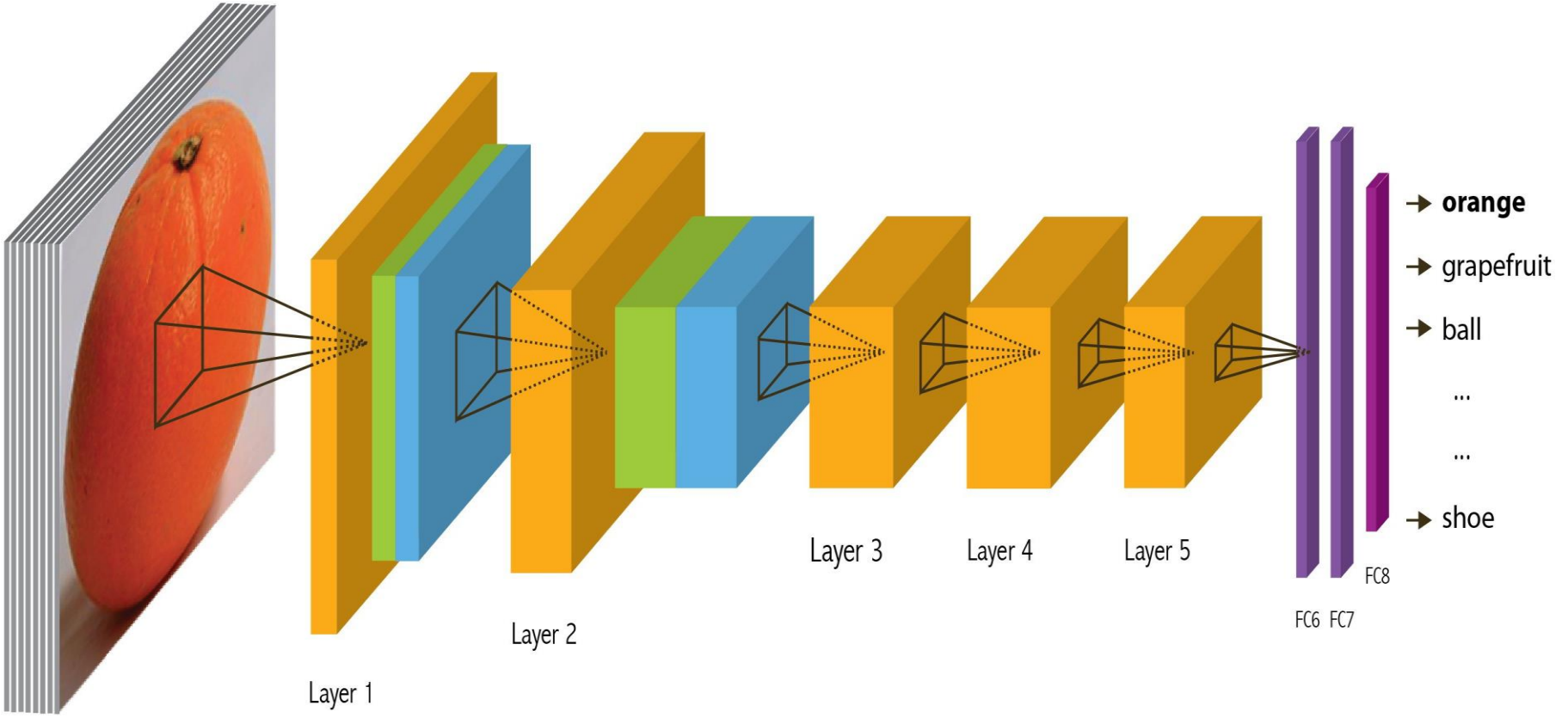

2021-May-23 14:32:32.672 (BST)

Deeper Deep Learning and Semantic Segmentation

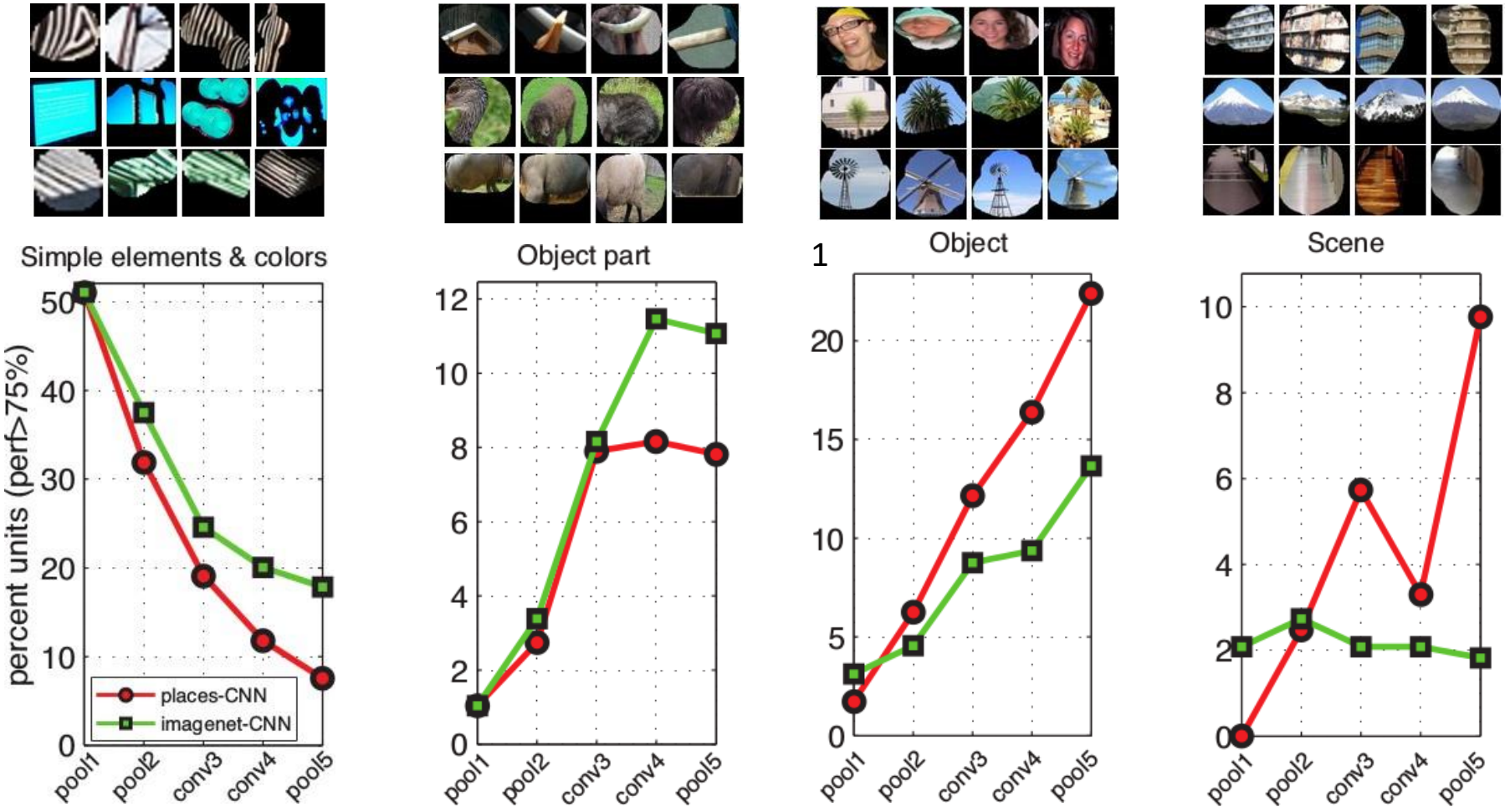
Today’s outline

- What do more recent deep learning architectures look like?
  - VGG Net
  - Google Inception architectures
  - ResNet

Recap: Convolutional Network, AlexNet



Recap: Convolutional Network Interpretation



Object detectors emerge within CNN trained to classify scenes, without any object supervision!

Beyond AlexNet

VERY DEEP CONVOLUTIONAL NETWORKS FOR LARGE-SCALE IMAGE RECOGNITION

Karen Simonyan & Andrew Zisserman 2015

These are the “VGG” networks.
“Perceptual Loss” in generative deep learning refers to these networks

150k citations as of 10/14/2025

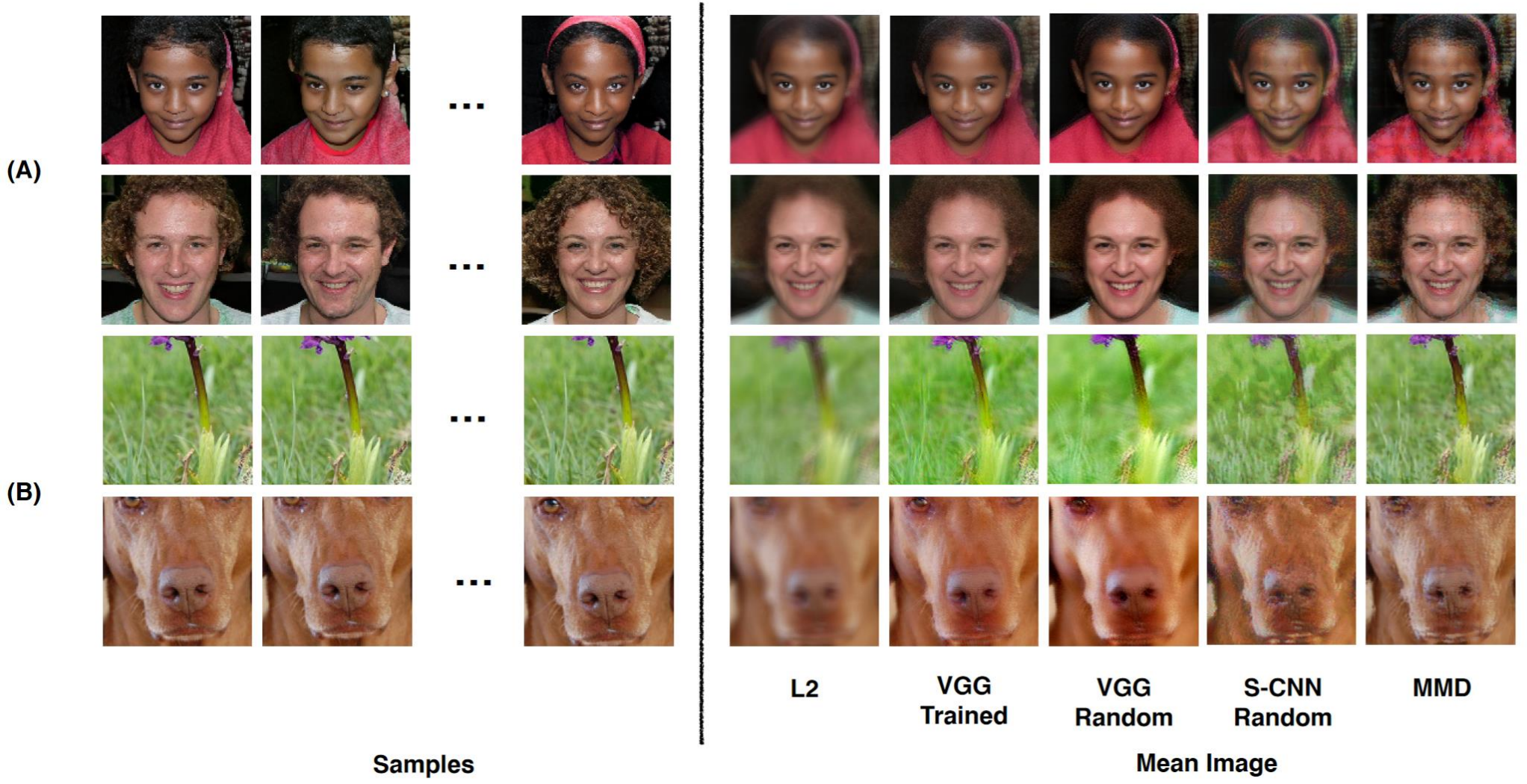
ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	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 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
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 7: **Comparison with the state of the art in ILSVRC classification**. Our method is denoted as "VGG". Only the results obtained without outside training data are reported.

Method	top-1 val. error (%)	top-5 val. error (%)	top-5 test error (%)
VGG (2 nets, multi-crop & dense eval.)	**23.7**	**6.8**	**6.8**
VGG (1 net, multi-crop & dense eval.)	24.4	7.1	7.0
VGG (ILSVRC submission, 7 nets, dense eval.)	24.7	7.5	7.3
GoogLeNet (Szegedy et al., 2014) (1 net)	-	7.9	
GoogLeNet (Szegedy et al., 2014) (7 nets)	-	**6.7**	
MSRA (He et al., 2014) (11 nets)	-	-	8.1
MSRA (He et al., 2014) (1 net)	27.9	9.1	9.1
Clarifai (Russakovsky et al., 2014) (multiple nets)	-	-	11.7
Clarifai (Russakovsky et al., 2014) (1 net)	-	-	12.5
Zeiler & Fergus (Zeiler & Fergus, 2013) (6 nets)	36.0	14.7	14.8
Zeiler & Fergus (Zeiler & Fergus, 2013) (1 net)	37.5	16.0	16.1
OverFeat (Sermanet et al., 2014) (7 nets)	34.0	13.2	13.6
OverFeat (Sermanet et al., 2014) (1 net)	35.7	14.2	-
Krizhevsky et al. (Krizhevsky et al., 2012) (5 nets)	38.1	16.4	16.4
Krizhevsky et al. (Krizhevsky et al., 2012) (1 net)	40.7	18.2	-



“VGG” networks are commonly used as the basis for “Perceptual Loss”.
The images on the right are as close as possible to all images on the left in various feature spaces.

Understanding and Simplifying Perceptual Distances. Dan Amir and Yair Weiss. CVPR 2021

Generative Image Dynamics

Zhengqi Li, Richard Tucker, Noah Snavely, Aleksander Holynski
Google Research

CVPR 2024 Best Paper Award

Paper arXiv Demo Supp

https://generative-dynamics.github.io/

Abstract

We present an approach to modeling an image-space prior on scene motion. Our prior is learned from a collection of motion trajectories extracted from real video sequences depicting natural, oscillatory dynamics such as trees, flowers, candles, and clothes swaying in the wind. We model this dense, long-term motion prior in the Fourier domain:given a single image, our trained model uses a frequency-coordinated diffusion sampling process to predict a spectral volume, which can be converted into a motion texture that spans an entire video. Along with an image-based rendering module, these trajectories can be used for a number of downstream applications, such as turning still images into seamlessly looping videos, or allowing users to realistically interact with objects in real pictures by interpreting the spectral volumes as image-space modal bases, which approximate object dynamics.

Generative Image Dynamics

Zhengqi Li, Richard Tucker, Noah Snavely, Aleksander Holynski
Google Research

CVPR 2024 Best Paper Award

https://generative-dynamics.github.io/

Input still picture

Interactive dynamics

We jointly train the feature extractor and synthesis networks with start and target frames (I_0, I_t) randomly sampled from real videos, using the estimated flow field from I_0 to I_t to warp encoded features from I_0, and supervising predictions $\hat{I}_t$ against I_t with a VGG perceptual loss [49].

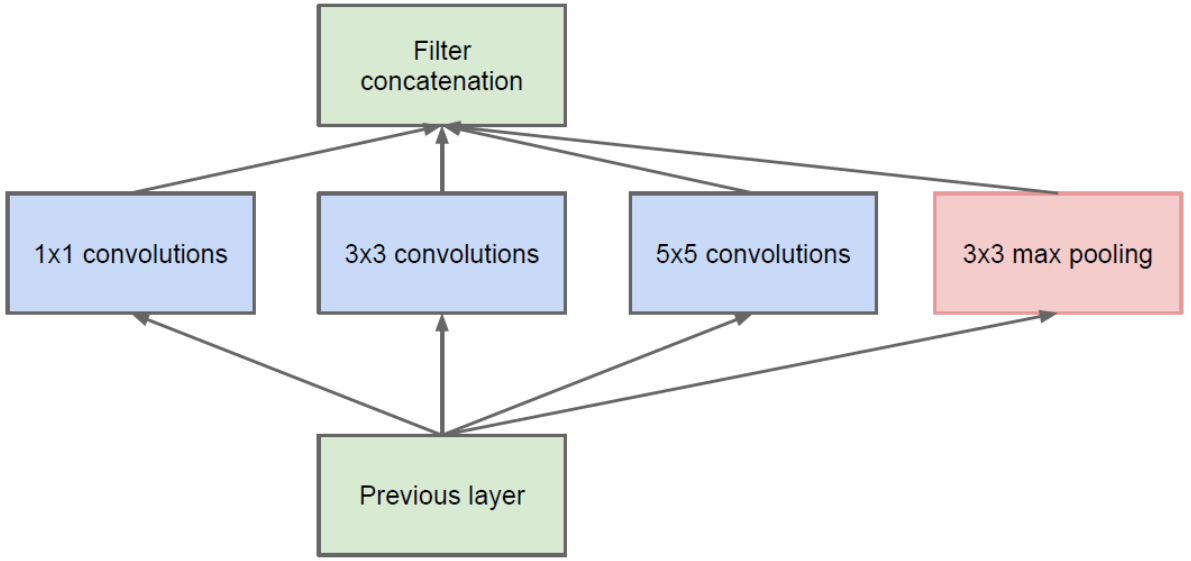
Going Deeper with Convolutions

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

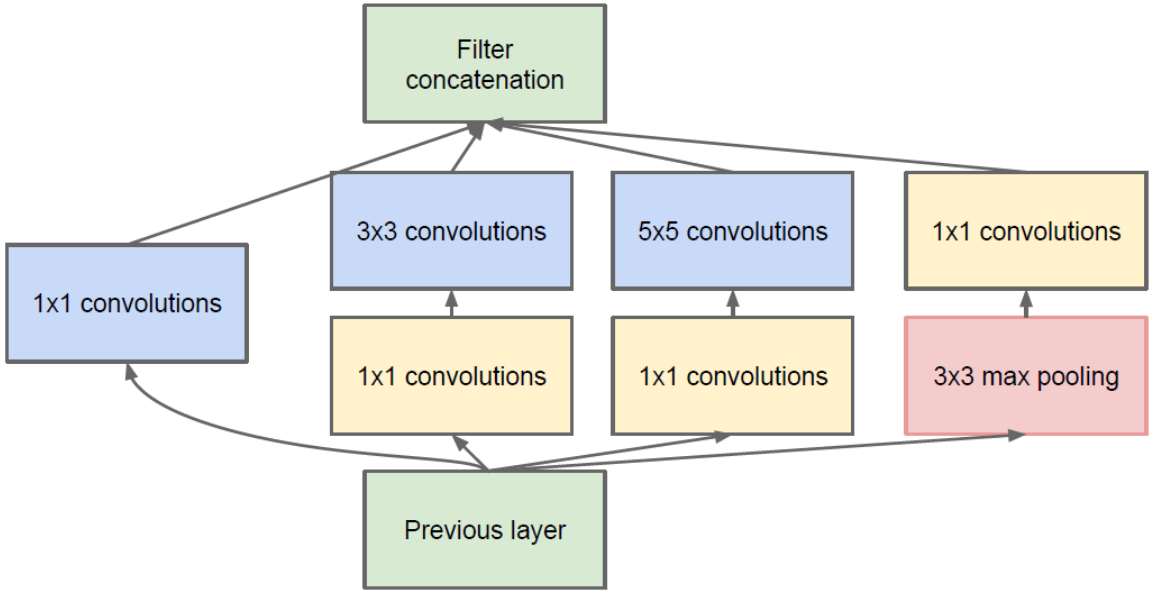
This is the “Inception” architecture or “GoogLeNet”

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

70k citations as of 10/14/2025



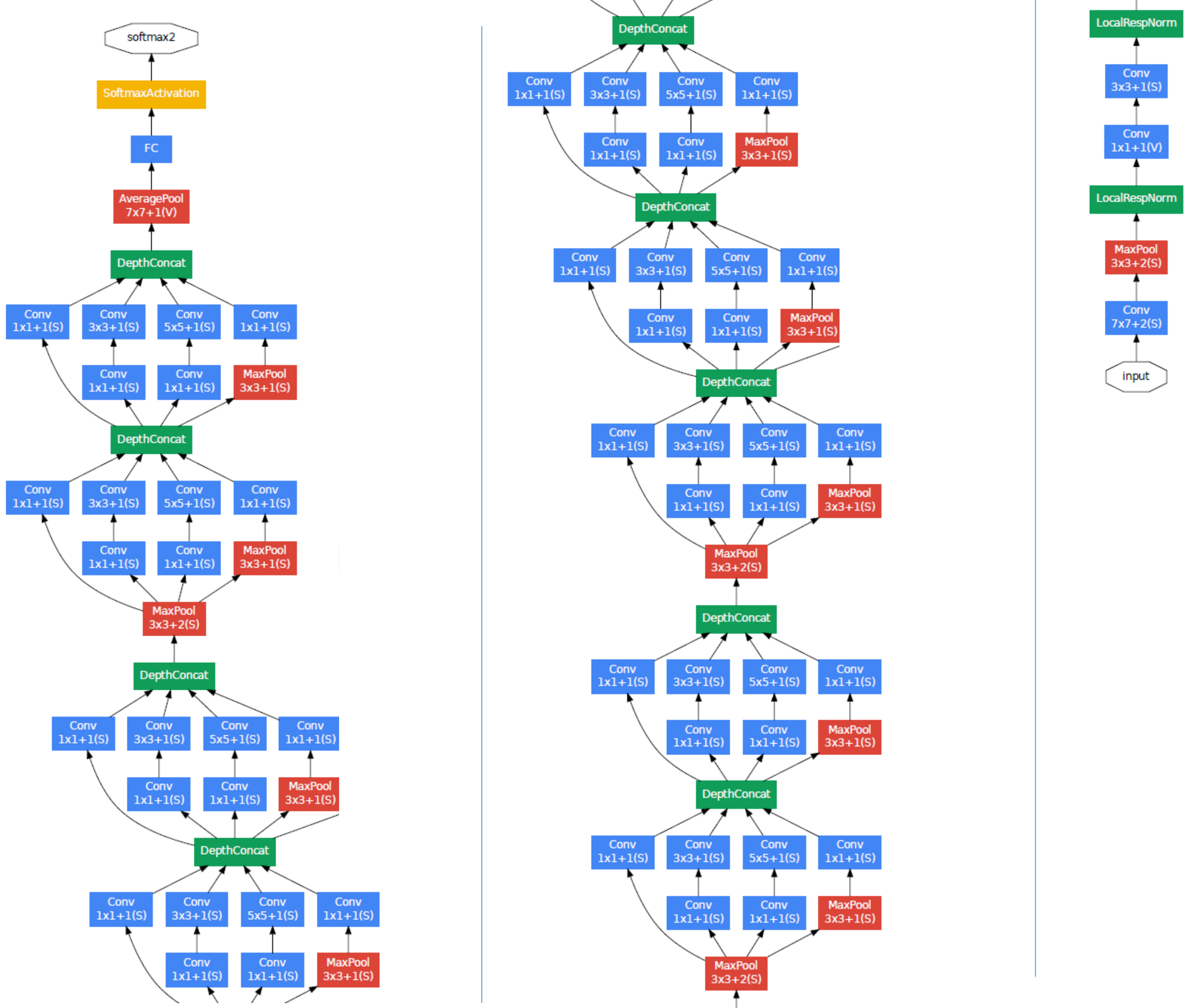
(a) Inception module, naïve version



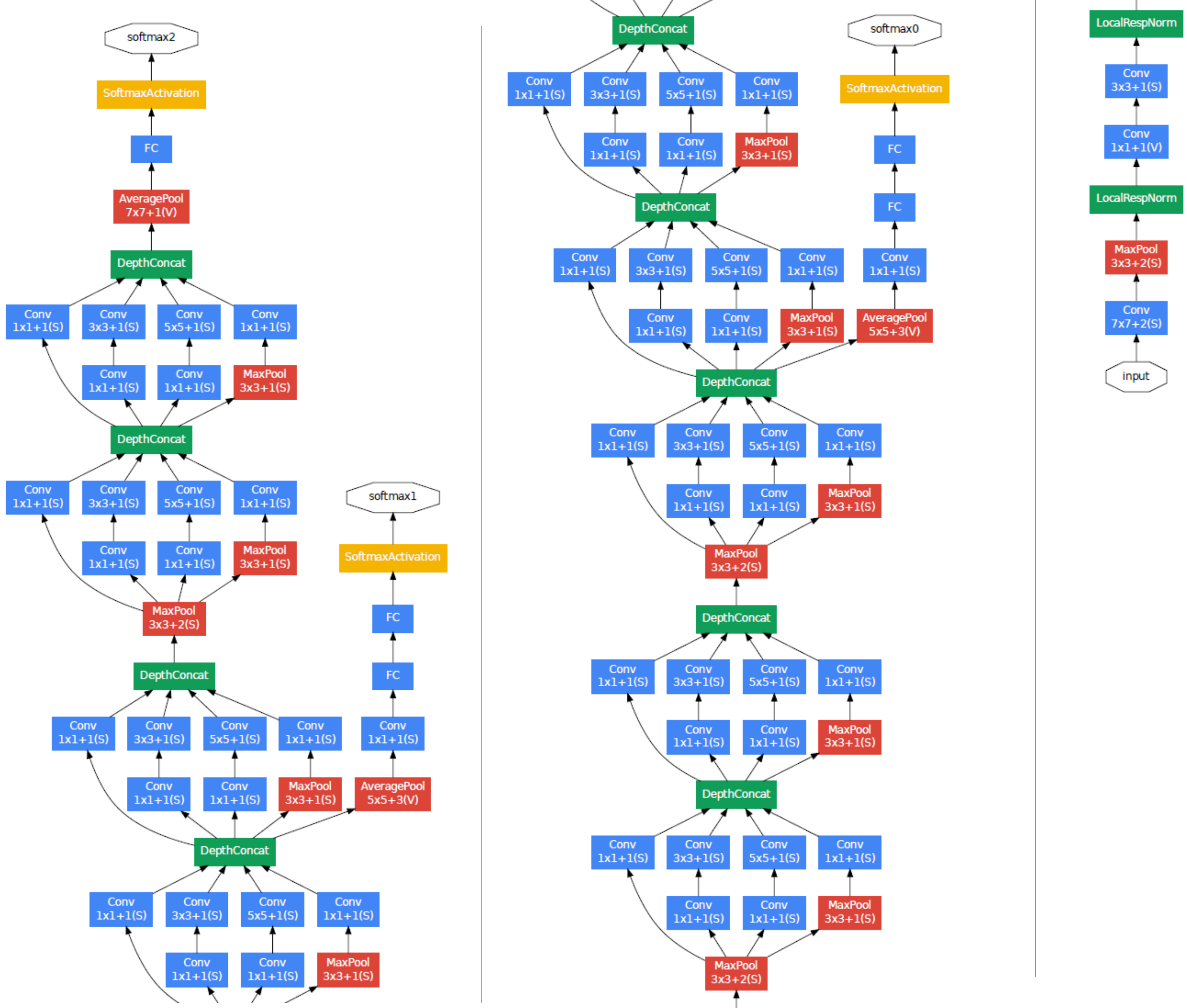
(b) Inception module with dimensionality reduction

