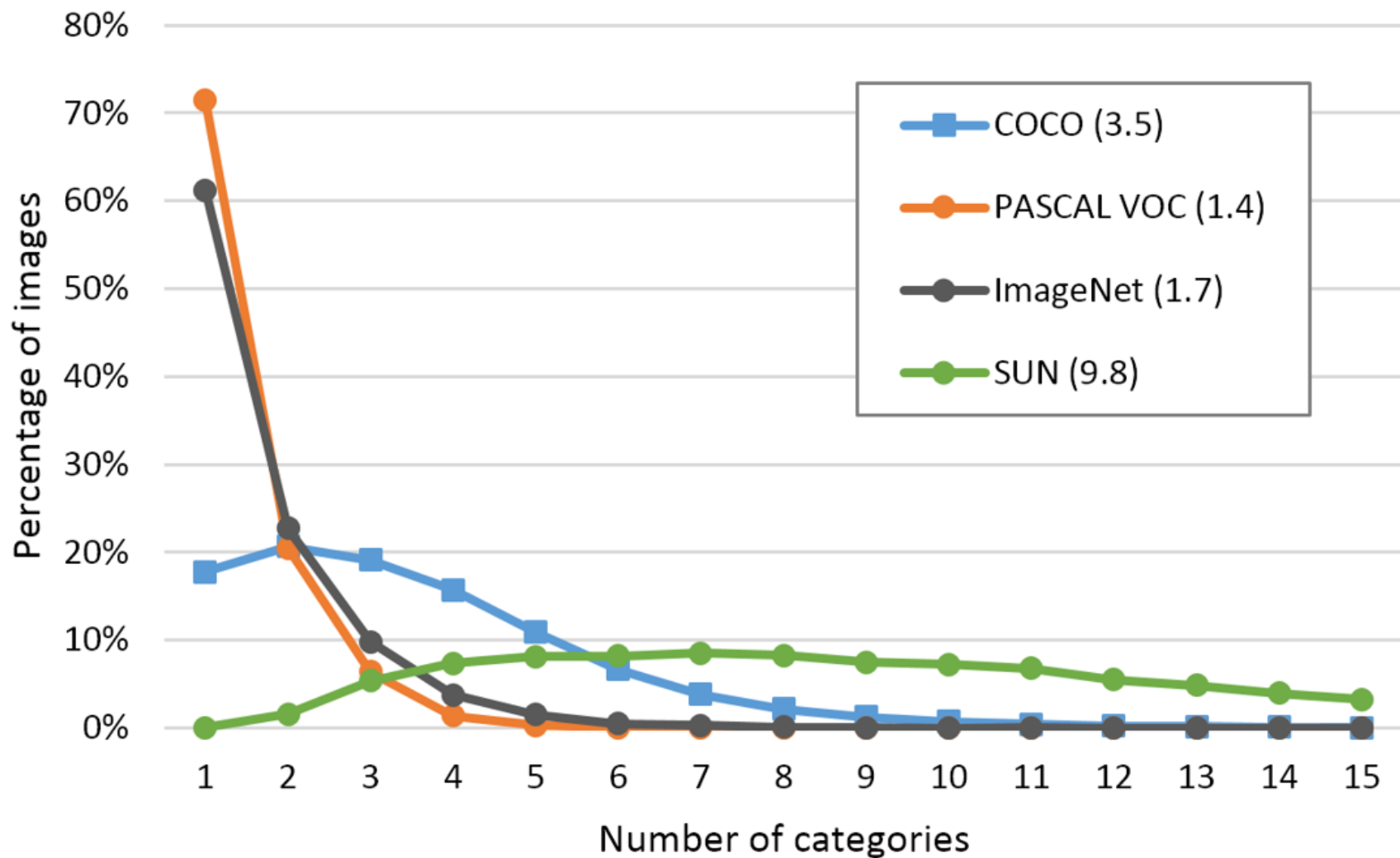
# Number of categories vs. number of instances



- 330,000 images
- >2 million instances (700k people)
- Every instance is segmented
- 7.7 instances per image (3.5 categories)