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



softmax2
SoftmaxActivation
FC
AveragePool 7x7+1(V)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
MaxPool 3x3+2(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
MaxPool 3x3+2(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
MaxPool 3x3+2(S)
LocalRespNorm
Conv 3x3+1(S)
Conv 1x1+1(V)
LocalRespNorm
MaxPool 3x3+2(S)
Conv 7x7+2(S)
input

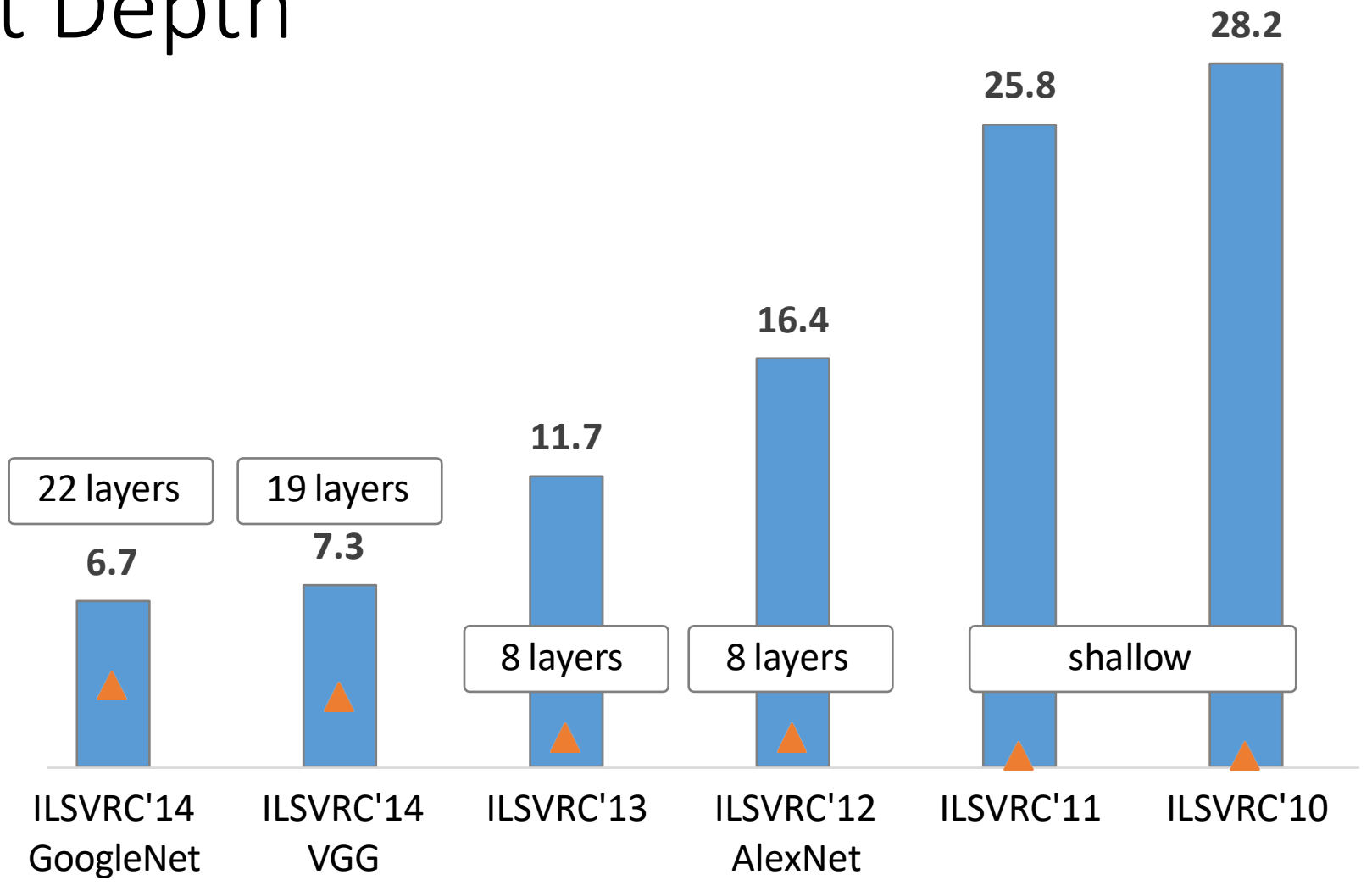

softmax2
SoftmaxActivation
FC
AveragePool 7x7+1(V)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
MaxPool 3x3+2(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
softmax1
SoftmaxActivation
FC
FC
Conv 1x1+1(S)
AveragePool 5x5+3(V)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
softmax0
SoftmaxActivation
FC
FC
Conv 1x1+1(S)
AveragePool 5x5+3(V)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
MaxPool 3x3+2(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
DepthConcat
Conv 1x1+1(S)
Conv 3x3+1(S)
Conv 5x5+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
Conv 1x1+1(S)
MaxPool 3x3+1(S)
MaxPool 3x3+2(S)
LocalRespNorm
Conv 3x3+1(S)
Conv 1x1+1(V)
LocalRespNorm
MaxPool 3x3+2(S)
Conv 7x7+2(S)
input

Team	Year	Place	Error (top-5)	Uses external data
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.

Number of models	Number of Crops	Cost	Top-5 error	compared to base
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%

ConvNet Depth

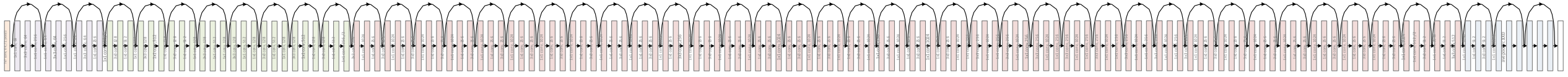


ImageNet Classification top-5 error (%)

Surely it would be ridiculous to go any deeper…

Deep Residual Learning for Image Recognition

Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun



Cited by 290k papers as of 10/14/2025
Nearing the most cited paper *ever*

Top ten most-cited works of the twenty-first century

Rank (median)	Citation	Times cited (range across databases)
1	Deep residual learning for image recognition (2016, preprint 2015)	103,756–254,074
2	Analysis of relative gene expression data using real-time quantitative PCR and the $2^{-\Delta\Delta C_T}$ method (2001)	149,953–185,480
3	Using thematic analysis in psychology (2006)	100,327–230,391
4	*Diagnostic and Statistical Manual of Mental Disorders, DSM-5* (2013)	98,312–367,800
5	A short history of SHELX (2007)	76,523–99,470
6	Random forests (2001)	31,809–146,508
7	Attention is all you need (2017)	56,201–150,832
8	ImageNet classification with deep convolutional neural networks (2017)	46,860–137,997

https://www.nature.com/articles/d41586-025-01125-9

	Publication	h5-index	h5-median
1.	Nature	444	667
2.	The New England Journal of Medicine	432	780
3.	Science	401	614
4.	IEEE/CVF Conference on Computer Vision and Pattern Recognition	389	627
5.	The Lancet	354	635
6.	Advanced Materials	312	418
7.	Nature Communications	307	428
8.	Cell	300	505
9.	International Conference on Learning Representations	286	533
10.	Neural Information Processing Systems	278	436
11.	JAMA	267	425
12.	Chemical Reviews	265	444
13.	Proceedings of the National Academy of Sciences	256	364
14.	Angewandte Chemie	245	332
15.	Chemical Society Reviews	244	386
16.	Journal of the American Chemical Society	242	344
17.	IEEE/CVF International Conference on Computer Vision	239	415
18.	Nucleic Acids Research	238	550
19.	International Conference on Machine Learning	237	421

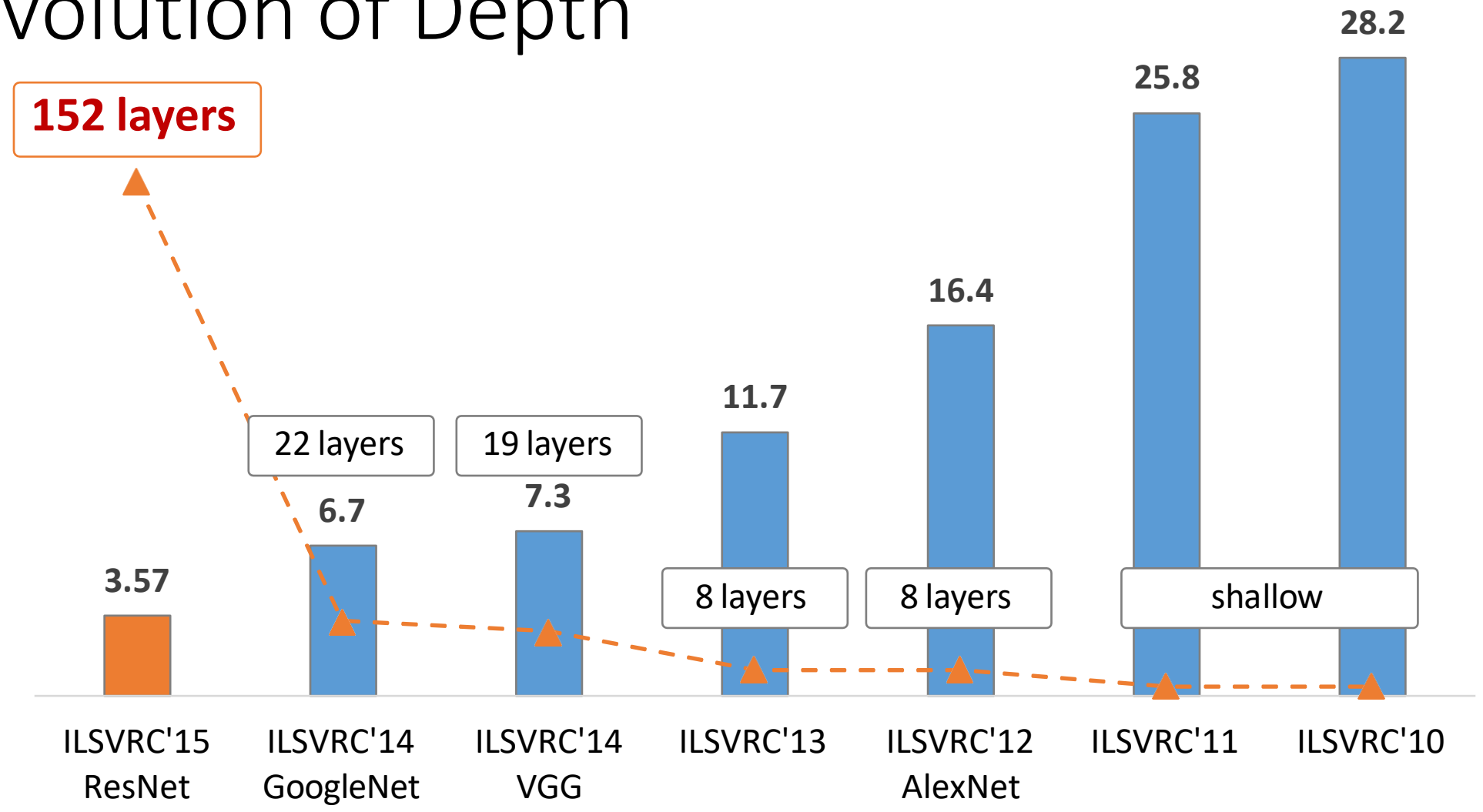
ResNet @ ILSVRC & COCO 2015 Competitions

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

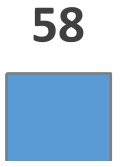
Revolution of Depth



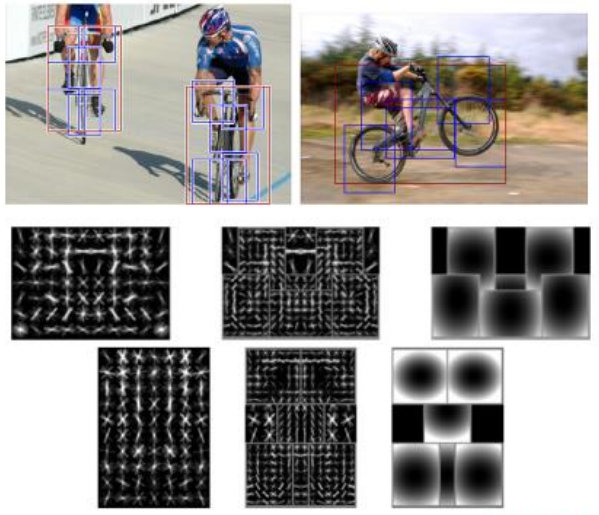
ImageNet Classification top-5 error (%)

Revolution of Depth

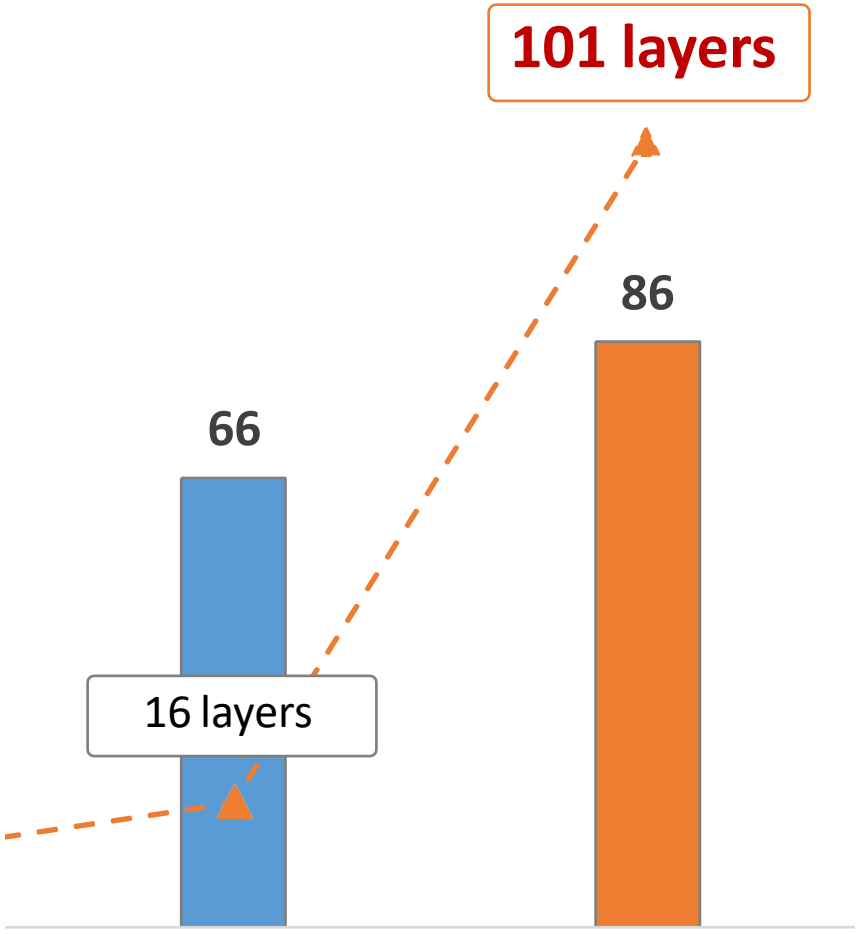
Engines of visual recognition



Discriminatively trained part-based models



P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, "Object Detection with Discriminatively Trained Part-Based Models," PAMI 2009

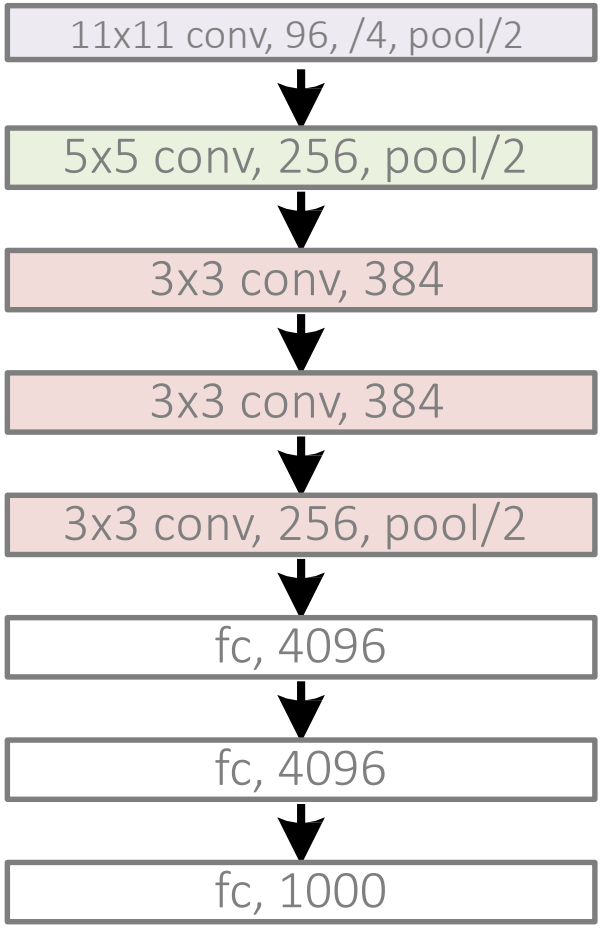


ject Detection mAP (%)

*w/ other improvements & more data

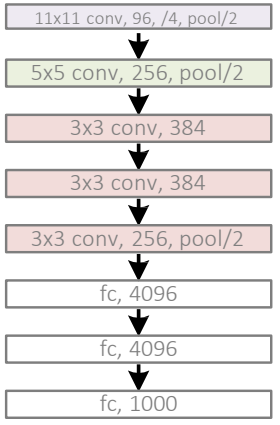
Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)

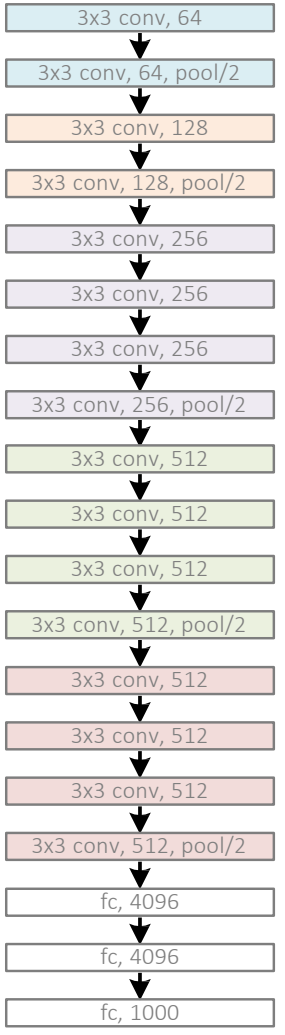


Revolution of Depth

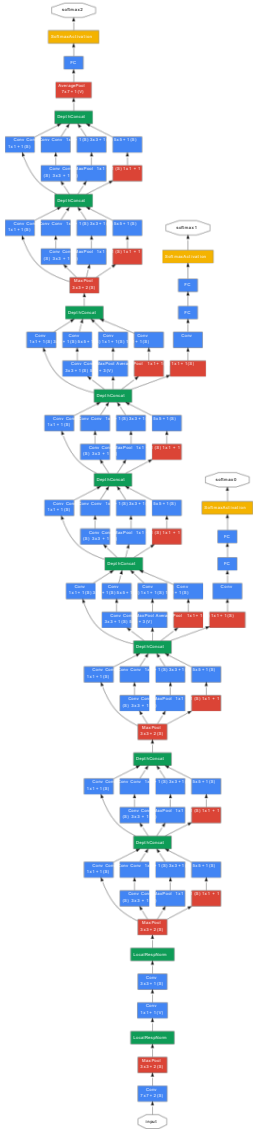
AlexNet, 8 layers
(ILSVRC 2012)



VGG, 19 layers
(ILSVRC 2014)



GoogleNet, 22 layers
(ILSVRC 2014)

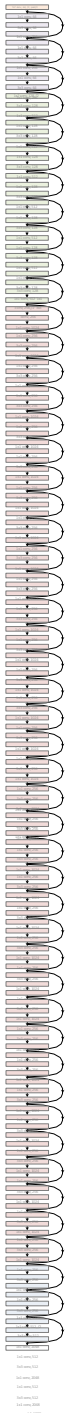


Revolution of Depth

AlexNet, 8 layers
(ILSVRC 2012)

VGG, 19 layers
(ILSVRC 2014)

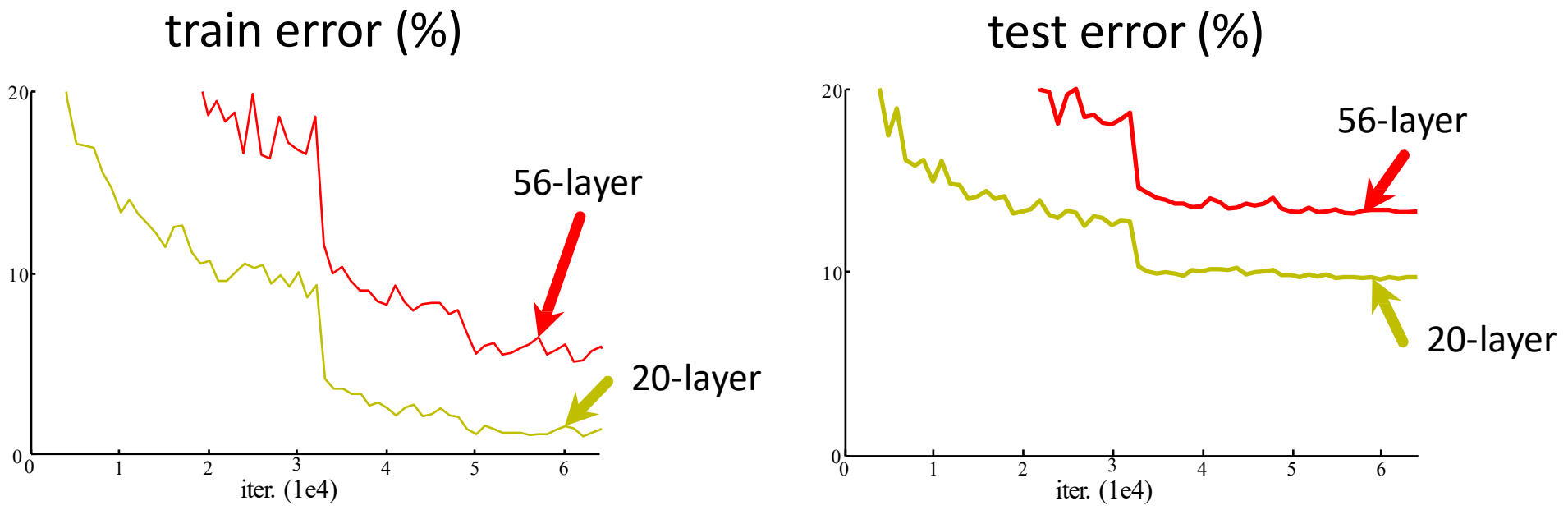
ResNet, 152 layers
(ILSVRC 2015)



Is learning better networks as simple as stacking more layers?

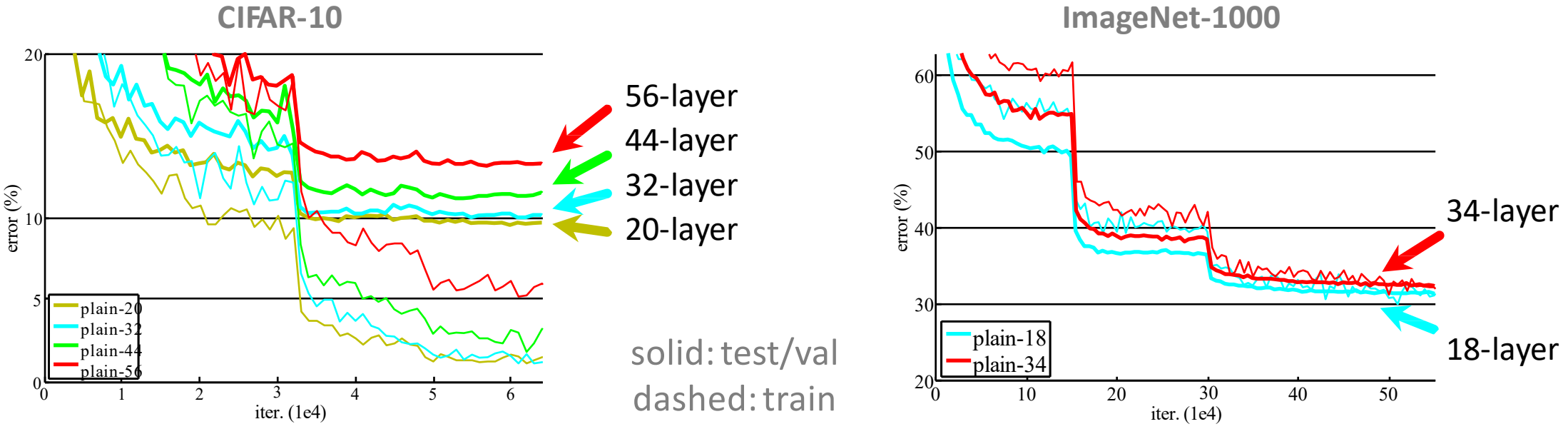
Simply stacking layers?

CIFAR-10



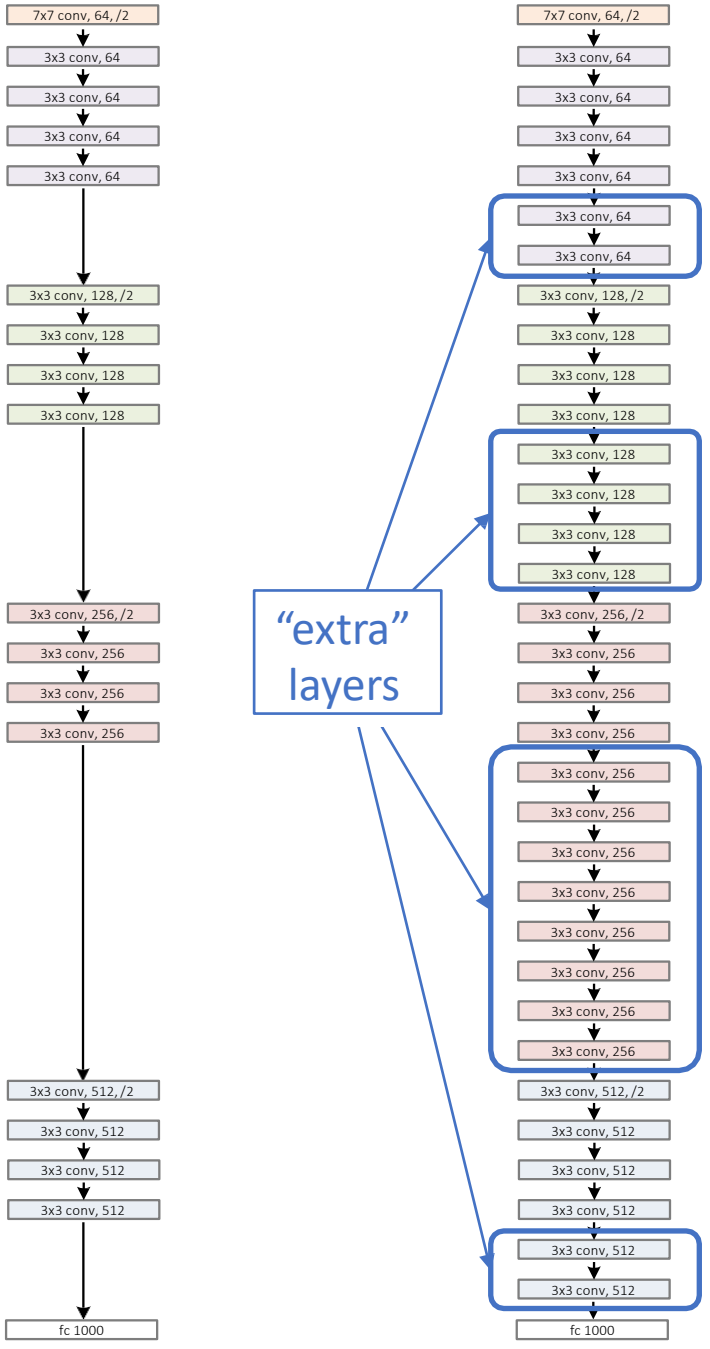
- *Plain* nets: stacking 3x3 conv layers…
- 56-layer net has **higher training error** and test error than 20-layer net

Simply stacking layers?



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

a shallower model (18 layers)



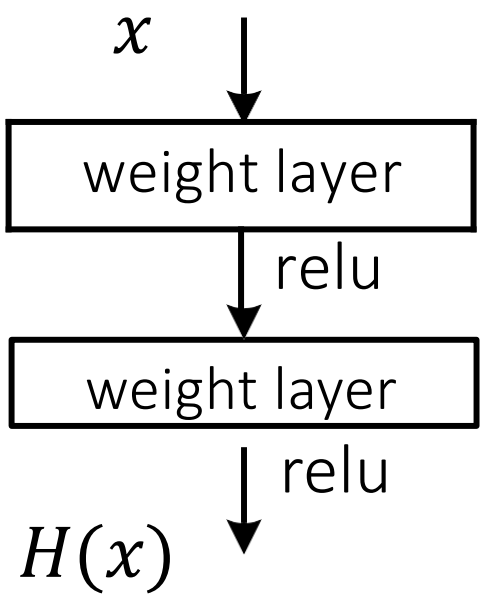
a deeper counterpart (34 layers)

- 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

any two stacked layers

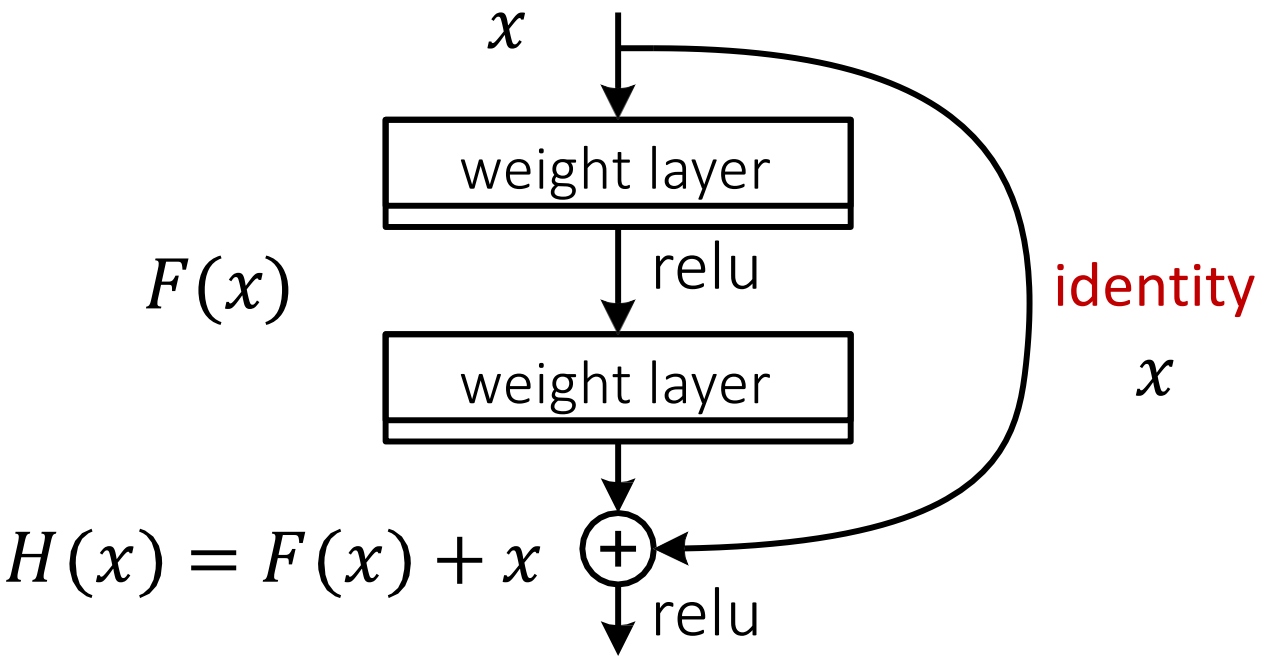


$H(x)$ is any desired mapping,

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

Deep Residual Learning

- Residual net



$H(x)$ is any desired mapping,

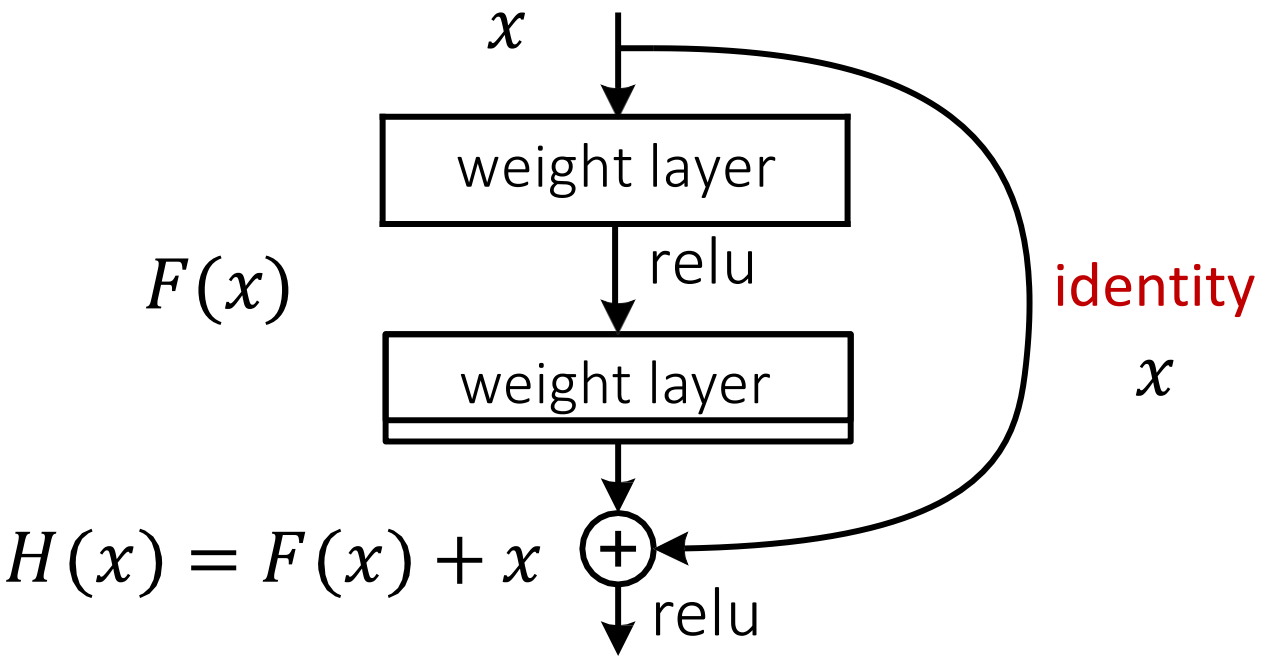
~~hope the 2 weight layers fit $H(x)$~~

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

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

Deep Residual Learning

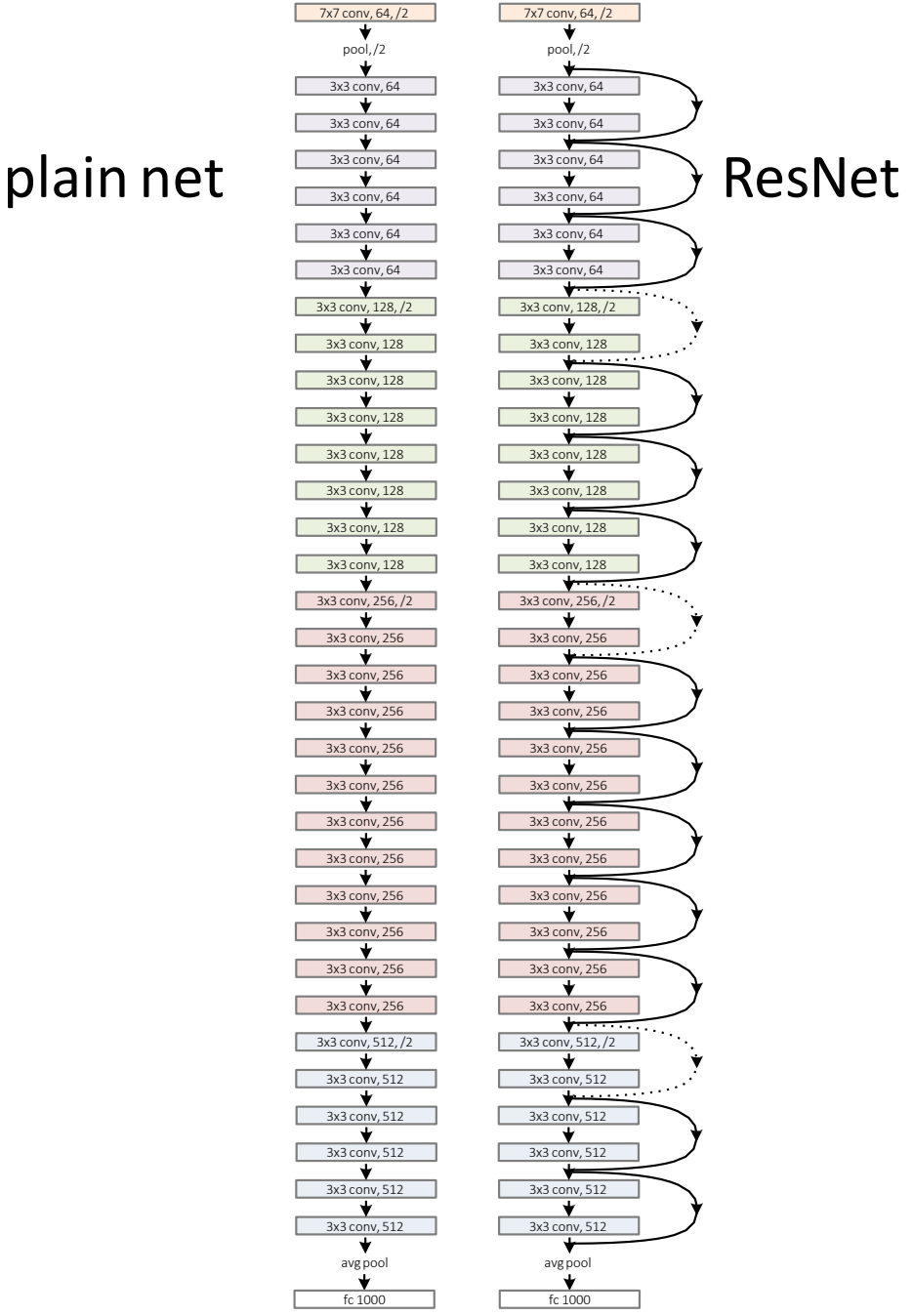
- $F(x)$ is a residual mapping w.r.t. identity



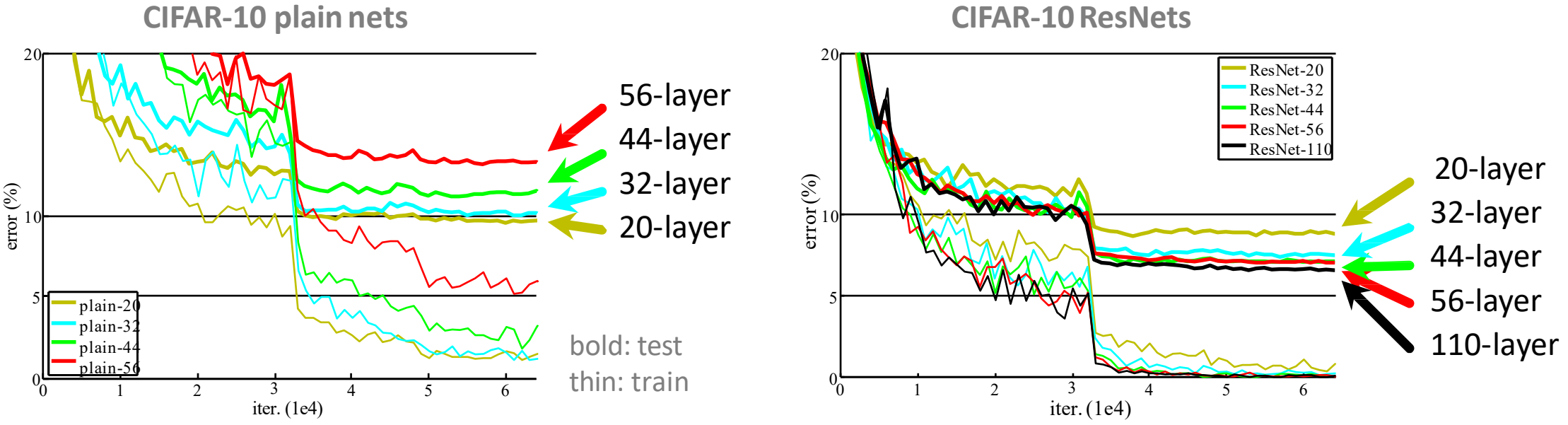
- If identity were optimal, easy to set weights as 0
- If optimal mapping is closer to identity, easier to find small fluctuations

Network “Design”

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

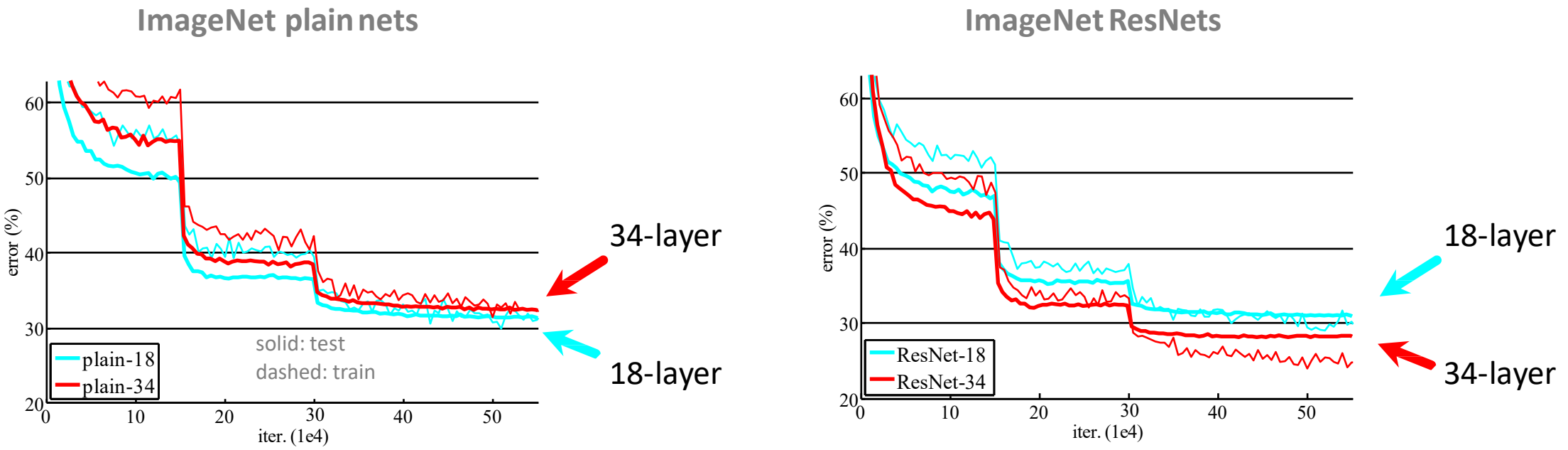


CIFAR-10 experiments



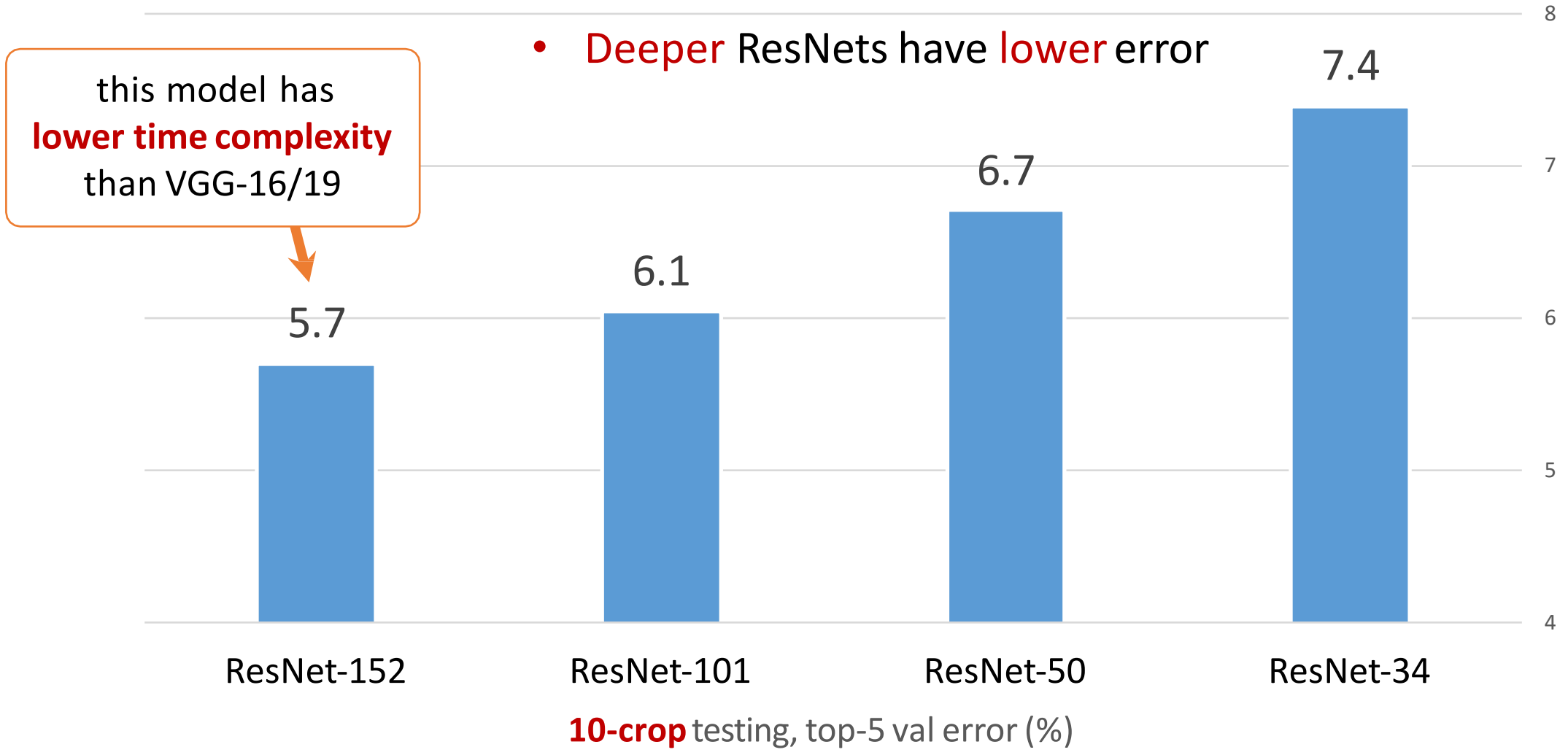
- Deep ResNets can be trained without difficulties
- Deeper ResNets have **lower training error**, and also lower test error

ImageNet experiments



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

ImageNet experiments



Beyond classification

A treasure from ImageNet is on learning features.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. arXiv 2015.

"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

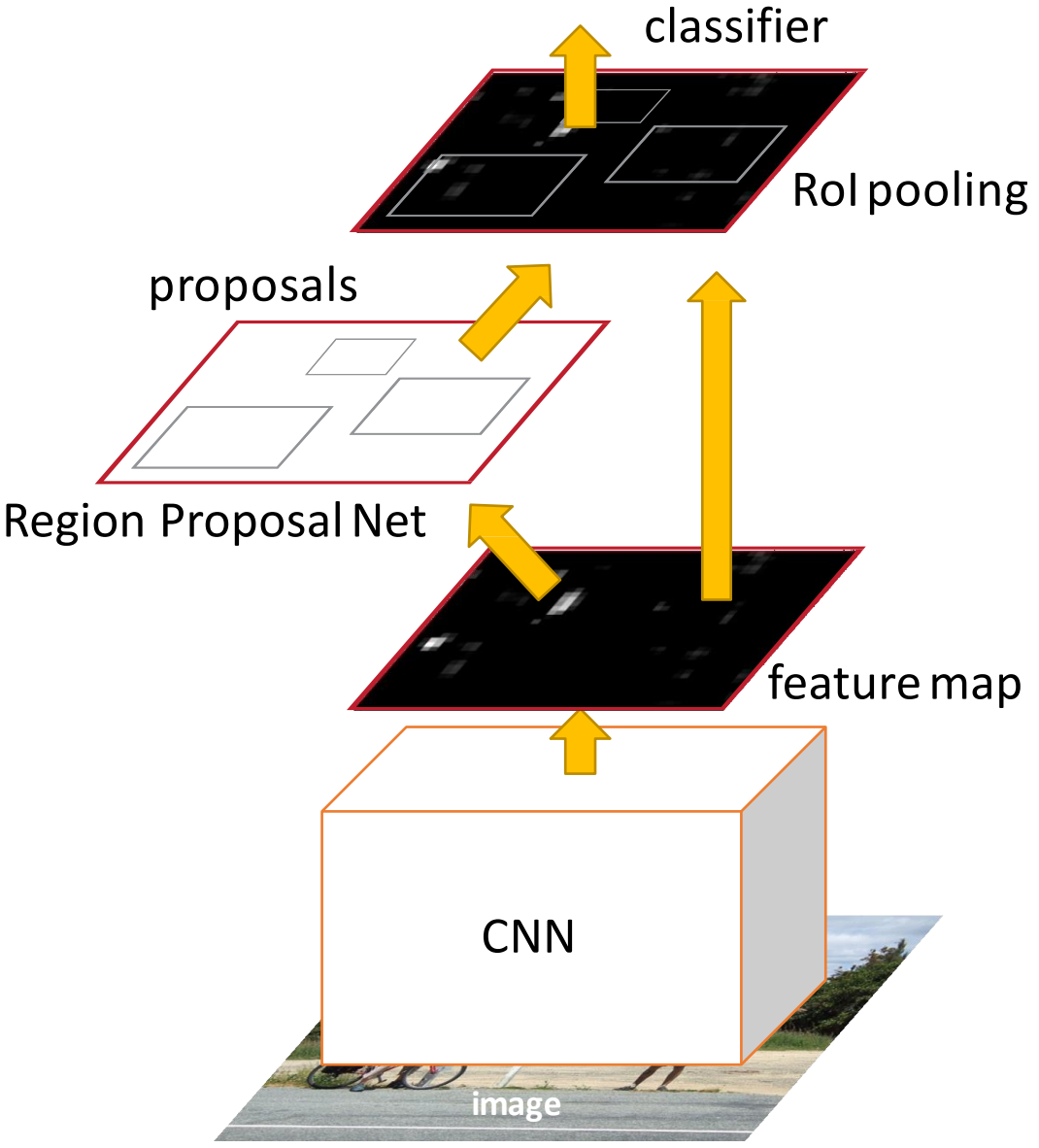
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. "Deep Residual Learning for Image Recognition". CVPR 2016.