# Categories per image





# Detection Performance

( DPM V5 )

	Person (mAP)	Average (mAP)
PASCAL VOC	41.3	29.6
MS COCO	17.5	16.9

# http://mscoco.org



 Microsoft COCO  
Common Objects in Context

cocodataset@outlook.com

Home People **Explore** Download

Disclaimer: Annotations are currently in progress (final annotations will be available soon).



person\* elephant\* search

899 results

URL    



http://mscoco.org



Microsoft COCO  
Common Objects in Context

# Beyond detection

# Beyond detection

## ✓ Sentences

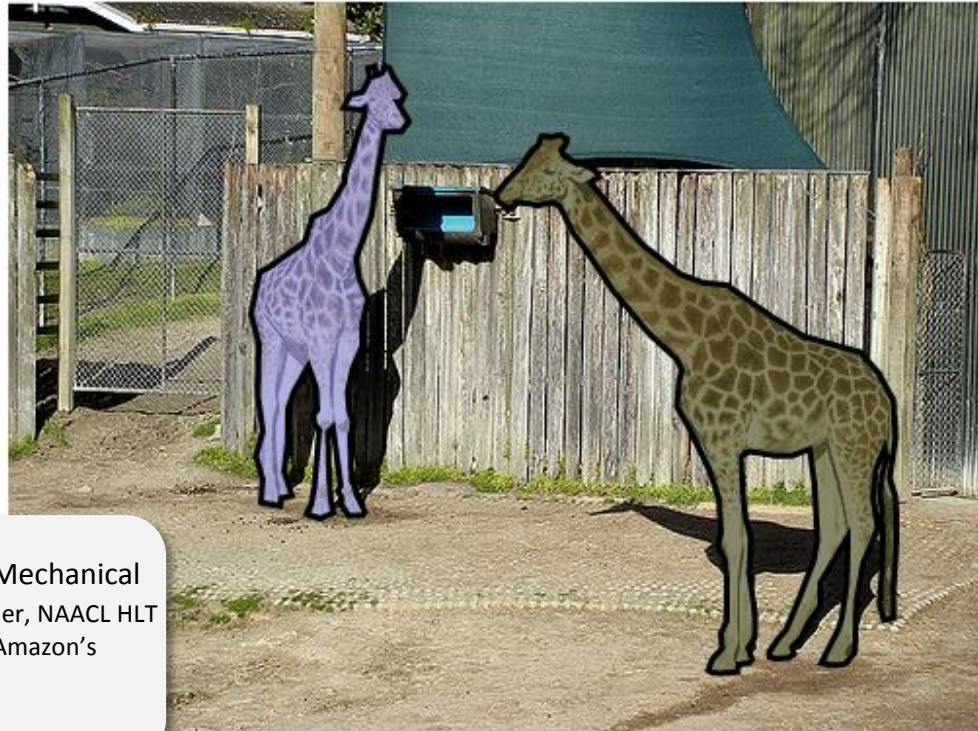
two giraffe standing next to each other in front of a wooden fence.

two giraffes standing in the dirt near a gate.

two giraffes stand by a food box awaiting the goods.

two giraffes are standing next to a wooden fence.

two giraffes standing alone by a picket fence.

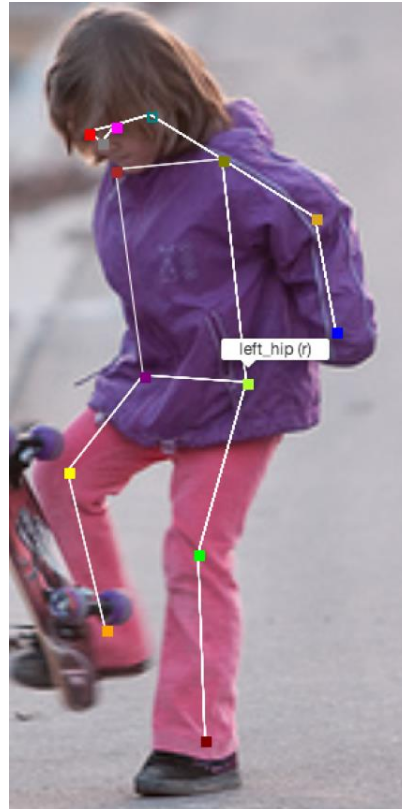


Collecting Image Annotations Using Amazon's Mechanical Turk, C. Rashtchian, P. Young, M. Hodosh, J. Hockenmaier, NAACL HLT Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk, 2010