Object Detection (brief)

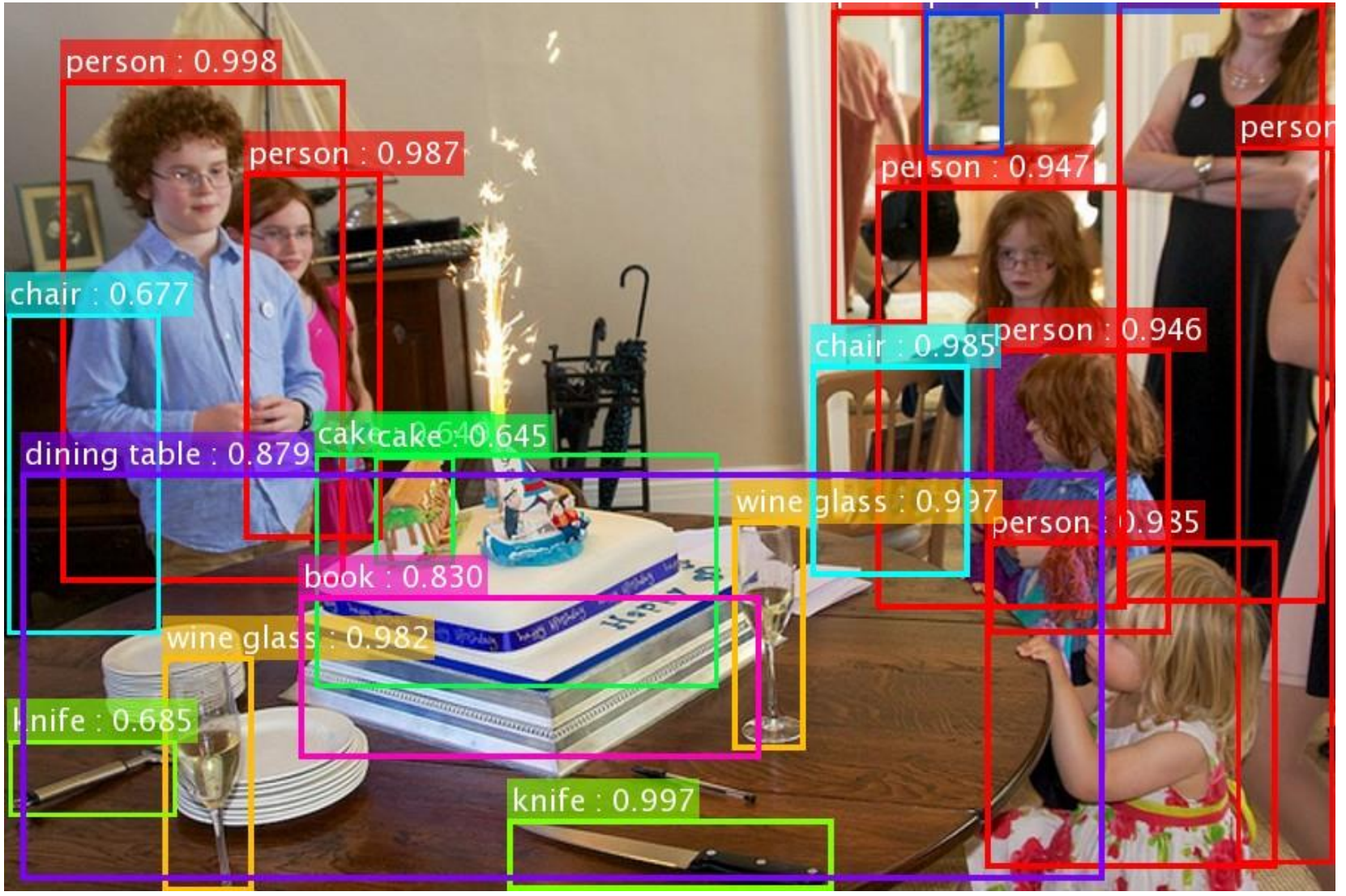
- Simply "Faster R-CNN + ResNet"

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

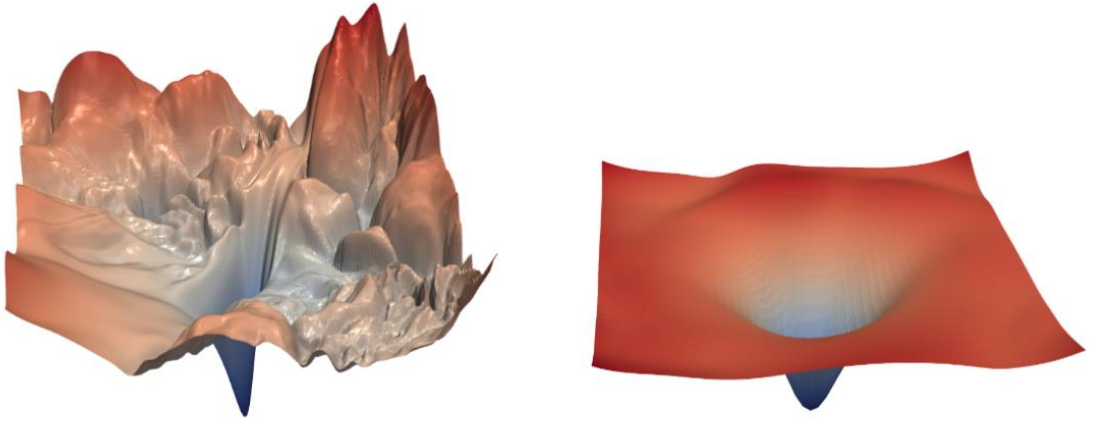
Kaiming He, Xiangyu Zhang, Shaoqing Ren, & Jian Sun. “Deep Residual Learning for Image Recognition”. CVPR 2016.
Shaoqing Ren, Kaiming He, Ross Girshick, & Jian Sun. “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”. NIPS 2015.

Why does ResNet work so well?

- The architecture is somehow easier to optimize.
- The authors argue it probably isn't because it solves the "vanishing gradient" problem.
- While the gradients might not be "vanishing" in "plain" nets, they don't seem as stable and trustworthy, according to follow up work, e.g.

Visualizing the Loss Landscape of Neural Nets. Hao Li, Zheng Xu , Gavin Taylor, Christoph Studer, Tom Goldstein. NeurIPS 2018.

We argue that this optimization difficulty is *unlikely* to be caused by vanishing gradients. These plain networks are trained with BN [16], which ensures forward propagated signals to have non-zero variances. We also verify that the backward propagated gradients exhibit healthy norms with BN. So neither forward nor backward signals vanish. In



(a) without skip connections (b) with skip connections

Figure 1: The loss surfaces of ResNet-56 with/without skip connections. The proposed filter normalization scheme is used to enable comparisons of sharpness/flatness between the two figures.

Semantic Segmentation

Project 4

Dataset

The dataset to be used in this assignment is the Camvid dataset, a small dataset of 701 images for self-driving perception. It was first introduced in 2008 by researchers at the University of Cambridge [1]. You can read more about it at the original dataset page or in the paper describing it. The images have a typical size of around 720 by 960 pixels. We'll downsample them for training though since even at 240 x 320 px, most of the scene detail is still recognizable.

Today there are much larger semantic segmentation datasets for self-driving, like Cityscapes, WildDashV2, Audi A2D2, but they are too large to work with for a homework assignment.

The original Camvid dataset has 32 ground truth semantic categories, but most evaluate on just an 11-class subset, so we'll do the same. These 11 classes are 'Building', 'Tree', 'Sky', 'Car', 'SignSymbol', 'Road', 'Pedestrian', 'Fence', 'Column_Pole', Sidewalk', 'Bicyclist'. A sample collection of the Camvid images can be found below:

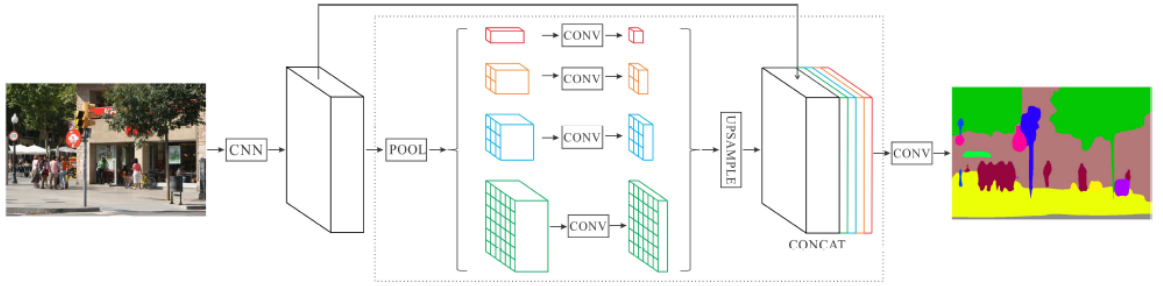
(a) Image A, RGB (b) Image A, Ground Truth (c) Image B, RGB (d) Image B, Ground Truth

Figure 2: Example scenes from the Camvid dataset. The RGB image is shown on the left, and the corresponding ground truth "label map" is shown on the right.

1 Implementation

For this project, the majority of the details will be provided into two separate Jupyter notebooks. The first, `proj4_local.ipynb` includes unit tests to help guide you with local implementation. After finishing that, upload `proj4_colab.ipynb` to Colab. Next, zip up the files for Colab with our script `zip_for_colab.py`, and upload these to your Colab environment.

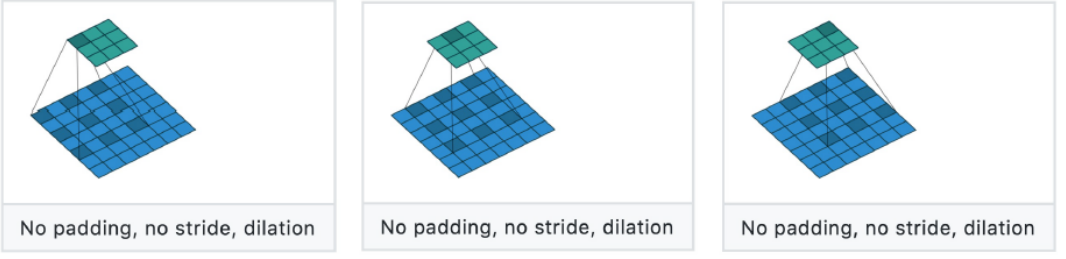
We will be implementing the PSPNet [3] architecture. You can read the original paper here. This network uses a ResNet [2] backbone, but uses *dilation* to increase the receptive field, and aggregates context over different portions of the image with a "Pyramid Pooling Module" (PPM).

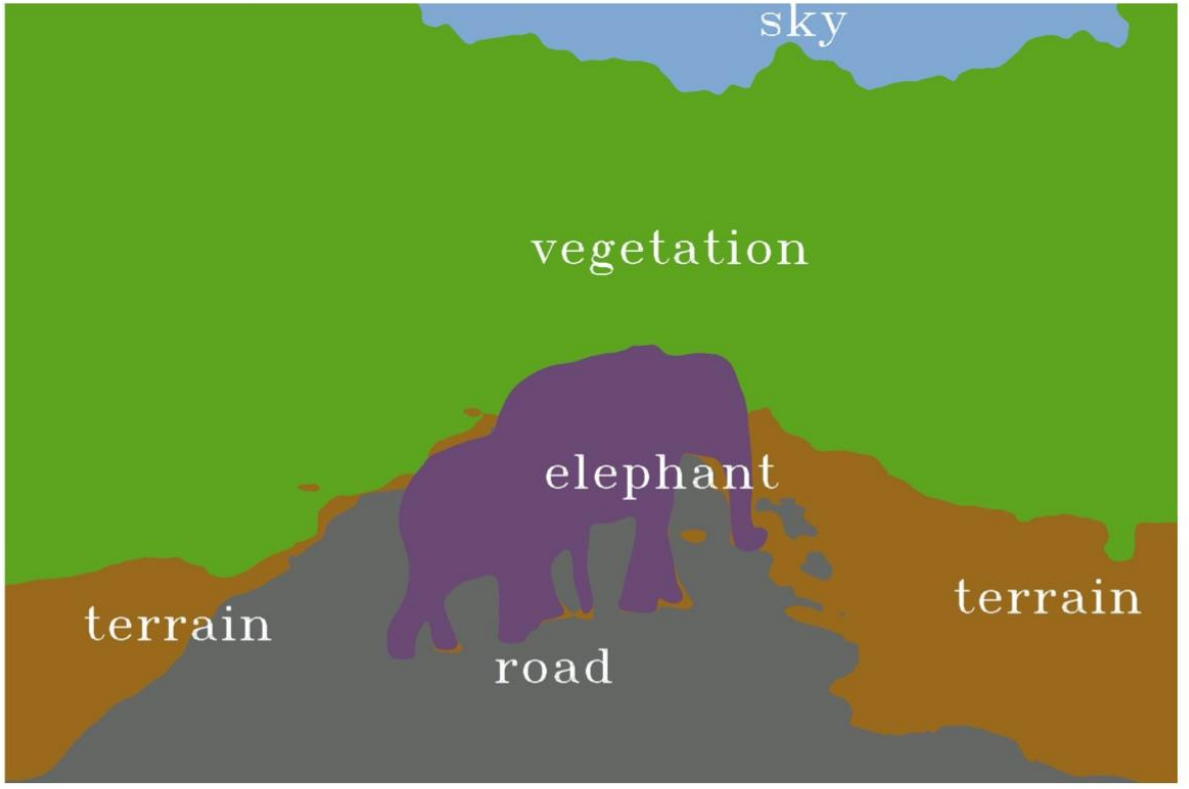


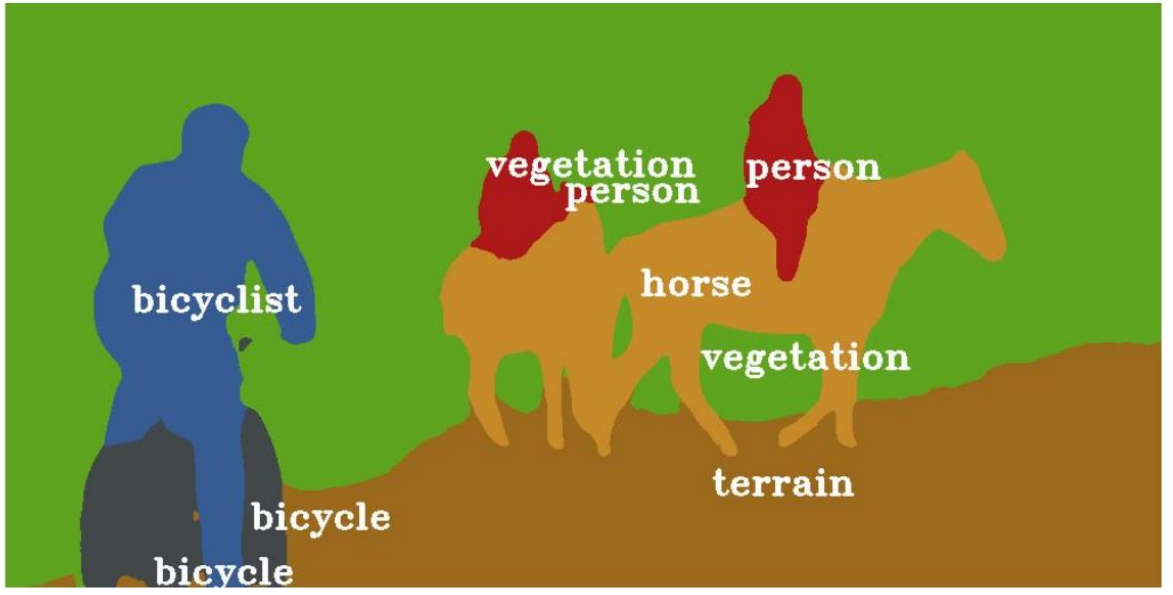
(a) Input Image (b) Feature Map (c) Pyramid Pooling Module (d) Final Prediction

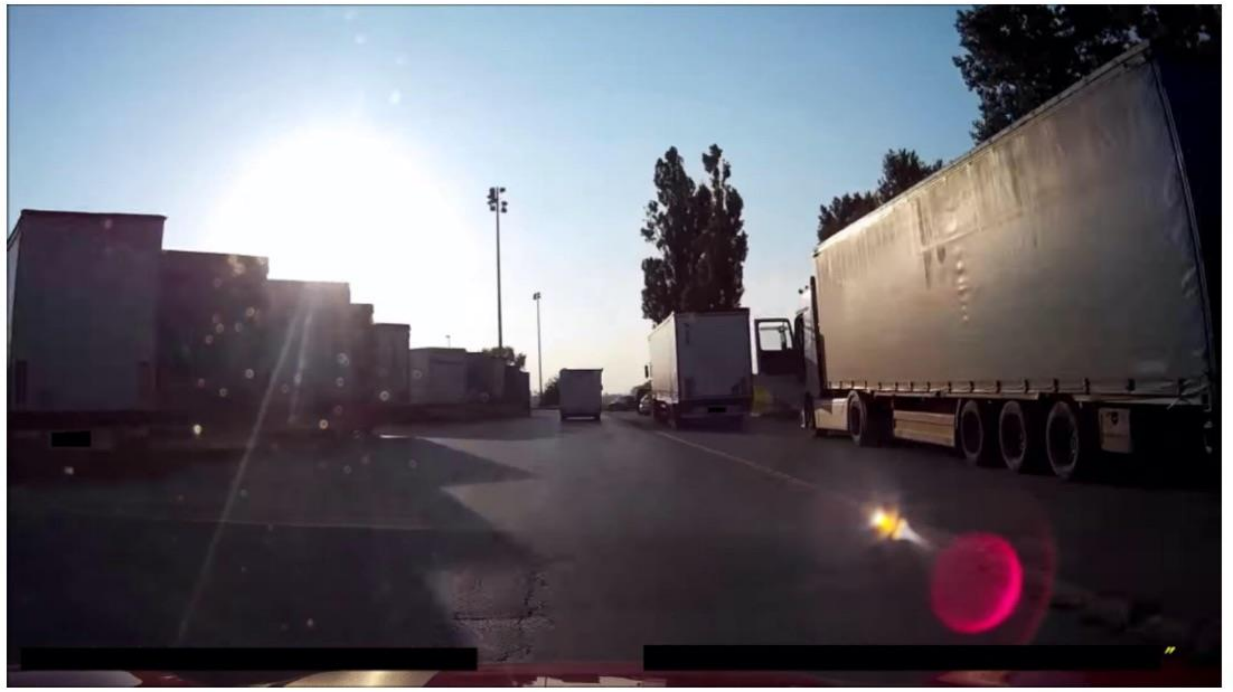
Figure 3: PSPNet architecture. The Pyramid Pooling Module (PPM) splits the $H \times W$ feature map into KxK grids. Here, 1×1, 2×2, 3×3, and 6×6 grids are formed, and features are average-pooled within each grid cell. Afterwards, the 1×1, 2×2, 3×3, and 6×6 grids are upsampled back to the original $H \times W$ feature map resolution, and are stacked together along the channel dimension.

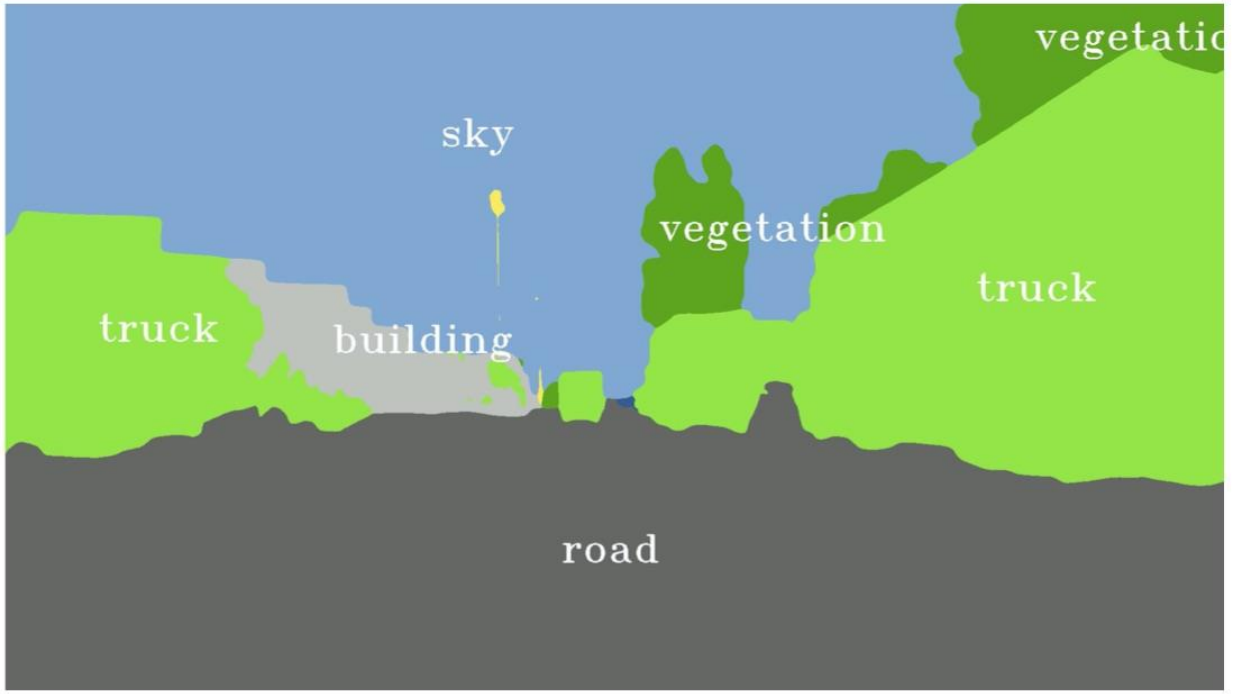
You can read more about dilated convolution in the Dilated Residual Network here, which PSPNet takes some ideas from. Also, you can watch a helpful animation about dilated convolution here.

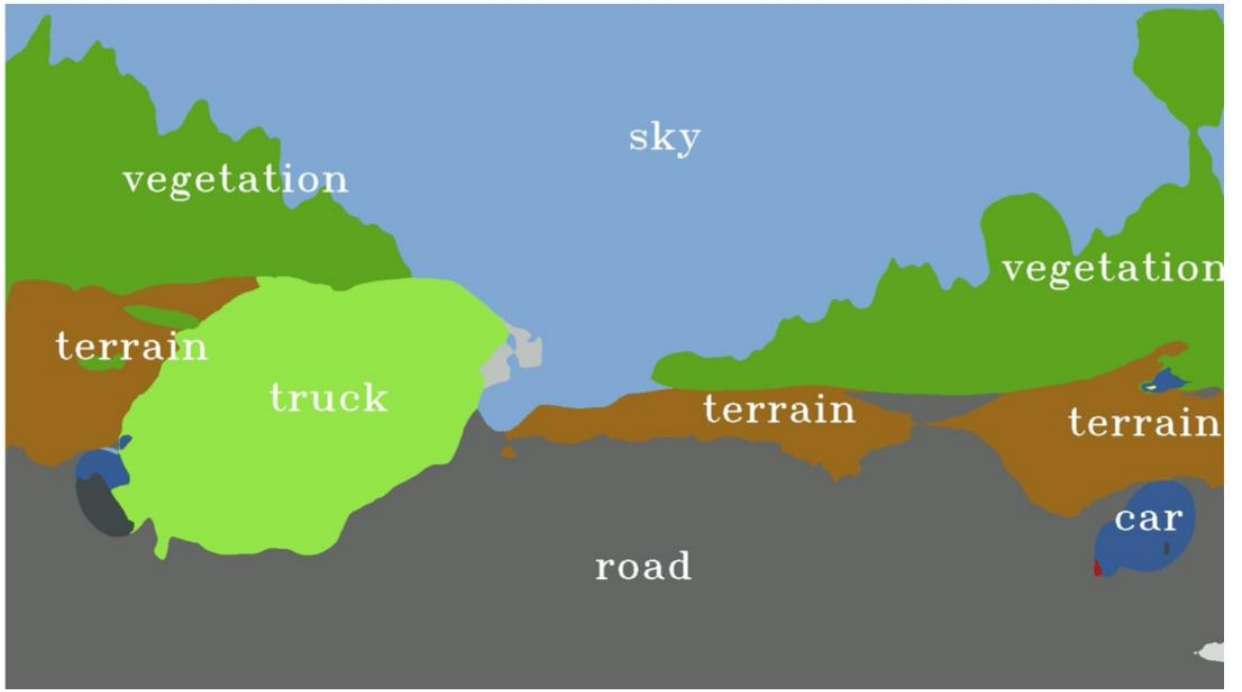


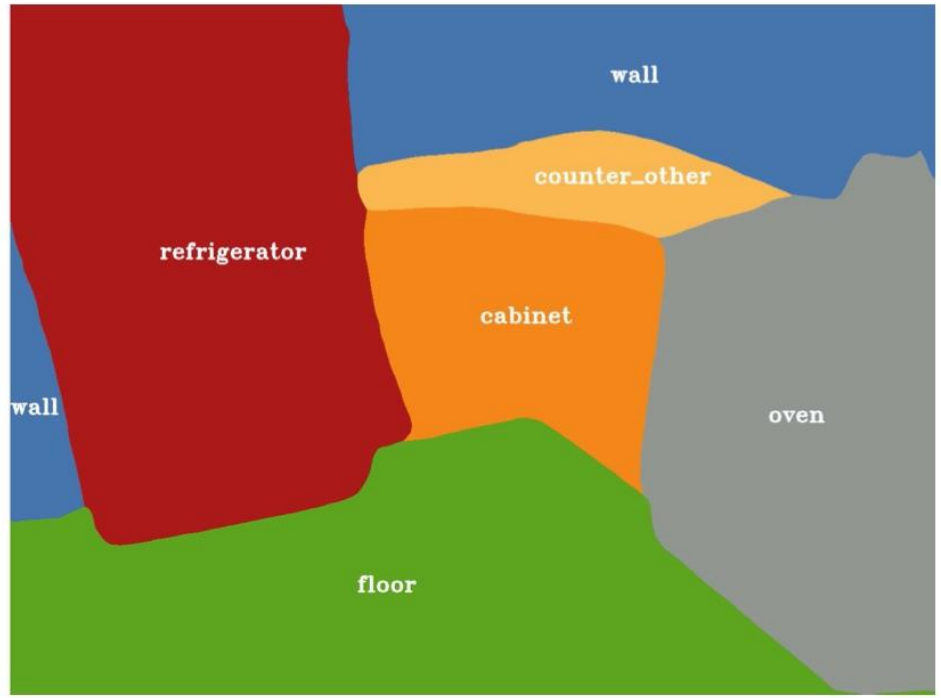

sky
vegetation
elephant
terrain
terrain
road


vegetation
person
person
bicyclist
horse
vegetation
terrain
bicycle
bicycle



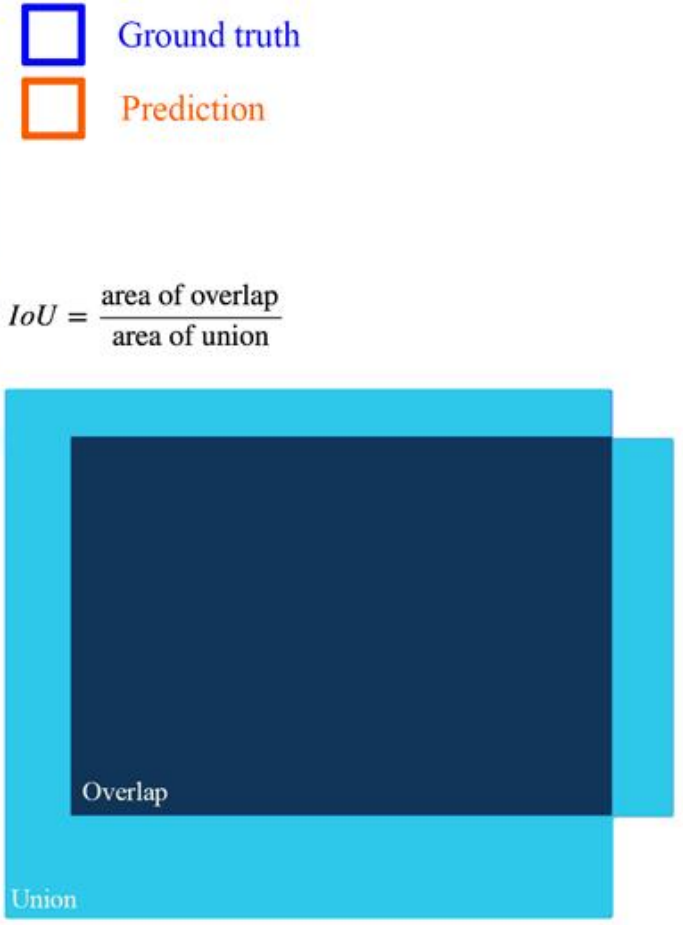

vegetatio
sky
vegetation
truck
truck
building
road


sky
vegetation
vegetation
terrain
truck
terrain
terrain
car
road


wall
counter_other
refrigerator
cabinet
wall
oven
floor


cabinet
cabinet
paper
wall
paper
vegetation
sink
counter_other
wall
cabinet

Measuring Performance: Intersection over Union



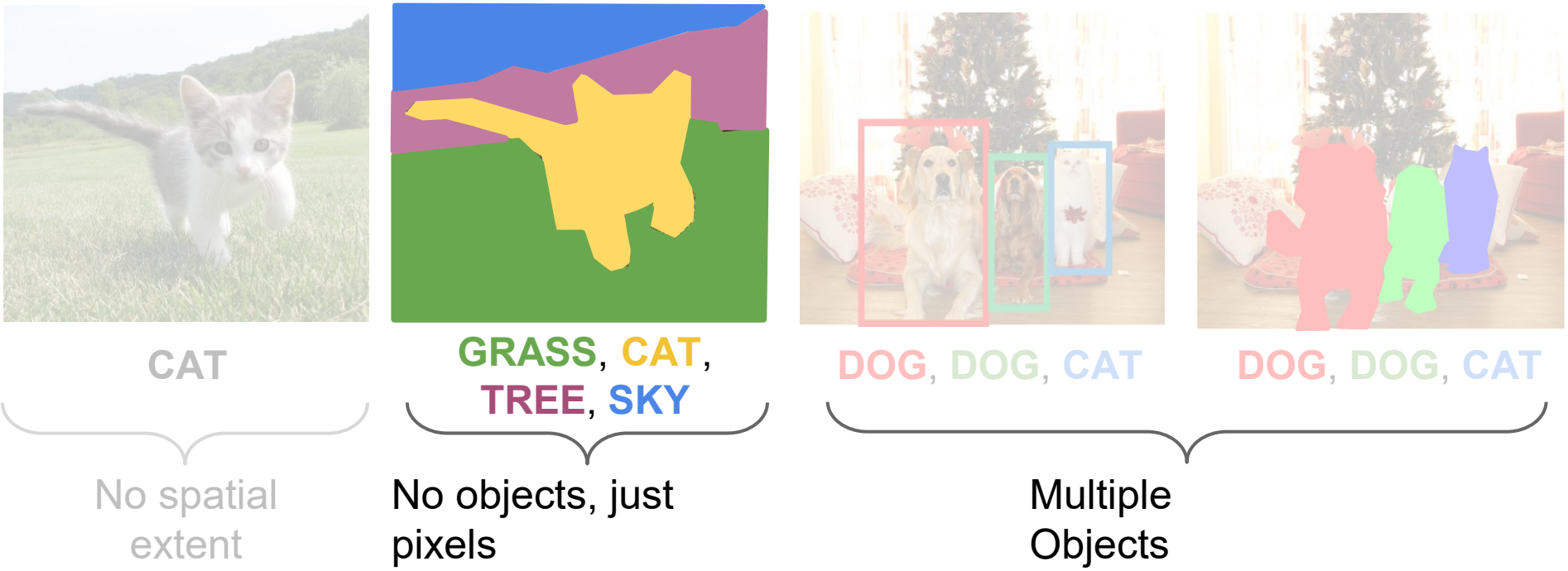
Applies to segmentations, as well

Figure source: http://cs230.stanford.edu/section/8/

Figure source: https://www.pinterest.com/pin/457959855830667185/

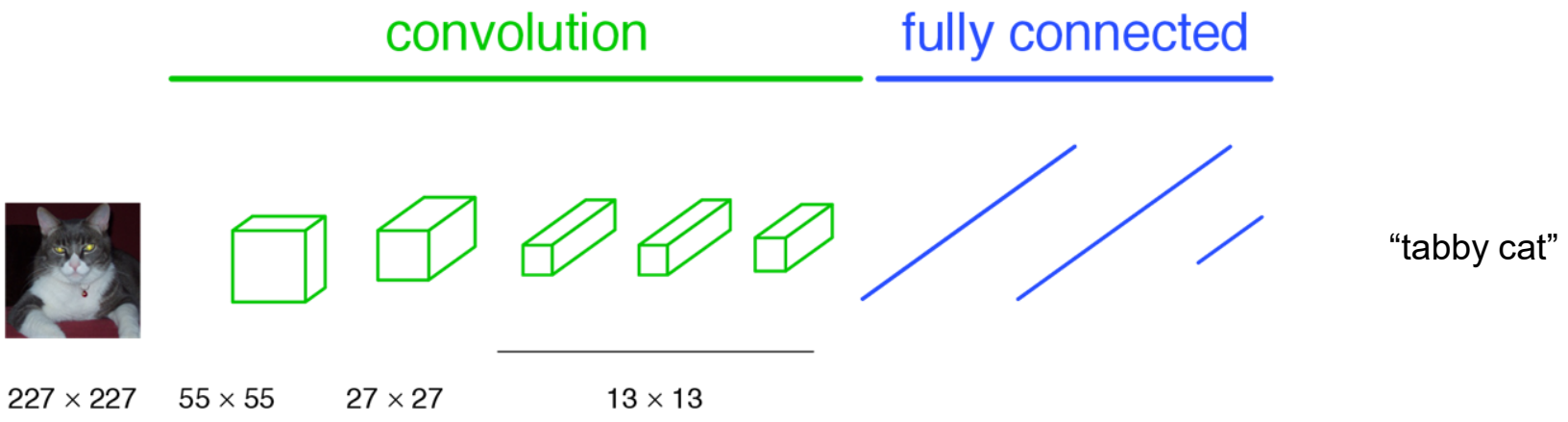
Figure source: https://www.gettyimages.com/photos/moss-rock?phrase=moss%20rock&sort=mostpopular

Tasks: Semantic Segmentation



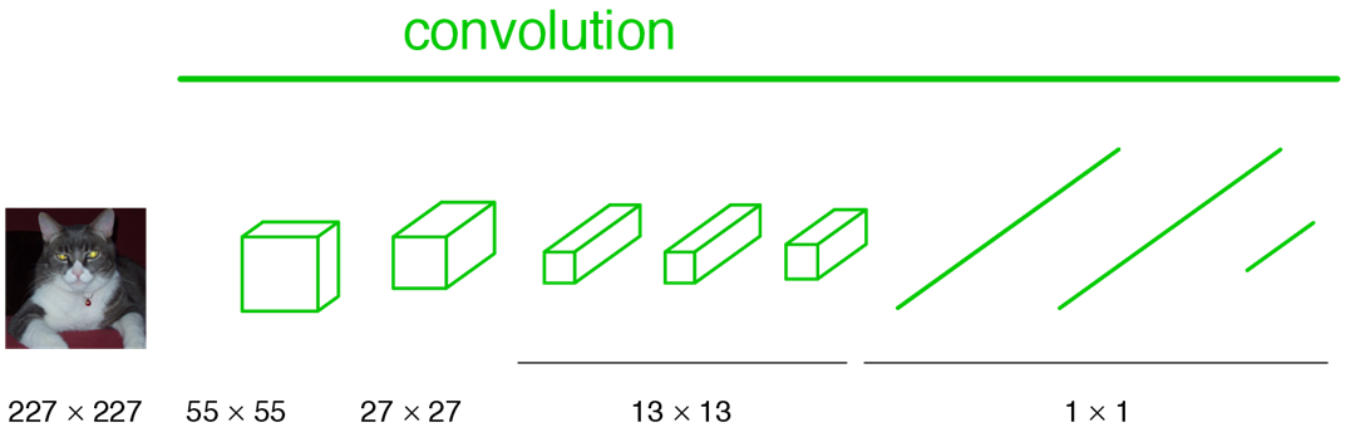
Slide Credit: Justin Johnson and David Fouhey

a classification network



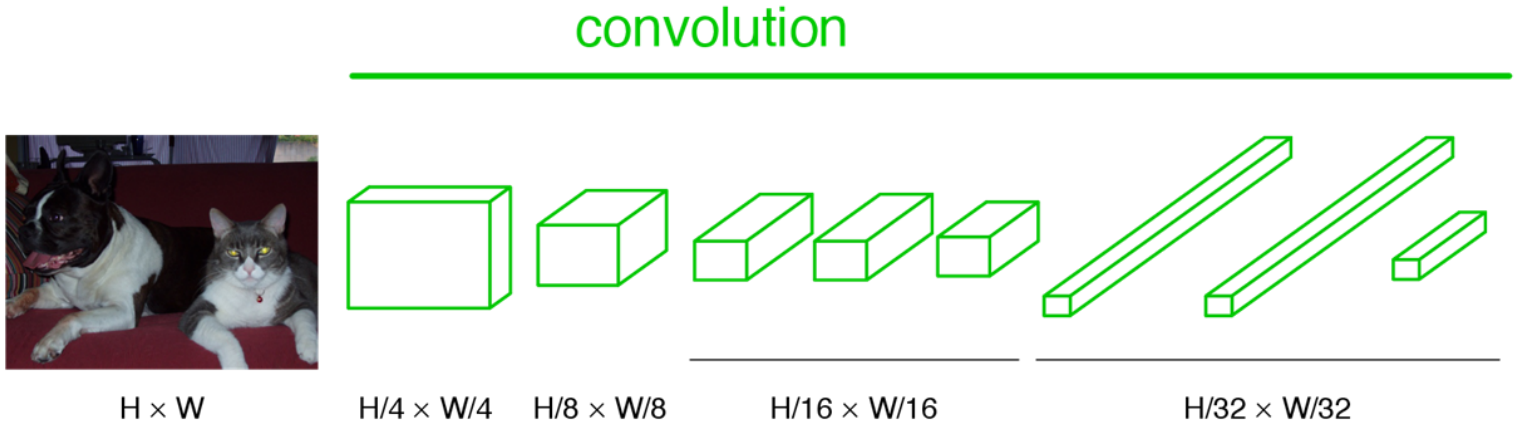
Fully Convolutional Networks for Semantic Segmentation.
Jon Long, Evan Shelhamer, Trevor Darrell. CVPR 2015

becoming fully convolutional

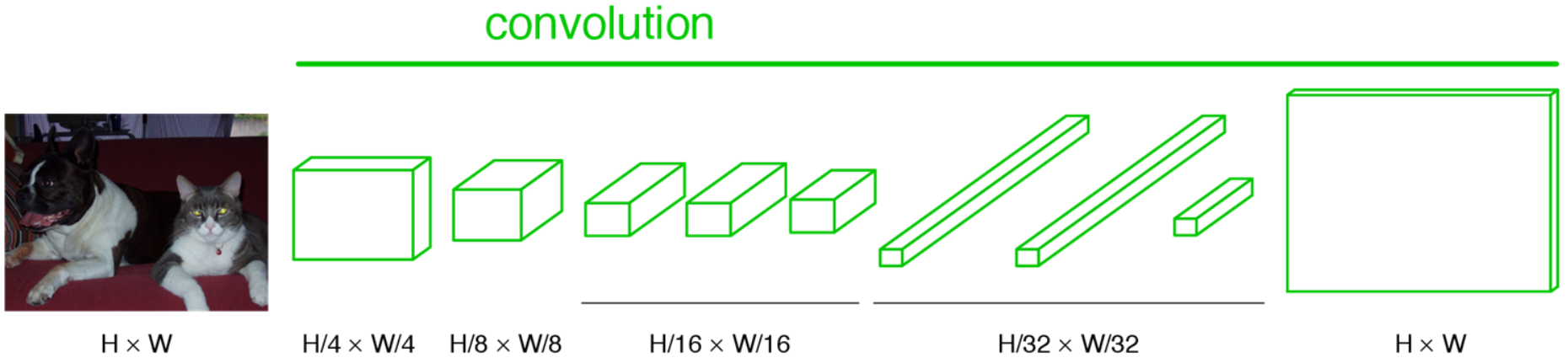


Note: “Fully Convolutional” and “Fully Connected” aren’t the same thing.
They’re almost opposites, in fact.

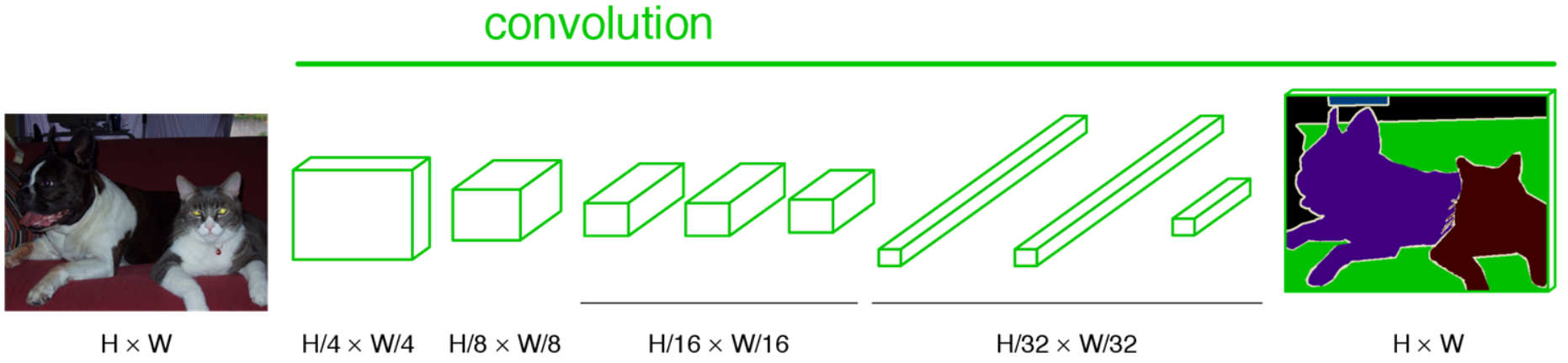
becoming fully convolutional



upsampling output

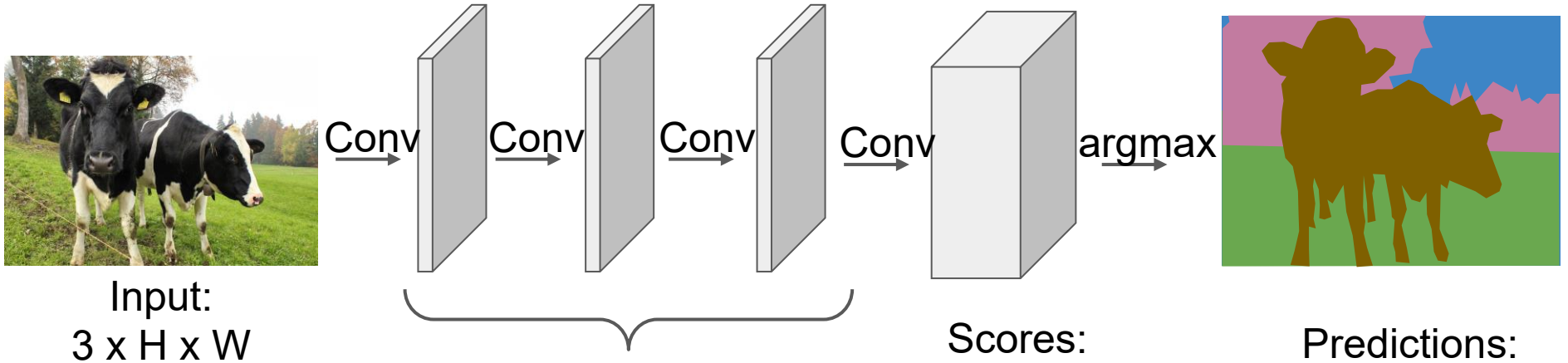


end-to-end, pixels-to-pixels network



Fully Convolutional Network

Design a network as a bunch of convolutional layers to make predictions for pixels all at once!



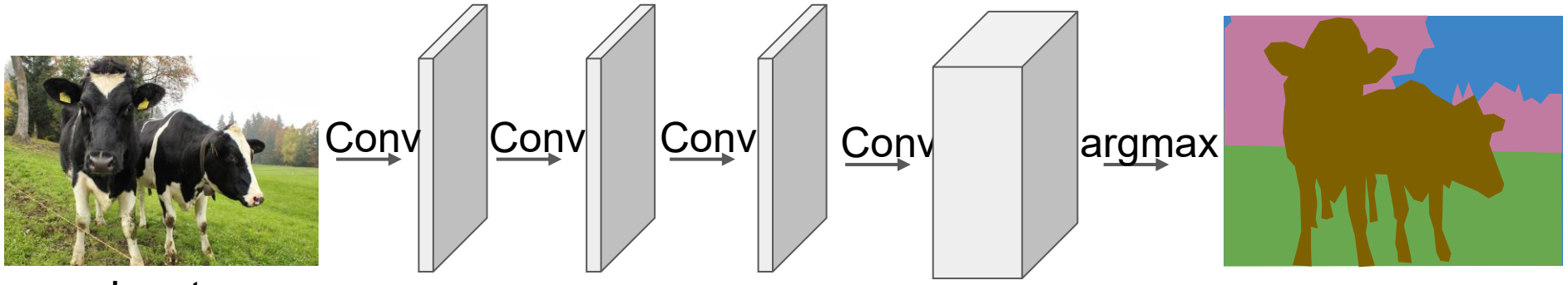
Loss function: Per-Pixel cross-entropy

Long et al, “Fully convolutional networks for semantic segmentation”, CVPR 2015

Slide Credit: Justin Johnson and David Fouhey

Fully Convolutional Network

Design a network as a bunch of convolutional layers to make predictions for pixels all at once!