# Beyond detection

- ✓ Keypoints  
(provided by Facebook)



# MS COCO 2014 release

(half of COCO)

Over 77,000 worker hours (8+ years)

- 160k images
- 80 object categories (things not stuff)
- 1M+ instances (300k people)
- Every instance segmented
- 5 sentences per image
- Separate train and validation set



<http://mscoco.org>

# Beyond AlexNet



# VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

**Karen Simonyan & Andrew Zisserman 2015**

**These are the “VGG” networks.  
Including what you use in Project 6**

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input ( $224 \times 224$ RGB image)					
conv3-64	conv3-64 <b>LRN</b>	conv3-64 <b>conv3-64</b>	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 <b>conv3-128</b>	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 <b>conv1-256</b>	conv3-256 conv3-256 <b>conv3-256</b>	conv3-256 conv3-256 conv3-256 <b>conv3-256</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 <b>conv1-512</b>	conv3-512 conv3-512 <b>conv3-512</b>	conv3-512 conv3-512 conv3-512 <b>conv3-512</b>
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Table 2: **Number of parameters** (in millions).

Network	A,A-LRN	B	C	D	E
Number of parameters	133	133	134	138	144

Table 4: **ConvNet performance at multiple test scales.**

ConvNet config. (Table 1)	smallest image side		top-1 val. error (%)	top-5 val. error (%)
	train ( $S$ )	test ( $Q$ )		
B	256	224,256,288	28.2	9.6
C	256	224,256,288	27.7	9.2
	384	352,384,416	27.8	9.2
	[256; 512]	256,384,512	26.3	8.2
D	256	224,256,288	26.6	8.6
	384	352,384,416	26.5	8.6
	[256; 512]	256,384,512	<b>24.8</b>	<b>7.5</b>
E	256	224,256,288	26.9	8.7
	384	352,384,416	26.7	8.6
	[256; 512]	256,384,512	<b>24.8</b>	<b>7.5</b>