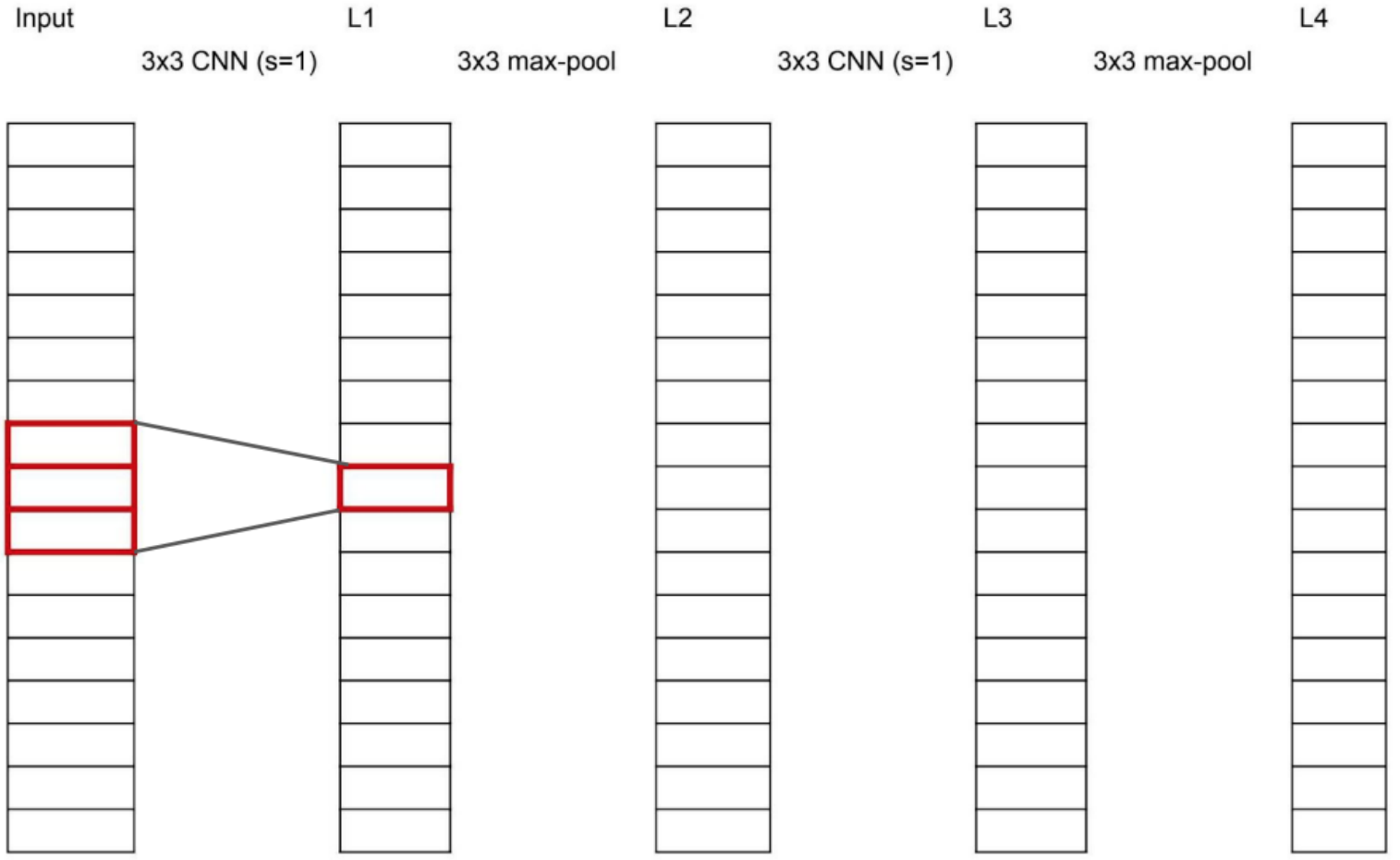
Input:
3 x H x W

Problem #1: Effective receptive field size is linear in number of conv layers: With L 3x3 conv layers, receptive field is 1+2L

Long et al, “Fully convolutional networks for semantic segmentation”, CVPR 2015

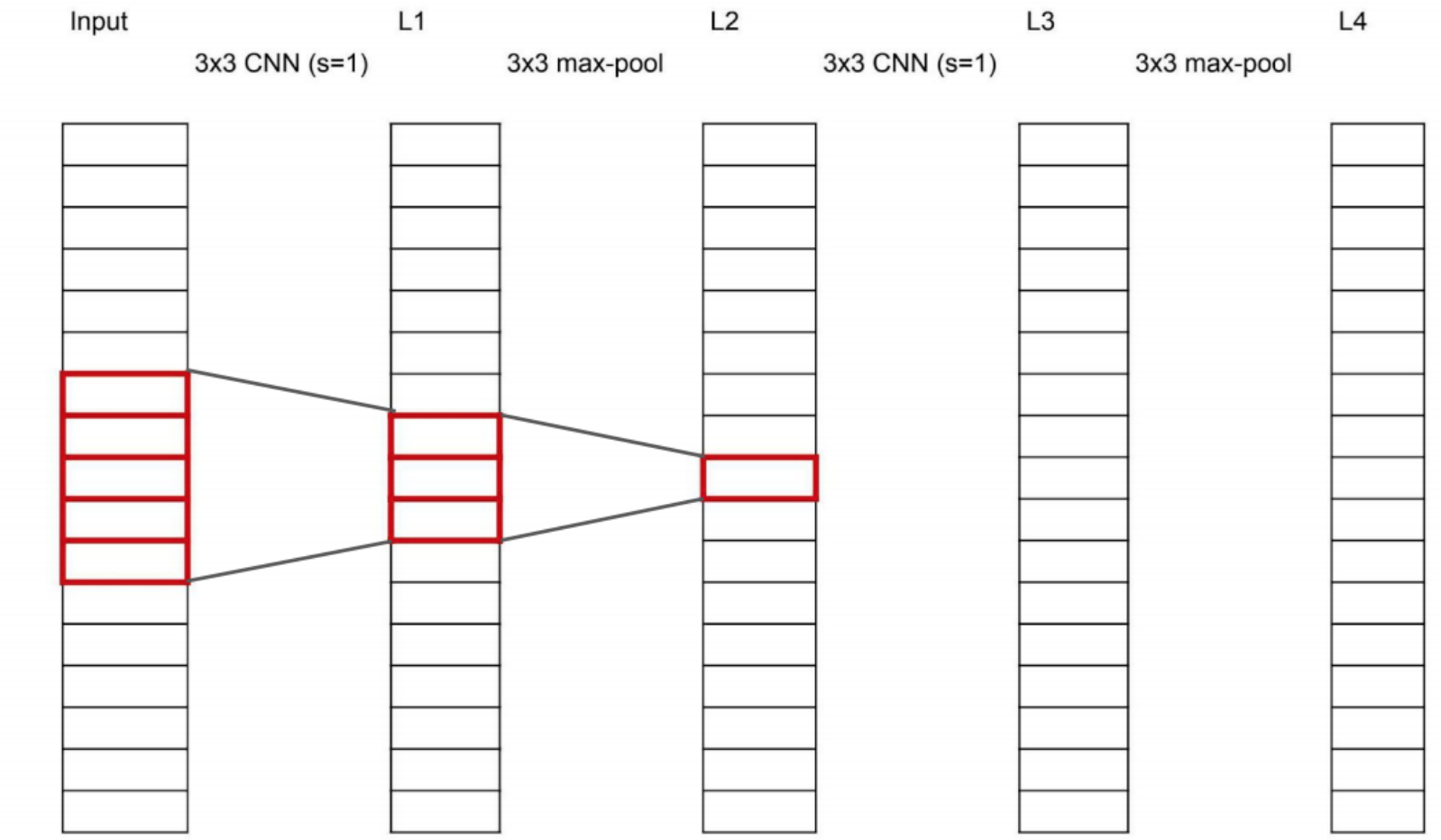
Slide Credit: Justin Johnson and David Fouhey

Receptive field



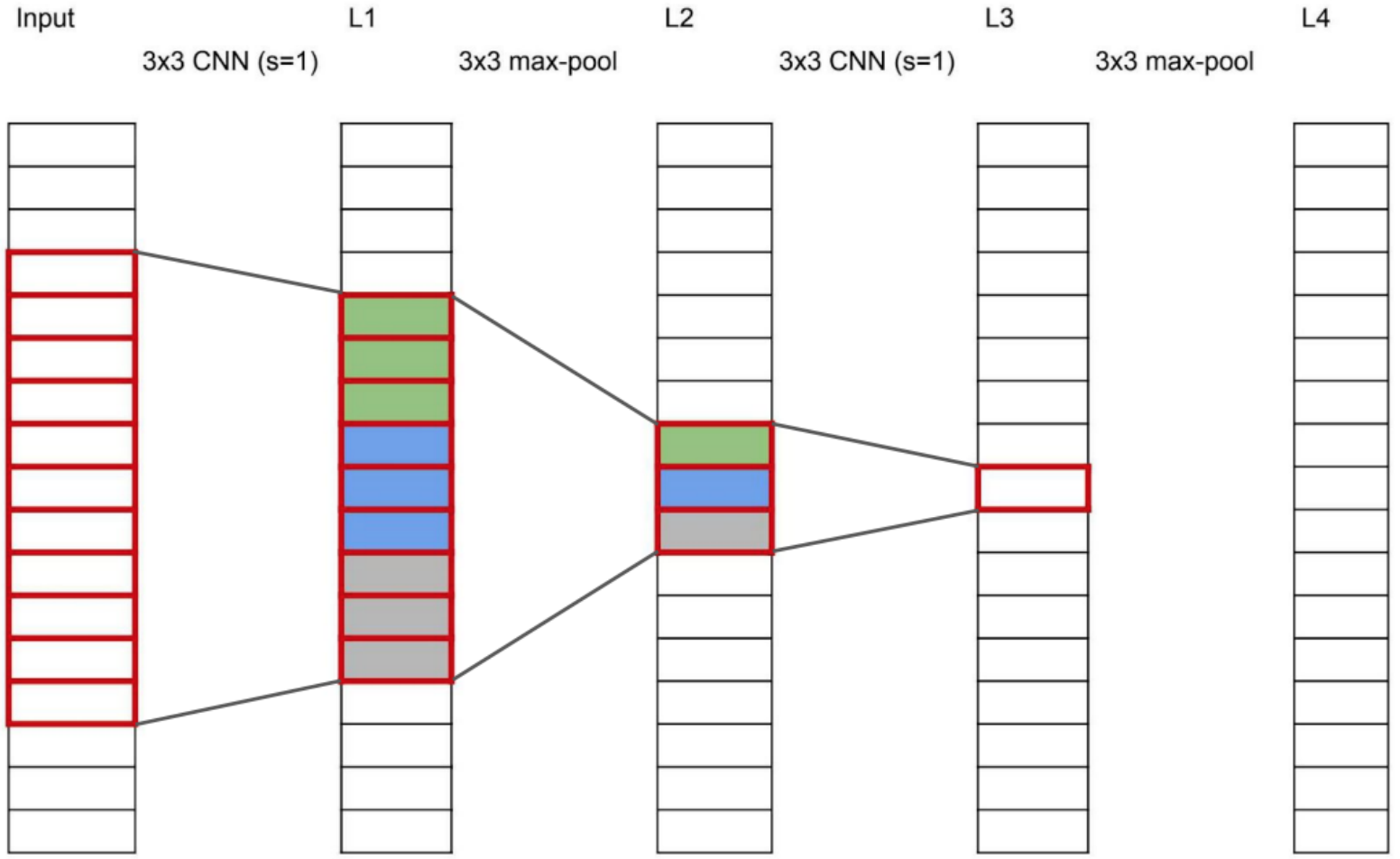
Slide Credit: Frank Dellaert https://dellaert.github.io/19F-4476/resources/receptiveField.pdf

Receptive field



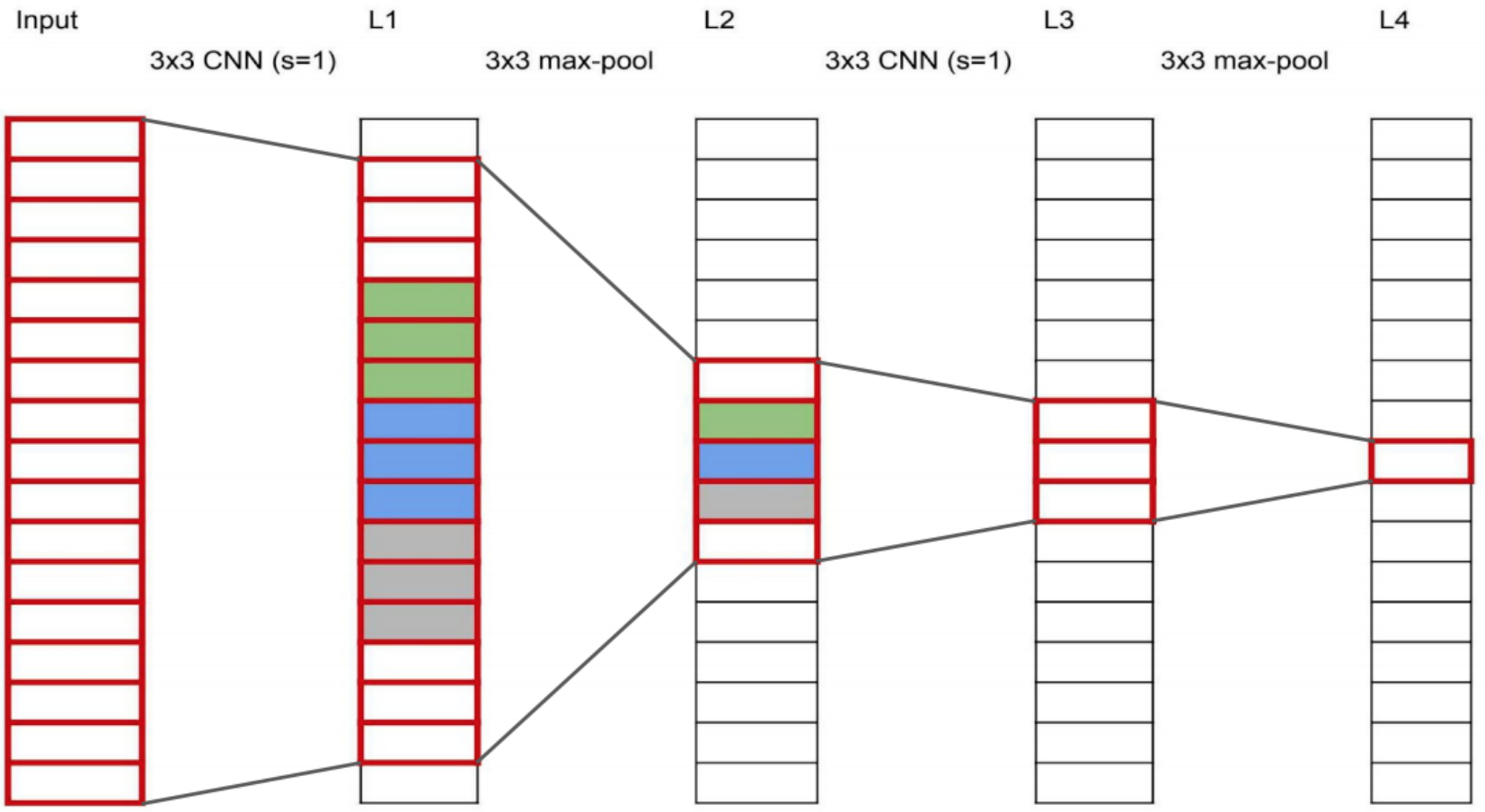
Slide Credit: Frank Dellaert https://dellaert.github.io/19F-4476/resources/receptiveField.pdf

Receptive field



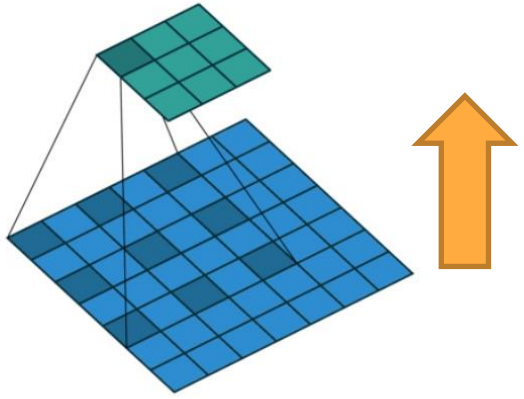
Slide Credit: Frank Dellaert https://dellaert.github.io/19F-4476/resources/receptiveField.pdf

Receptive field

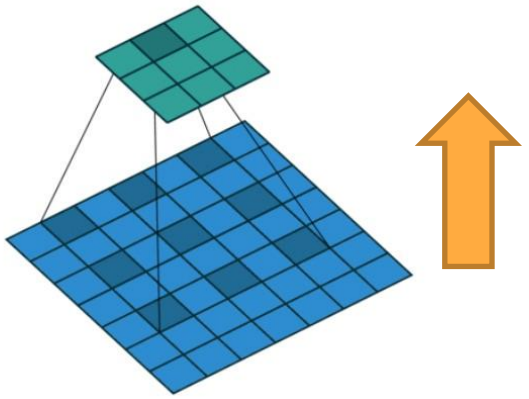


Slide Credit: Frank Dellaert https://dellaert.github.io/19F-4476/resources/receptiveField.pdf

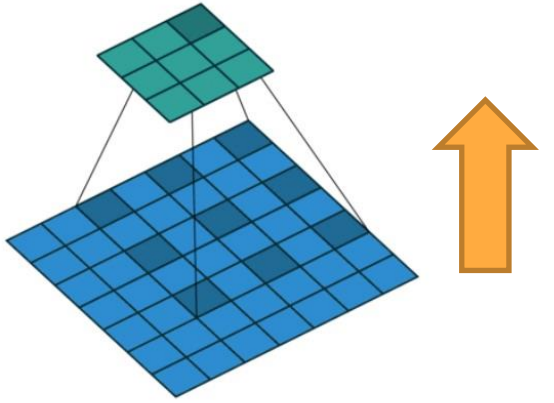
Dilated Convolution



No padding, no stride, dilation



No padding, no stride, dilation

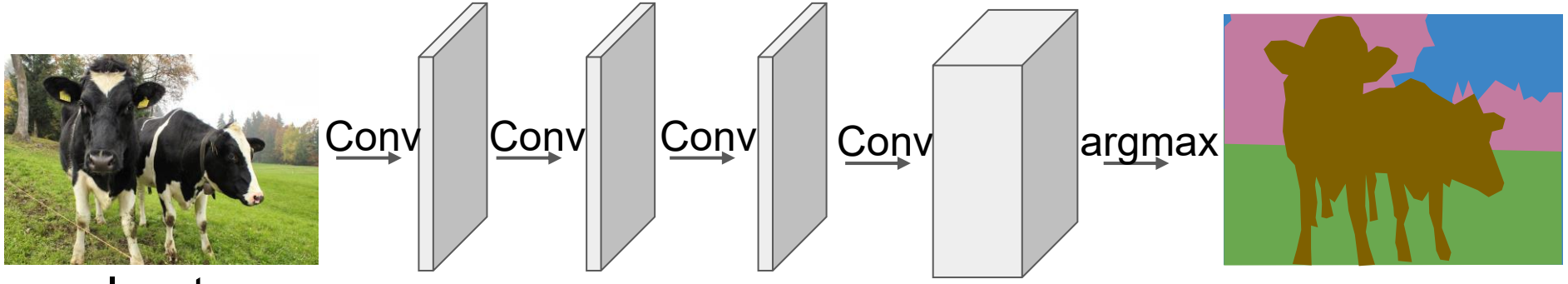


No padding, no stride, dilation

Figure source: https://github.com/vdumoulin/conv_arithmetic

Fully Convolutional Network

Design a network as a bunch of convolutional layers to make predictions for pixels all at once!



Input:
3 x H x W

Problem #1: Effective receptive field size is linear in number of conv layers: With L 3x3 conv layers, receptive field is 1+2L

Problem #2: Convolution on high res images is expensive!

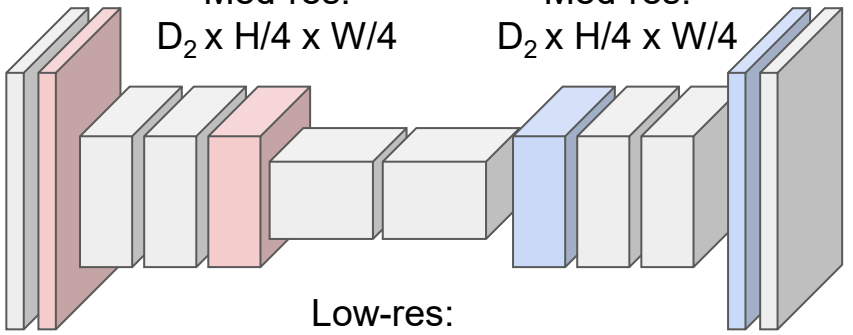
Long et al, "Fully convolutional networks for semantic segmentation", CVPR 2015

Slide Credit: Justin Johnson and David Fouhey

Fully Convolutional Network

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Input:
3 x H x W



Prediction
s:
H x W

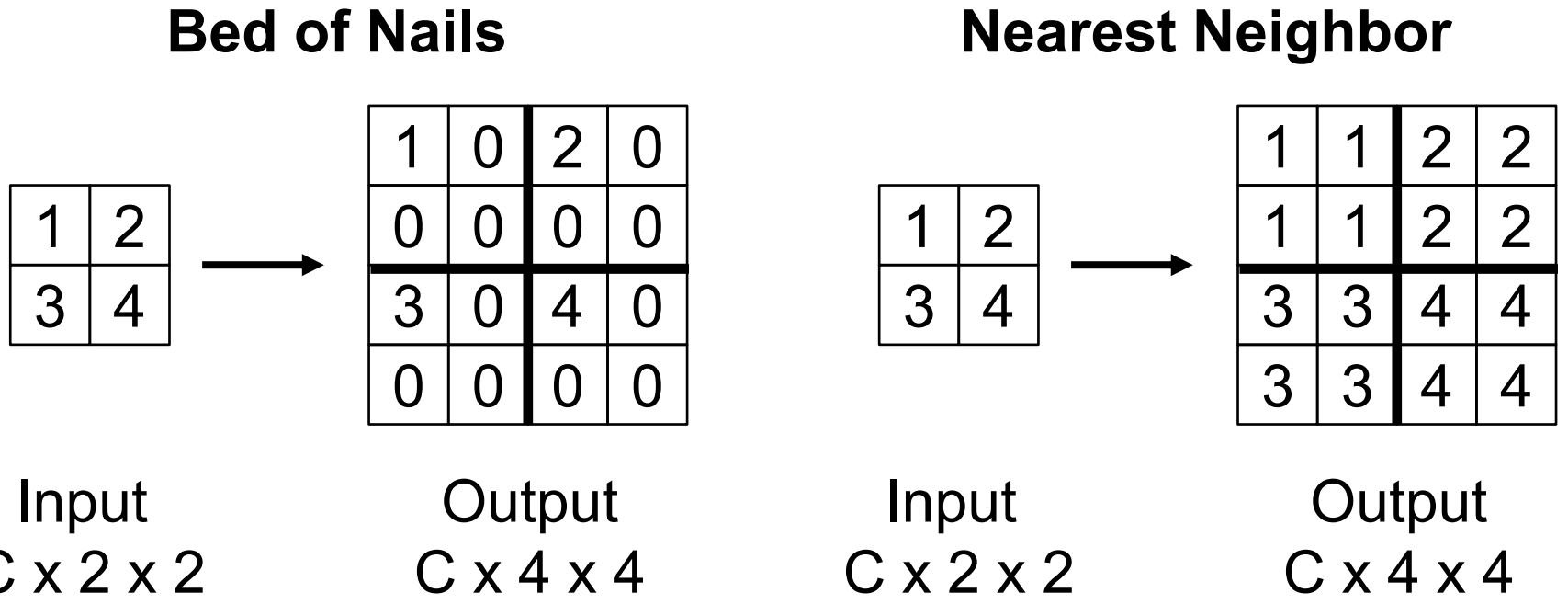
Downsampling:
Pooling, strided convolution

Upsampling:
???

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

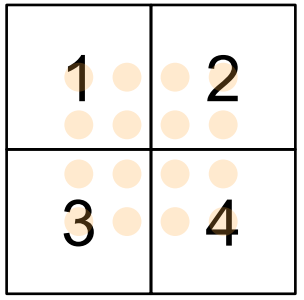
Slide Credit: Justin Johnson and David Fouhey

In-Network Upsampling: “Unpooling”



Slide Credit: Justin Johnson and David Fouhey

Upsampling: Bilinear Interpolation



1.00	1.25	1.75	2.00
1.50	1.75	2.25	2.50
2.50	2.75	3.25	3.50
3.00	3.25	3.75	4.00

Input: C x 2 x 2

Output: C x 4 x 4

$$f_{x,y} = \sum_{i,j} f_{i,j} \max(0, 1 - |x - i|) \max(0, 1 - |y - j|) \quad i \in \{\lfloor x \rfloor - 1, \ldots, \lceil x \rceil + 1\} \quad j \in \{\lfloor y \rfloor - 1, \ldots, \lceil y \rceil + 1\}$$

Use two closest neighbors in x and y to construct linear approximations

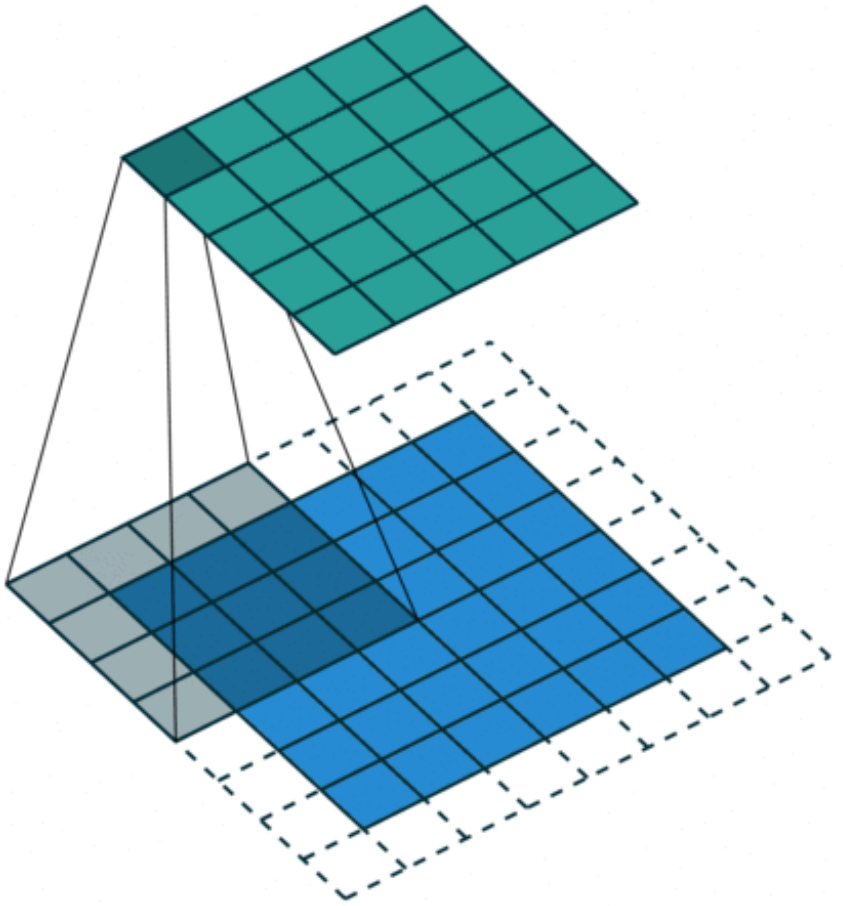
Slide Credit: Justin Johnson and David Fouhey

Upsampling: Transpose Convolution

Sometimes called "Deconvolution" but that is a problematic name

I like the term "broadcast" convolution

In this case, the filter is 4x4 and the outer boundary of the output is unused

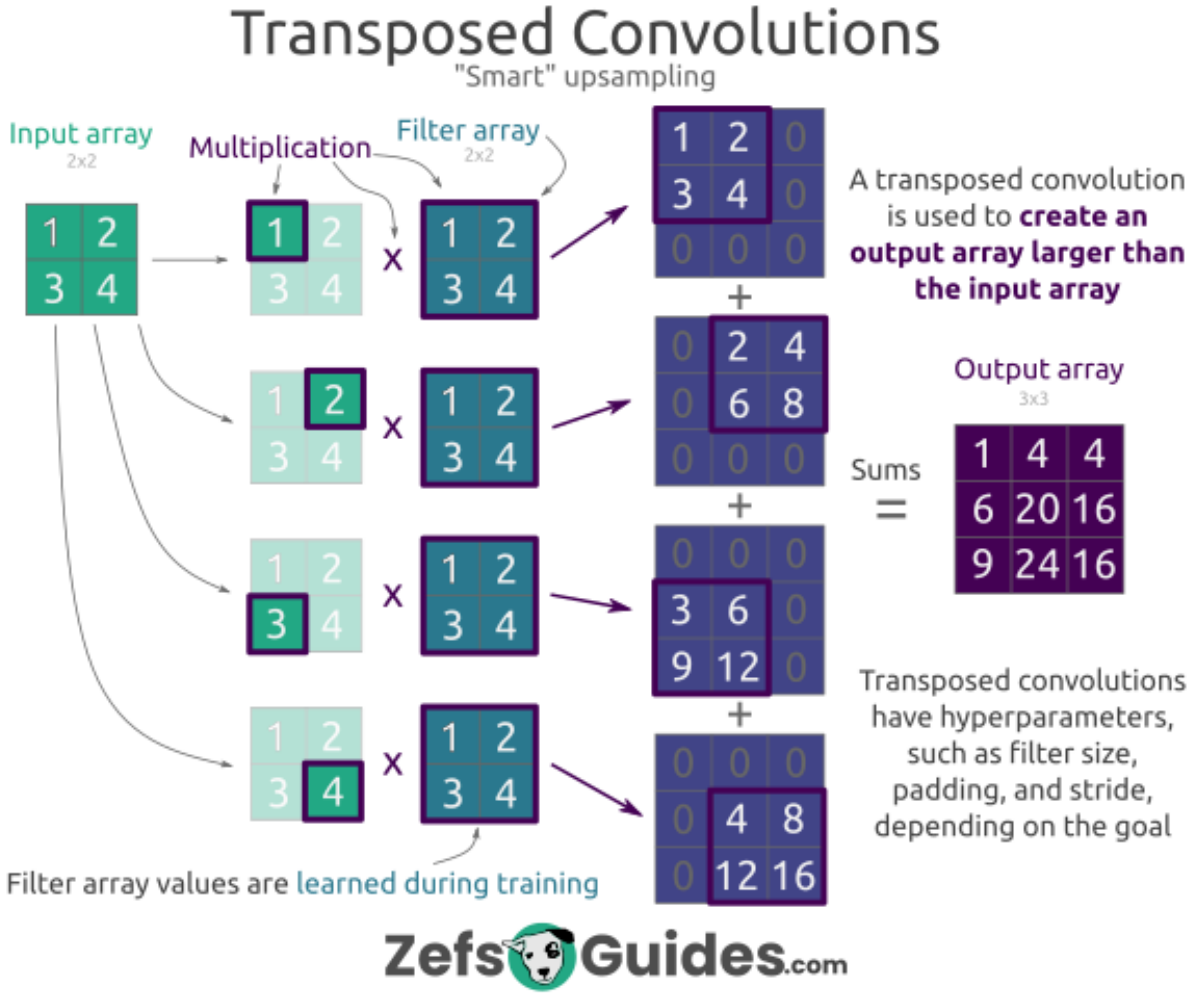


A guide to convolution arithmetic for deep learning
Vincent Dumoulin, Francesco Visin

Upsampling: Transpose Convolution

Sometimes called “Deconvolution” but that is a problematic name

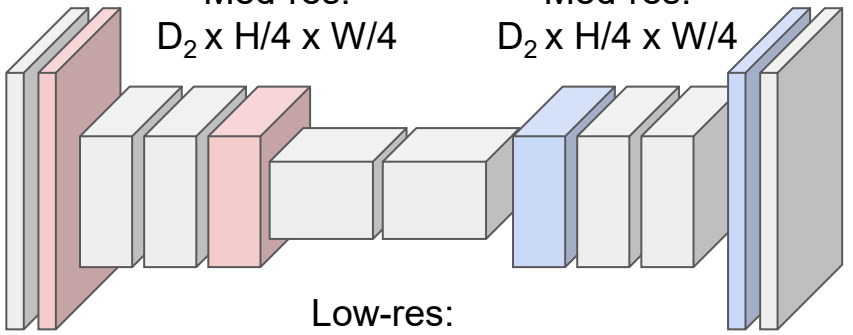
I like the term “broadcast” convolution



Fully Convolutional Network

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Input:
3 x H x W



Prediction
s:
H x W

Downsampling:
Pooling, strided convolution

Upsampling:
???

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Slide Credit: Justin Johnson and David Fouhey

PSPNet

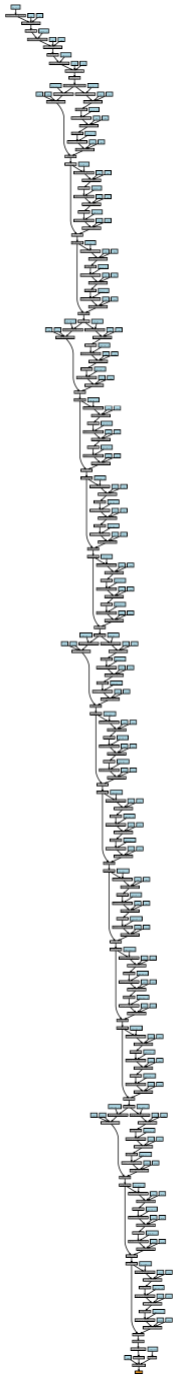
PSPNet uses a ResNet backbone

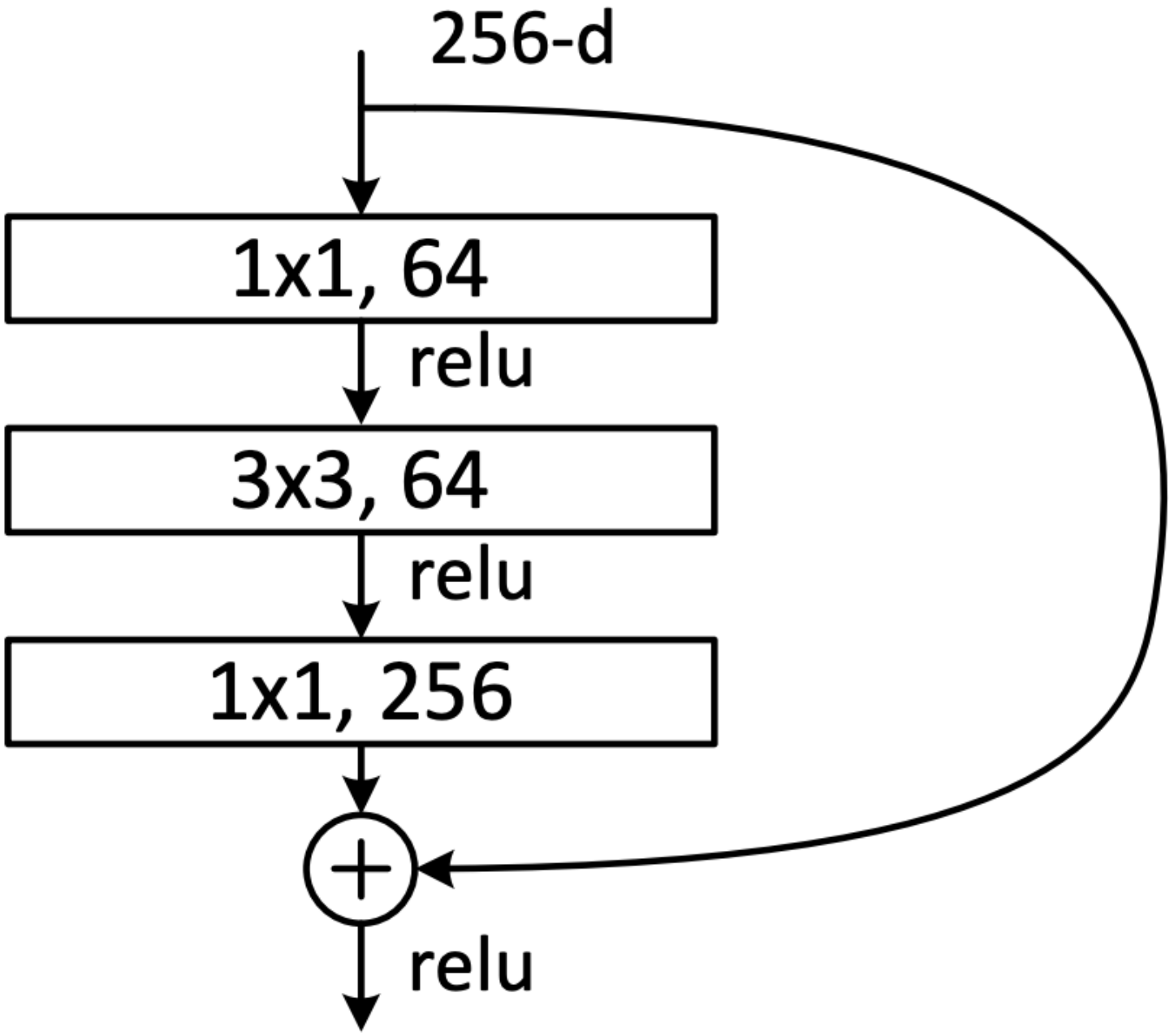
- 50, 101, or 152 Layers
- 50 Layers is already quite deep!

3.2. Pyramid Pooling Module

With above analysis, in what follows, we introduce the pyramid pooling module, which empirically proves to be an effective global contextual prior.

In a deep neural network, the size of receptive field can roughly indicates how much we use context information. Although theoretically the receptive field of ResNet [13] is already larger than the input image, it is shown by Zhou *et al*. [42] that the empirical receptive field of CNN is much smaller than the theoretical one especially on high-level layers. This makes many networks not sufficiently incorporate





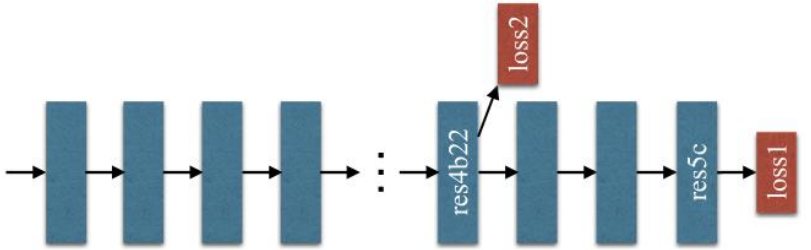
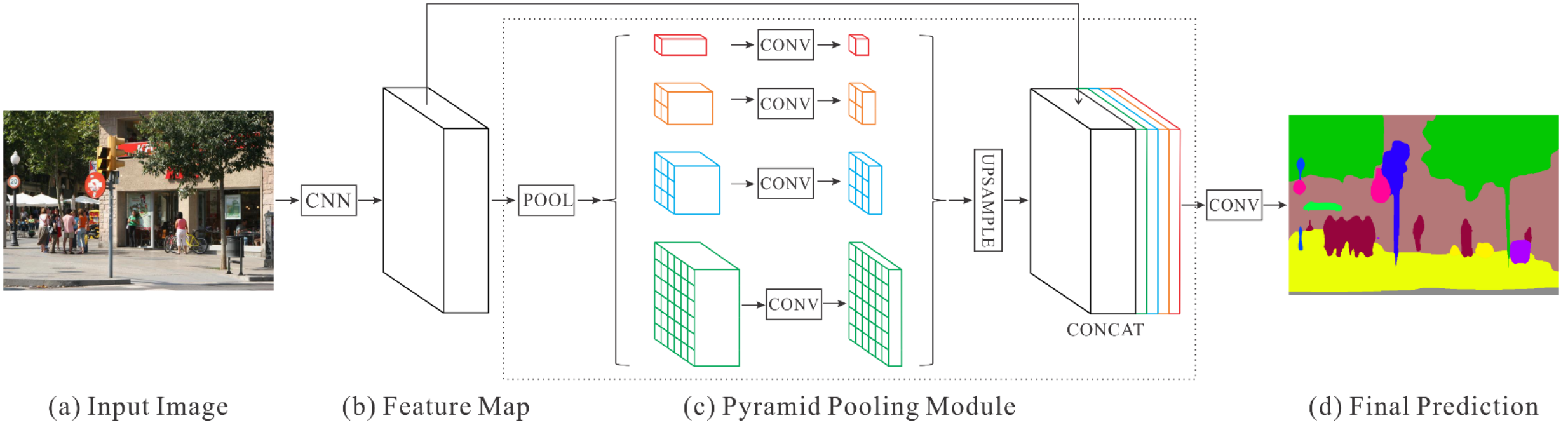


Figure 4. Illustration of auxiliary loss in ResNet101. Each blue box denotes a residue block. The auxiliary loss is added after the res4b22 residue block.

Pyramid Scene Parsing Network

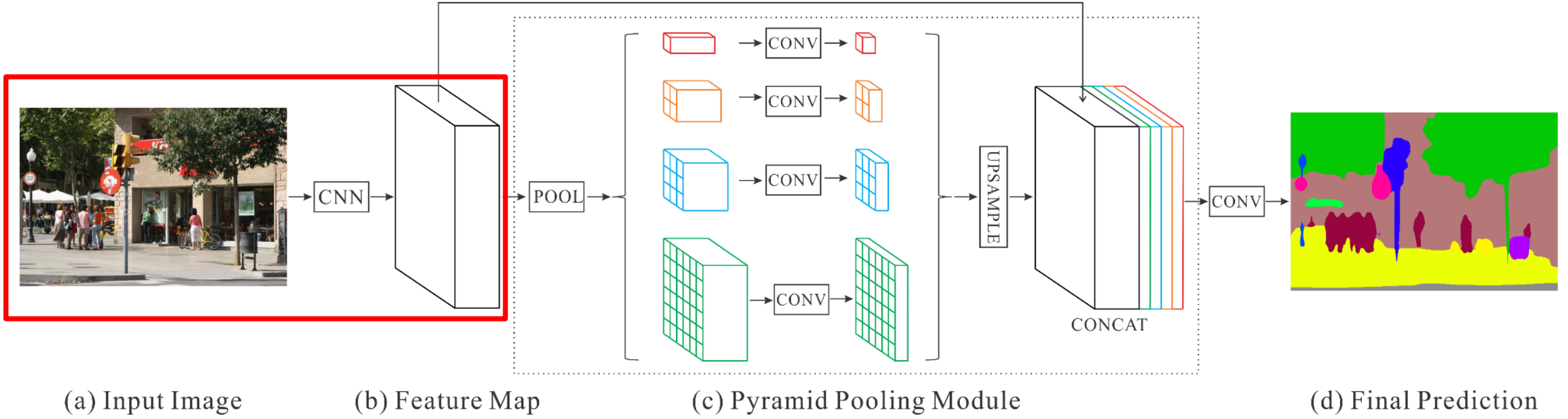


Framework overview of PSPNet

"Pyramid Scene Parsing Network", Zhao et al. CVPR 2017 [15,000+ citation]

Slide Credit: Hengshuang Zhao and Jiaya Jia

Pyramid Scene Parsing Network

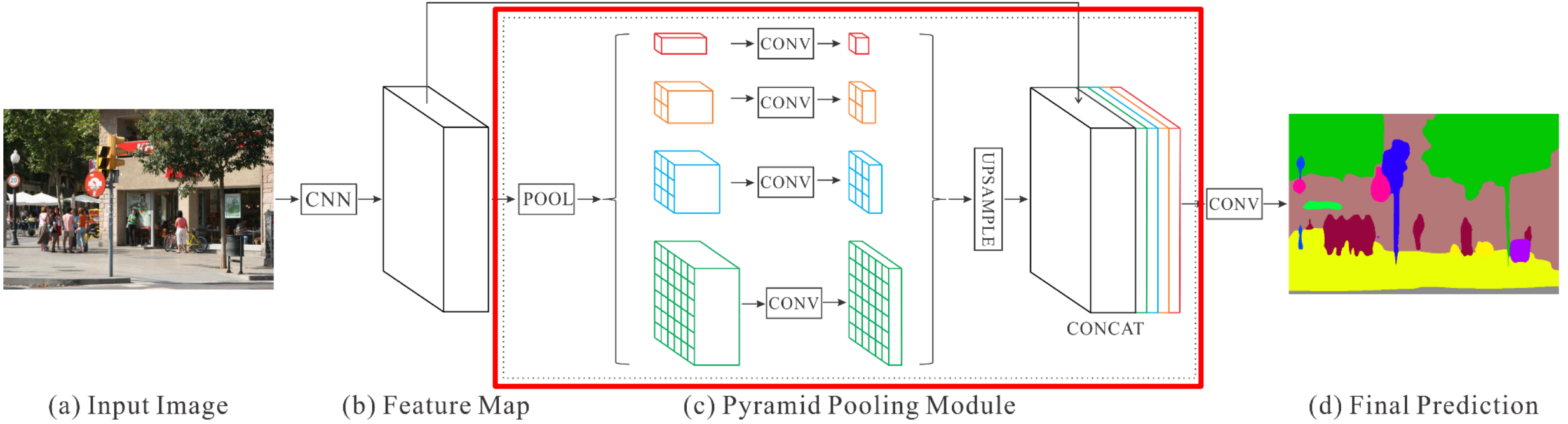


Regular feature extractor

"Pyramid Scene Parsing Network", Zhao et al. CVPR 2017 [15,000+ citation]

Slide Credit: Hengshuang Zhao and Jiaya Jia

Pyramid Scene Parsing Network

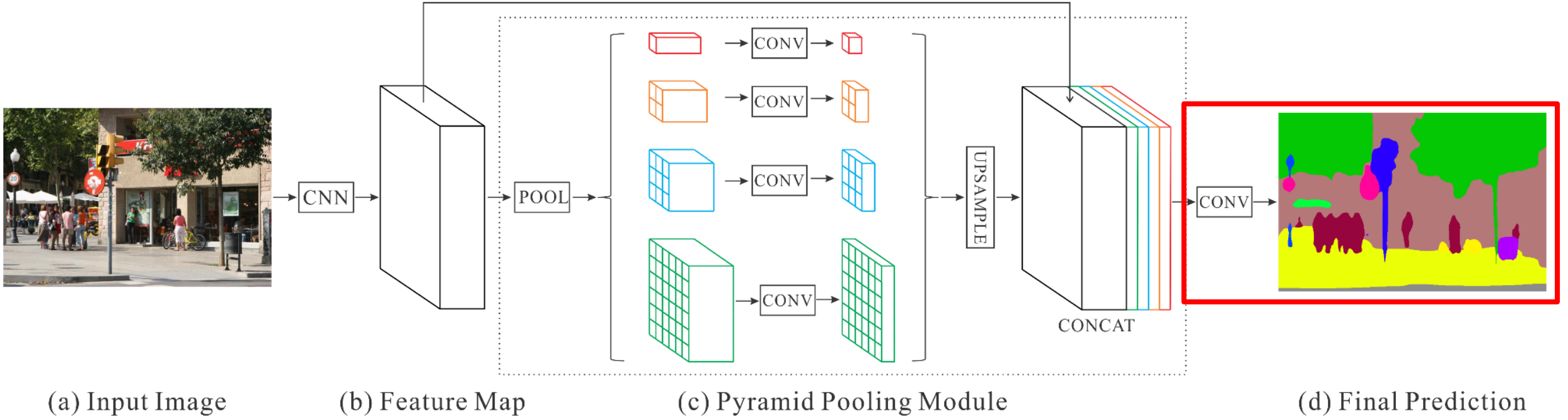


Context modeling: pyramid pooling module

"Pyramid Scene Parsing Network", Zhao et al. CVPR 2017 [15,000+ citation]

Slide Credit: Hengshuang Zhao and Jiaya Jia

Pyramid Scene Parsing Network

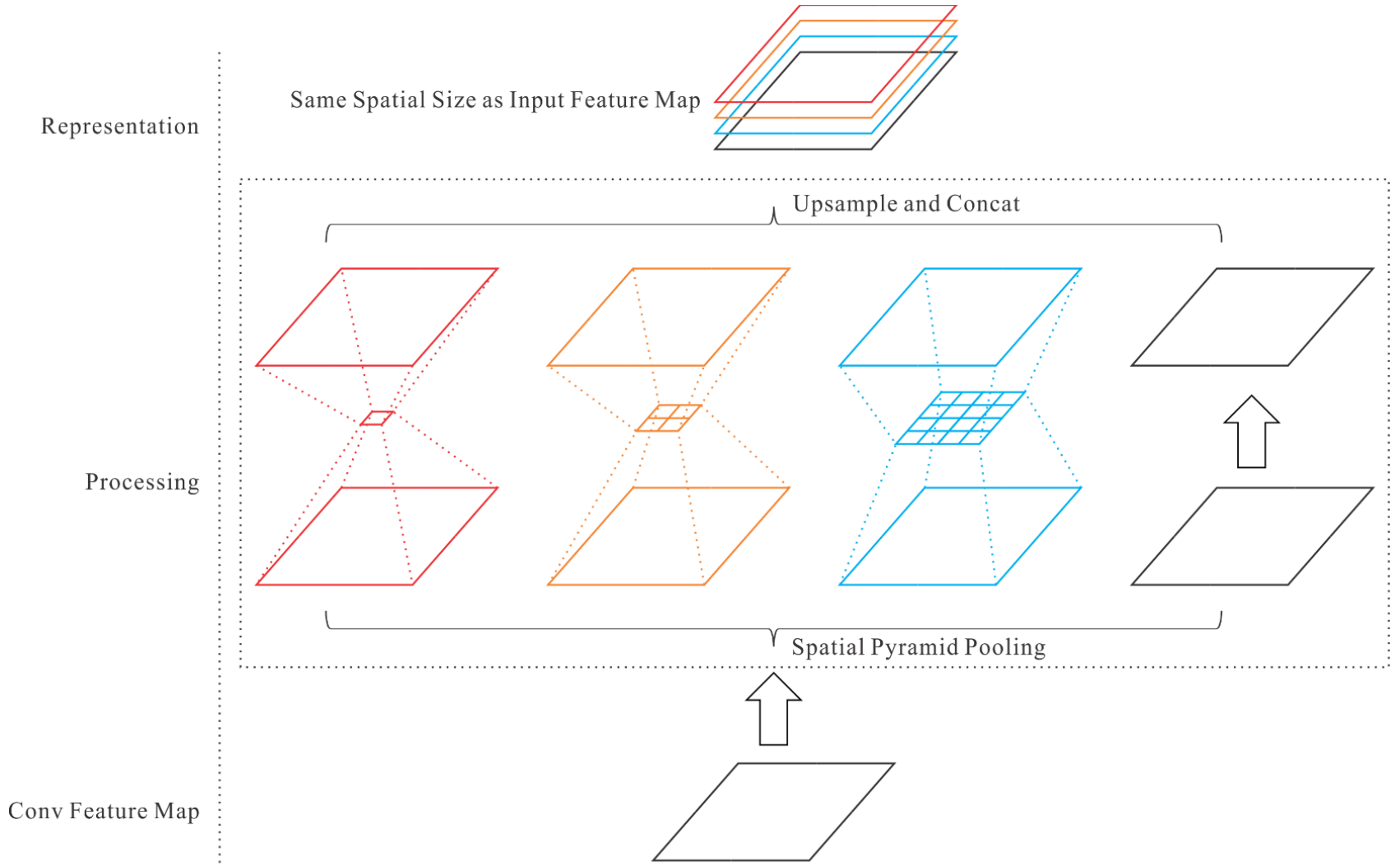


Convolutional classifier for pixel-wise prediction

"Pyramid Scene Parsing Network", Zhao et al. CVPR 2017 [15,000+ citation]

Slide Credit: Hengshuang Zhao and Jiaya Jia

Pyramid Pooling Module



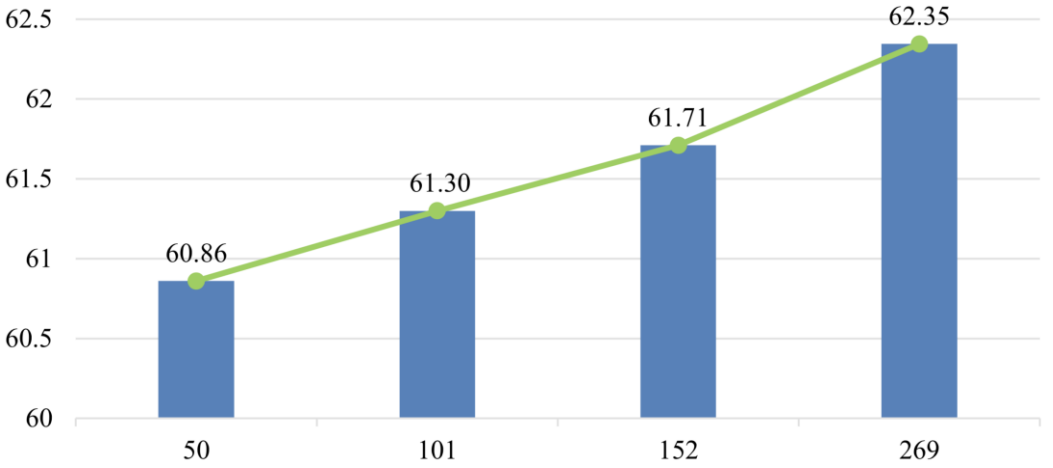
PPM: spatial illustration

Slide Credit: Hengshuang Zhao and Jiaya Jia

ImageNet Scene Parsing Challenge

Method	Mean IoU(%)	Pixel Acc.(%)
FCN [26]	29.39	71.32
SegNet [2]	21.64	71.00
DilatedNet [40]	32.31	73.55
CascadeNet [43]	34.90	74.52
ResNet50-Baseline	34.28	76.35
ResNet50+DA	35.82	77.07
ResNet50+DA+AL	37.23	78.01
ResNet50+DA+AL+PSP	**41.68**	**80.04**
ResNet269+DA+AL+PSP	43.81	80.88
ResNet269+DA+AL+PSP+MS	**44.94**	**81.69**

detailed performance analysis



consistent improvement over network depth

PSPNet: 1st place among totally 75 submissions worldwide.