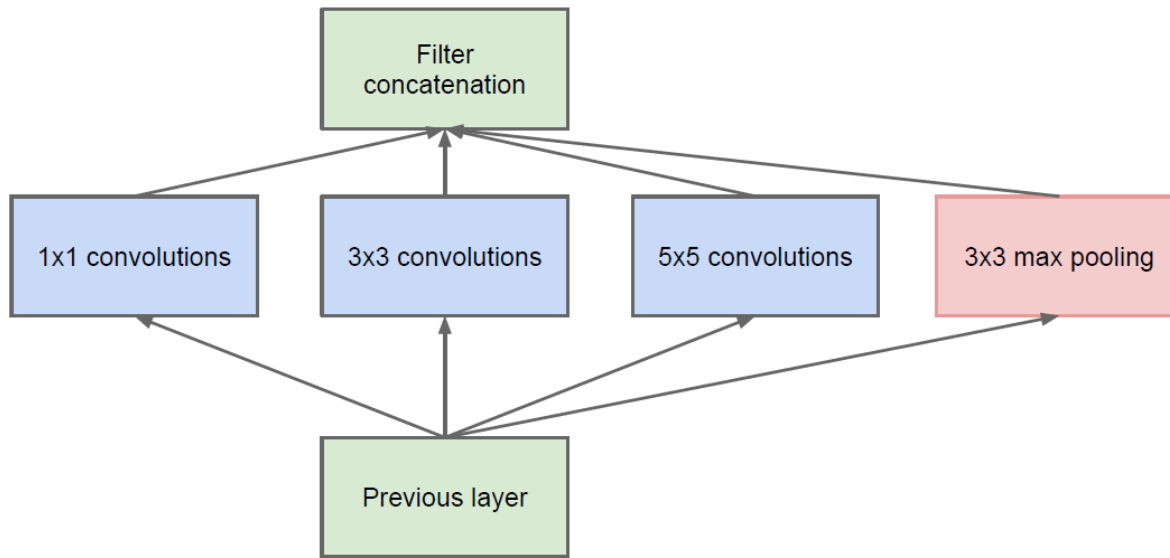


# Going Deeper with Convolutions

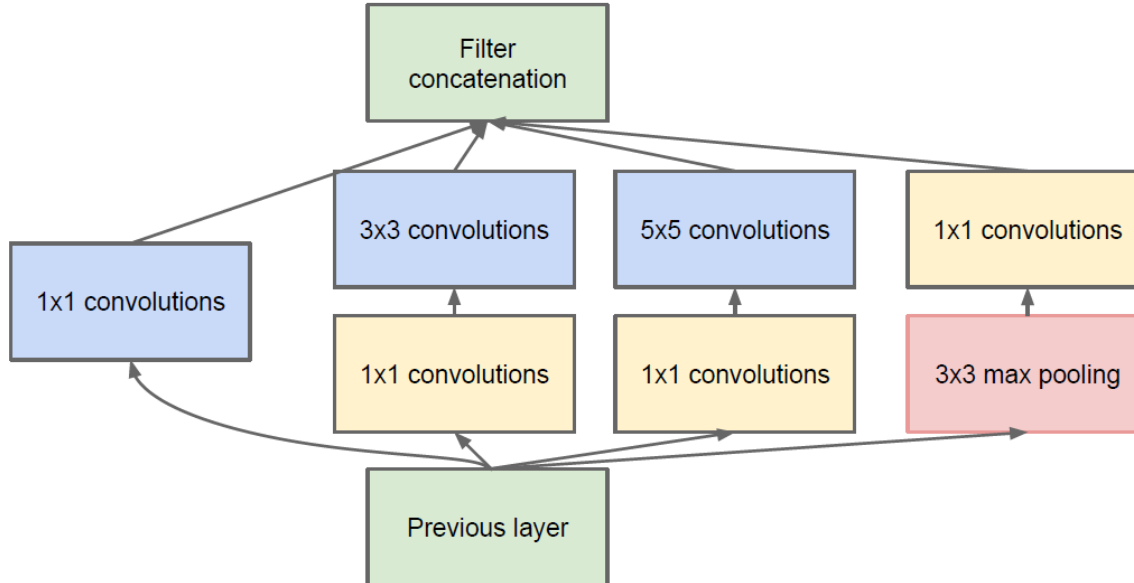
**Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed,  
Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich  
2015**

**This is the “Inception” architecture or “GoogLeNet”**

**\*The architecture blocks are called “Inception” modules  
and the collection of them into a particular net is “GoogLeNet”**



(a) Inception module, naïve version

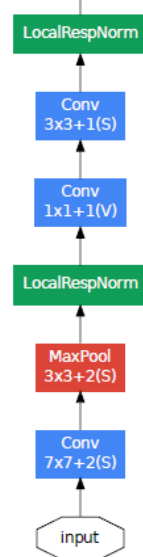
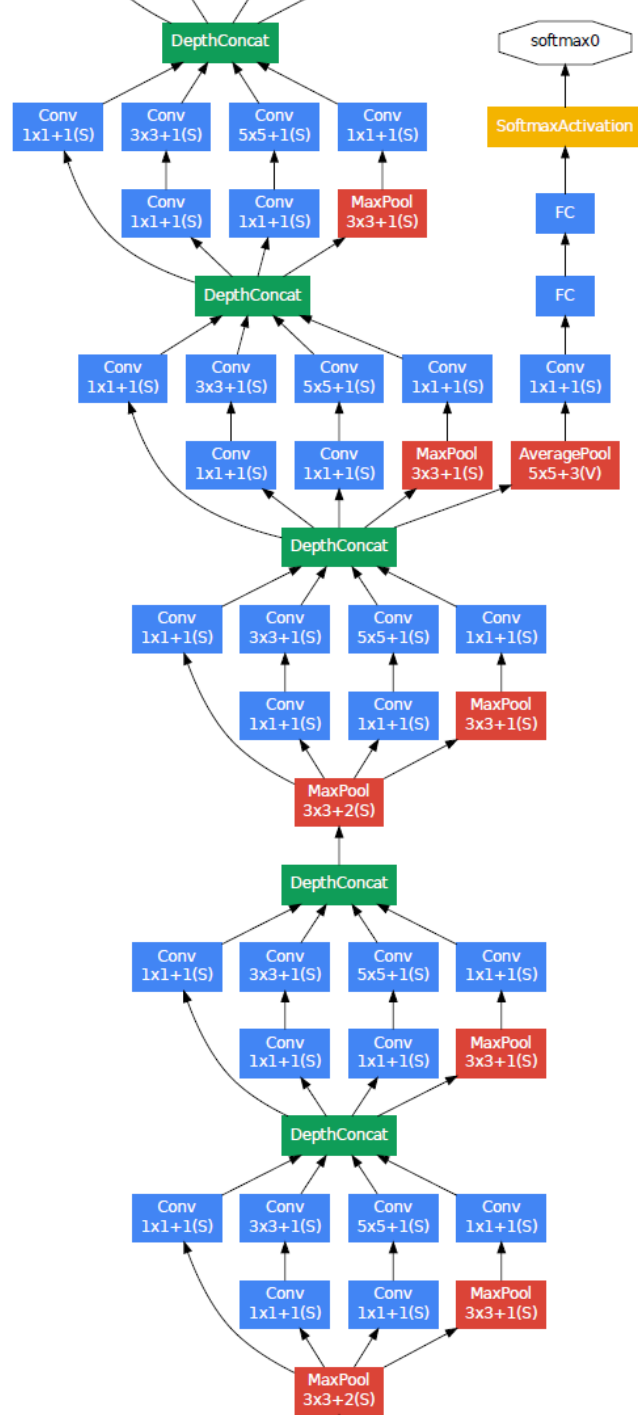
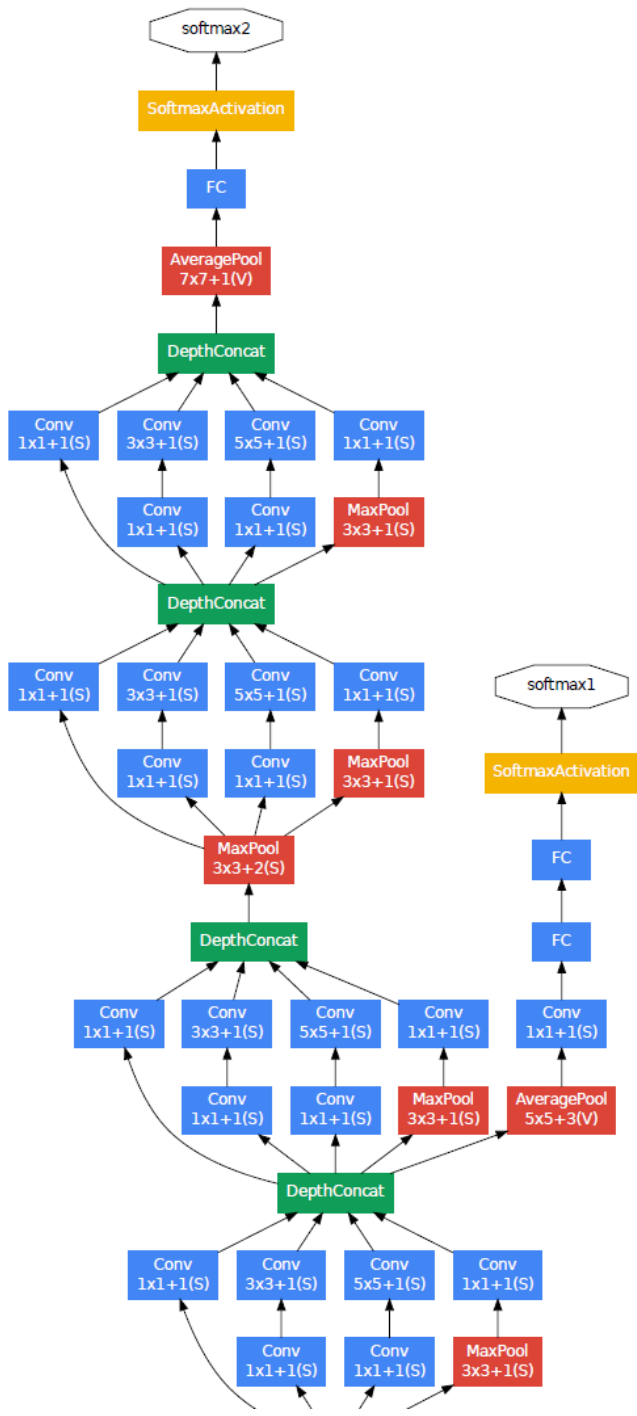


(b) Inception module with dimensionality reduction

type	patch size/ stride	output size	depth	#1×1	#3×3 reduce	#3×3	#5×5 reduce	#5×5	pool proj	params	ops
convolution	7×7/2	112×112×64	1							2.7K	34M
max pool	3×3/2	56×56×64	0								
convolution	3×3/1	56×56×192	2		64	192				112K	360M
max pool	3×3/2	28×28×192	0								
inception (3a)		28×28×256	2	64	96	128	16	32	32	159K	128M
inception (3b)		28×28×480	2	128	128	192	32	96	64	380K	304M
max pool	3×3/2	14×14×480	0								
inception (4a)		14×14×512	2	192	96	208	16	48	64	364K	73M
inception (4b)		14×14×512	2	160	112	224	24	64	64	437K	88M
inception (4c)		14×14×512	2	128	128	256	24	64	64	463K	100M
inception (4d)		14×14×528	2	112	144	288	32	64	64	580K	119M
inception (4e)		14×14×832	2	256	160	320	32	128	128	840K	170M
max pool	3×3/2	7×7×832	0								
inception (5a)		7×7×832	2	256	160	320	32	128	128	1072K	54M
inception (5b)		7×7×1024	2	384	192	384	48	128	128	1388K	71M
avg pool	7×7/1	1×1×1024	0								
dropout (40%)		1×1×1024	0								
linear		1×1×1000	1							1000K	1M
softmax		1×1×1000	0								

Only 6.8 million parameters. AlexNet ~60 million, VGG up to 138 million





<b>Team</b>	<b>Year</b>	<b>Place</b>	<b>Error (top-5)</b>	<b>Uses external data</b>
SuperVision	2012	1st	16.4%	no
SuperVision	2012	1st	15.3%	Imagenet 22k
Clarifai	2013	1st	11.7%	no
Clarifai	2013	1st	11.2%	Imagenet 22k
MSRA	2014	3rd	7.35%	no
VGG	2014	2nd	7.32%	no
GoogLeNet	2014	1st	6.67%	no

Table 2: Classification performance.

<b>Number of models</b>	<b>Number of Crops</b>	<b>Cost</b>	<b>Top-5 error</b>	<b>compared to base</b>
1	1	1	10.07%	base
1	10	10	9.15%	-0.92%
1	144	144	7.89%	-2.18%
7	1	7	8.09%	-1.98%
7	10	70	7.62%	-2.45%
7	144	1008	6.67%	-3.45%

Surely it would be ridiculous to go any deeper...

- ResNet

- See the following slides:

[http://kaiminghe.com/cvpr16resnet/cvpr2016\\_deep\\_residual\\_learning\\_kaiminghe.pdf](http://kaiminghe.com/cvpr16resnet/cvpr2016_deep_residual_learning_kaiminghe.pdf)