Slide Credit: Hengshuang Zhao and Jiaya Jia

Result on PASCAL VOC 2012

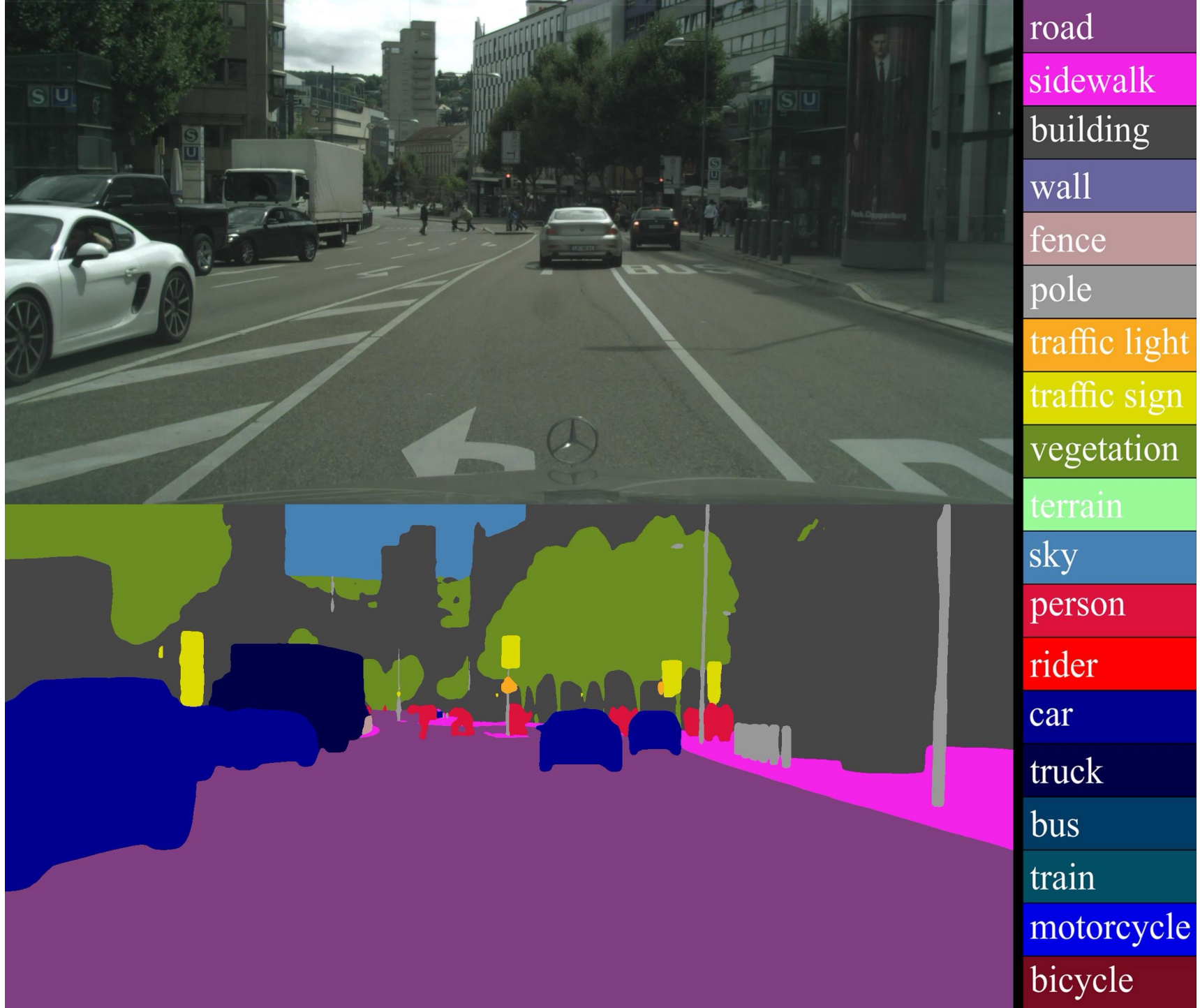
Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mIoU
FCN [26]	76.8	34.2	68.9	49.4	60.3	75.3	74.7	77.6	21.4	62.5	46.8	71.8	63.9	76.5	73.9	45.2	72.4	37.4	70.9	55.1	62.2
Zoom-out [28]	85.6	37.3	83.2	62.5	66.0	85.1	80.7	84.9	27.2	73.2	57.5	78.1	79.2	81.1	77.1	53.6	74.0	49.2	71.7	63.3	69.6
DeepLab [3]	84.4	54.5	81.5	63.6	65.9	85.1	79.1	83.4	30.7	74.1	59.8	79.0	76.1	83.2	80.8	59.7	82.2	50.4	73.1	63.7	71.6
CRF-RNN [41]	87.5	39.0	79.7	64.2	68.3	87.6	80.8	84.4	30.4	78.2	60.4	80.5	77.8	83.1	80.6	59.5	82.8	47.8	78.3	67.1	72.0
DeconvNet [30]	89.9	39.3	79.7	63.9	68.2	87.4	81.2	86.1	28.5	77.0	62.0	79.0	80.3	83.6	80.2	58.8	83.4	54.3	80.7	65.0	72.5
GCRF [36]	85.2	43.9	83.3	65.2	68.3	89.0	82.7	85.3	31.1	79.5	63.3	80.5	79.3	85.5	81.0	60.5	85.5	52.0	77.3	65.1	73.2
DPN [25]	87.7	59.4	78.4	64.9	70.3	89.3	83.5	86.1	31.7	79.9	62.6	81.9	80.0	83.5	82.3	60.5	83.2	53.4	77.9	65.0	74.1
Piecewise [20]	90.6	37.6	80.0	67.8	74.4	92.0	85.2	86.2	39.1	81.2	58.9	83.8	83.9	84.3	84.8	62.1	83.2	58.2	80.8	72.3	75.3
PSPNet	**91.8**	**71.9**	**94.7**	**71.2**	**75.8**	**95.2**	**89.9**	**95.9**	**39.3**	**90.7**	**71.7**	**90.5**	**94.5**	**88.8**	**89.6**	**72.8**	**89.6**	**64.0**	**85.1**	**76.3**	**82.6**
CRF-RNN† [41]	90.4	55.3	88.7	68.4	69.8	88.3	82.4	85.1	32.6	78.5	64.4	79.6	81.9	86.4	81.8	58.6	82.4	53.5	77.4	70.1	74.7
BoxSup† [7]	89.8	38.0	89.2	68.9	68.0	89.6	83.0	87.7	34.4	83.6	67.1	81.5	83.7	85.2	83.5	58.6	84.9	55.8	81.2	70.7	75.2
Dilation8† [40]	91.7	39.6	87.8	63.1	71.8	89.7	82.9	89.8	37.2	84.0	63.0	83.3	89.0	83.8	85.1	56.8	87.6	56.0	80.2	64.7	75.3
DPN† [25]	89.0	61.6	87.7	66.8	74.7	91.2	84.3	87.6	36.5	86.3	66.1	84.4	87.8	85.6	85.4	63.6	87.3	61.3	79.4	66.4	77.5
Piecewise† [20]	94.1	40.7	84.1	67.8	75.9	93.4	84.3	88.4	42.5	86.4	64.7	85.4	89.0	85.8	86.0	67.5	90.2	63.8	80.9	73.0	78.0
FCRNs† [38]	91.9	48.1	93.4	69.3	75.5	94.2	87.5	92.8	36.7	86.9	65.2	89.1	90.2	86.5	87.2	64.6	90.1	59.7	85.5	72.7	79.1
LRR† [9]	92.4	45.1	94.6	65.2	75.8	**95.1**	89.1	92.3	39.0	85.7	70.4	88.6	89.4	88.6	86.6	65.8	86.2	57.4	85.7	77.3	79.3
DeepLab† [4]	92.6	60.4	91.6	63.4	76.3	95.0	88.4	92.6	32.7	88.5	67.6	89.6	92.1	87.0	87.4	63.3	88.3	60.0	86.8	74.5	79.7
PSPNet†	**95.8**	**72.7**	**95.0**	**78.9**	**84.4**	94.7	**92.0**	**95.7**	**43.1**	**91.0**	**80.3**	**91.3**	**96.3**	**92.3**	**90.1**	**71.5**	**94.4**	**66.9**	**88.8**	**82.0**	**85.4**

Slide Credit: Hengshuang Zhao and Jiaya Jia

Result on Cityscapes

Method	road	swalk	build.	wall	fence	pole	tlight	sign	veg.	terrain	sky	person	rider	car	truck	bus	train	mbike	bike	mIoU
CRF-RNN [41]	96.3	73.9	88.2	47.6	41.3	35.2	49.5	59.7	90.6	66.1	93.5	70.4	34.7	90.1	39.2	57.5	55.4	43.9	54.6	62.5
FCN [26]	97.4	78.4	89.2	34.9	44.2	47.4	60.1	65.0	91.4	69.3	93.9	77.1	51.4	92.6	35.3	48.6	46.5	51.6	66.8	65.3
SiCNN+CRF [16]	96.3	76.8	88.8	40.0	45.4	50.1	63.3	69.6	90.6	67.1	92.2	77.6	55.9	90.1	39.2	51.3	44.4	54.4	66.1	66.3
DPN [25]	97.5	78.5	89.5	40.4	45.9	51.1	56.8	65.3	91.5	69.4	94.5	77.5	54.2	92.5	44.5	53.4	49.9	52.1	64.8	66.8
Dilation10 [40]	97.6	79.2	89.9	37.3	47.6	53.2	58.6	65.2	91.8	69.4	93.7	78.9	55.0	93.3	45.5	53.4	47.7	52.2	66.0	67.1
LRR [9]	97.7	79.9	90.7	44.4	48.6	58.6	68.2	72.0	92.5	69.3	94.7	81.6	60.0	94.0	43.6	56.8	47.2	54.8	69.7	69.7
DeepLab [4]	97.9	81.3	90.3	48.8	47.4	49.6	57.9	67.3	91.9	69.4	94.2	79.8	59.8	93.7	56.5	67.5	57.5	57.7	68.8	70.4
Piecewise [20]	98.0	82.6	90.6	44.0	50.7	51.1	65.0	71.7	92.0	72.0	94.1	81.5	61.1	94.3	61.1	65.1	53.8	61.6	70.6	71.6
PSPNet	**98.6**	**86.2**	**92.9**	**50.8**	**58.8**	**64.0**	**75.6**	**79.0**	**93.4**	**72.3**	**95.4**	**86.5**	**71.3**	**95.9**	**68.2**	**79.5**	**73.8**	**69.5**	**77.2**	**78.4**
LRR‡ [9]	97.9	81.5	91.4	50.5	52.7	59.4	66.8	72.7	92.5	70.1	95.0	81.3	60.1	94.3	51.2	67.7	54.6	55.6	69.6	71.8
PSPNet‡	**98.6**	**86.6**	**93.2**	**58.1**	**63.0**	**64.5**	**75.2**	**79.2**	**93.4**	**72.1**	**95.1**	**86.3**	**71.4**	**96.0**	**73.5**	**90.4**	**80.3**	**69.9**	**76.9**	**80.2**

Slide Credit: Hengshuang Zhao and Jiaya Jia


road
sidewalk
building
wall
fence
pole
traffic light
traffic sign
vegetation
terrain
sky
person
rider
car
truck
bus
train
motorcycle
bicycle

PSPNet paper

Pyramid Scene Parsing Network

Hengshuang Zhao[1] Jianping Shi[2] Xiaojuan Qi[1] Xiaogang Wang[1] Jiaya Jia[1]
[1]The Chinese University of Hong Kong [2]SenseTime Group Limited
{hszhao, xjqi, leojia}@cse.cuhk.edu.hk, xgwang@ee.cuhk.edu.hk, shijianping@sensetime.com

Abstract

Scene parsing is challenging for unrestricted open vocabulary and diverse scenes. In this paper, we exploit the capability of global context information by different-region-based context aggregation through our pyramid pooling module together with the proposed pyramid scene parsing network (PSPNet). Our global prior representation is effective to produce good quality results on the scene parsing task, while PSPNet provides a superior framework for pixel-level prediction. The proposed approach achieves state-of-the-art performance on various datasets. It came first in ImageNet scene parsing challenge 2016, PASCAL VOC 2012 benchmark and Cityscapes benchmark. A single PSPNet yields the new record of mIoU accuracy 85.4% on PASCAL VOC 2012 and accuracy 80.2% on Cityscapes.

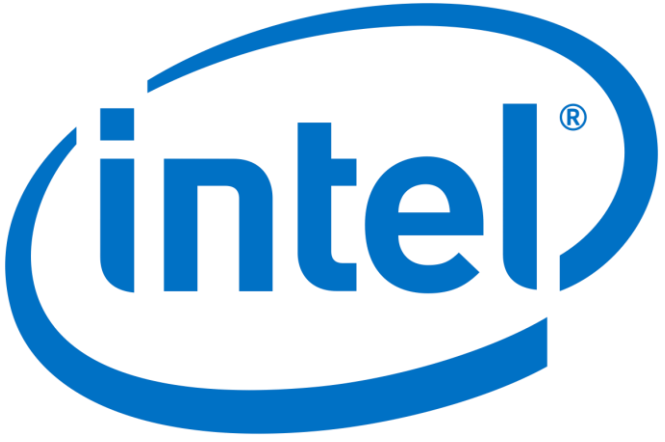
(a) Image (b) Ground Truth

Figure 1. Illustration of complex scenes in ADE20K dataset.

https://arxiv.org/pdf/1612.01105

MSeg: A Composite Dataset for Multi-Domain Semantic Segmentation

John Lambert*, Zhuang Liu*, Ozan Sener,
James Hays, Vladlen Koltun




https://www.youtube.com/watch?v=8wqNX7_4vAE

Which dataset to train on?

Driving: Cityscapes, Mapillary Vistas, CamVid, KITTI, VIPER, Indian Driving Dataset, Berkeley Driving Dataset, WildDash, …

Indoors: NYU, SUN RGBD, ScanNet, InteriorNet, ...

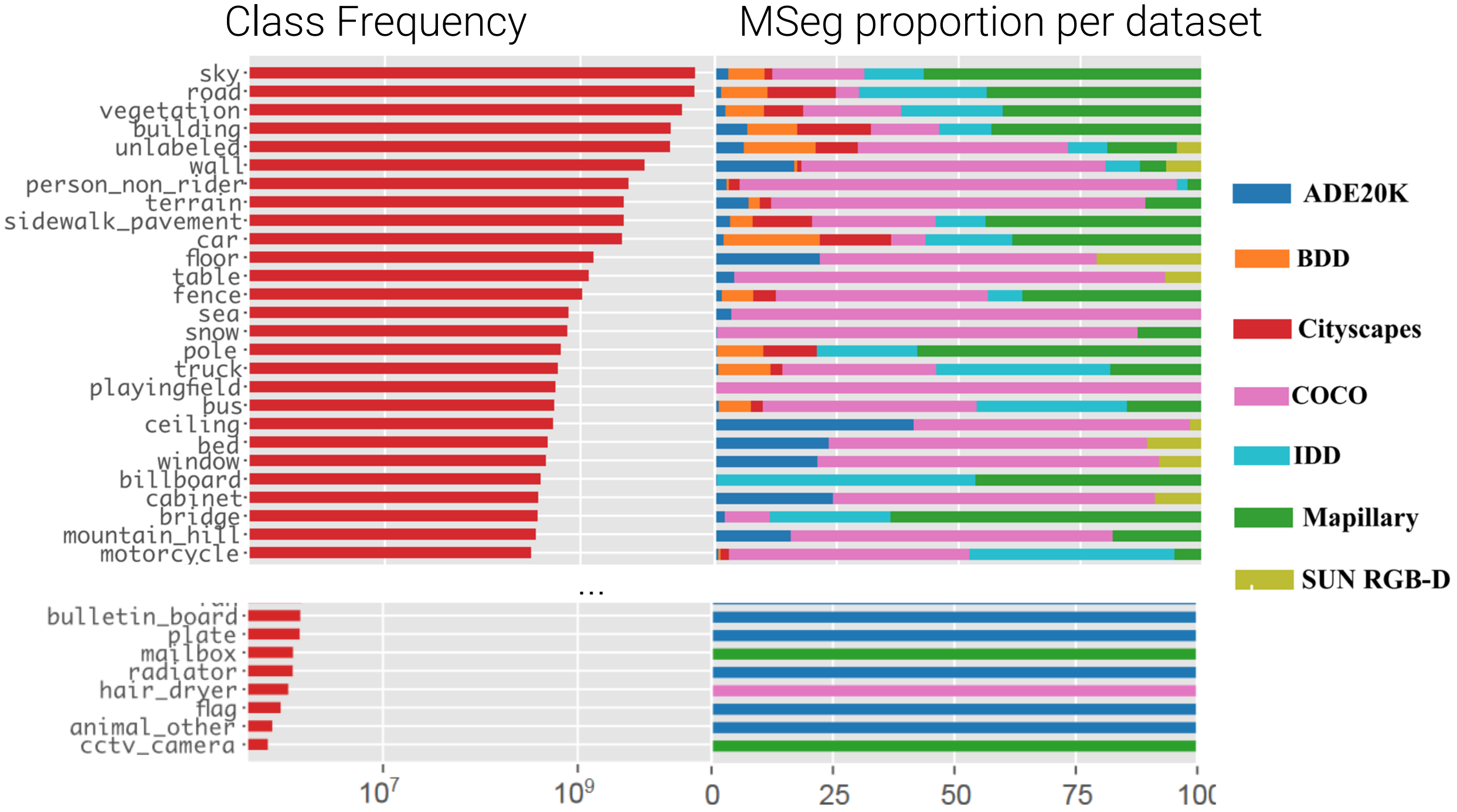
Multi-domain: COCO, ADE20K, PASCAL VOC, ...

John Lambert*, Zhuang Liu*, Ozan Sener, James Hays, Vladlen Koltun: *MSeg: A Composite Dataset for Multi-domain Semantic Segmentation.* CVPR 2020

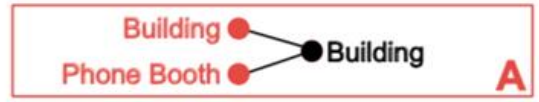
Methodology:
Dataset mixing and zero-shot transfer

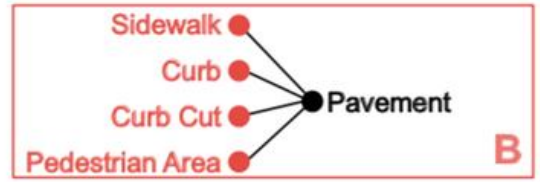
- Perform a training/test split at the level of datasets
- Train on many diverse datasets
- Test on datasets that were never seen during training
- Zero-shot cross-dataset transfer is a proxy for generality and robustness in the real world

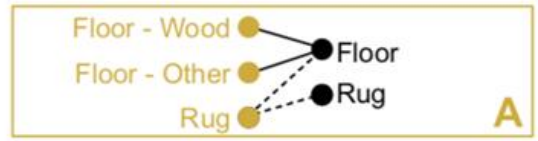
Dataset name	Origin domain	# Images
Training & Validation		
COCO [19] + COCO STUFF [4]	Everyday objects	123,287
ADE20K [46]	Everyday objects	22,210
MAPILLARY [25]	Driving (Worldwide)	20,000
IDD [40]	Driving (India)	7,974
BDD [43]	Driving (United States)	8,000
CITYSCAPES [7]	Driving (Germany)	3,475
SUN RGBD [36]	Indoor	5,285
Test		
PASCAL VOC [10]	Everyday objects	1,449
PASCAL CONTEXT [24]	Everyday objects	5,105
CAMVID [3]	Driving (U.K.)	101
WILDDASH [44]	Driving (Worldwide)	70
KITTI [11]	Driving (Germany)	200
SCANNET-20 [8]	Indoor	5,436

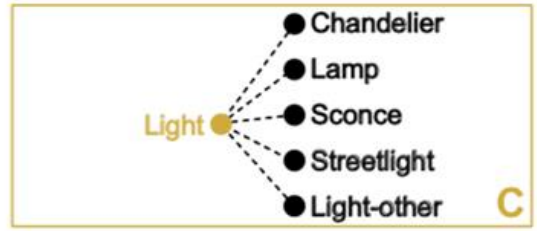

Class Frequency
MSeg proportion per dataset
sky
road
vegetation
building
unlabeled
wall
person_non_rider
terrain
sidewalk_pavement
car
floor
table
fence
sea
snow
pole
truck
playingfield
bus
ceiling
bed
window
billboard
cabinet
bridge
mountain_hill
motorcycle
...
bulletin_board
plate
mailbox
radiator
hair_dryer
flag
animal_other
cctv_camera
10^7
10^9
0
25
50
75
100
ADE20K
BDD
Cityscapes
COCO
IDD
Mapillary
SUN RGB-D


Mapillary
COCO
ADE20K
IDD
Cityscapes BDD
SUN RGB-D
MSeg
Bag
Tunnel
Building
Television
Floor
Rug
Armchair
Bench
Chair-other
Ottoman
Seat
Stool
Swivelchair
Sofa
Chandelier
Lamp
Light-other
Sconce
Streetlight
A
B
C

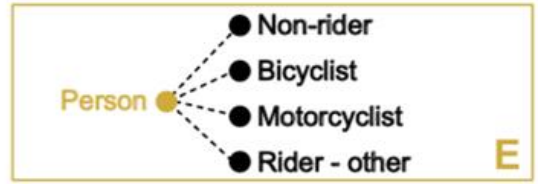

Building
Phone Booth
Building
A

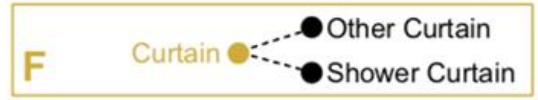

Sidewalk
Curb
Curb Cut
Pedestrian Area
Pavement
B

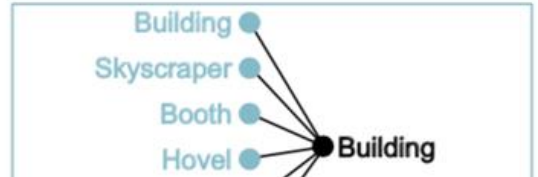

Floor - Wood
Floor - Other
Rug
Floor
Rug
A

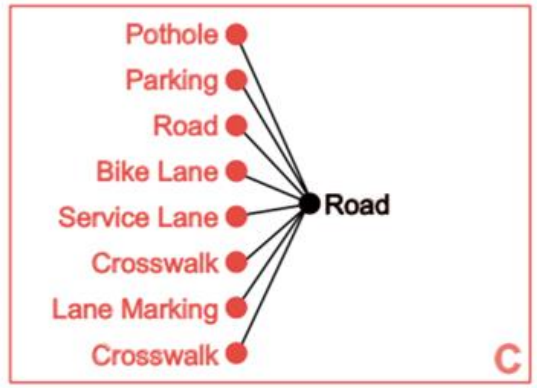

Light
Chandelier
Lamp
Sconce
Streetlight
Light-other
C

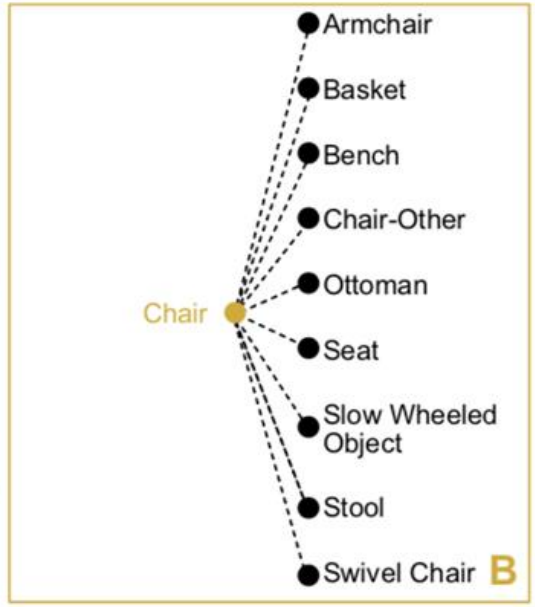

Pavement
Road
Runway
Pavement - Sidewalk
Road
D

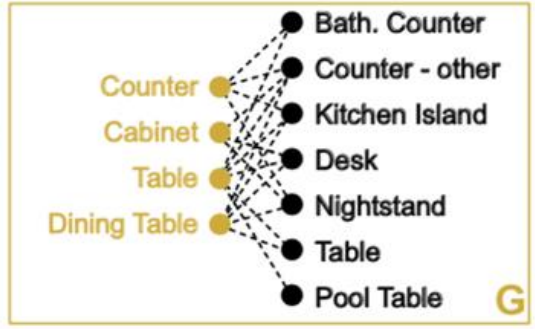

Person
Non-rider
Bicyclist
Motorcyclist
Rider - other
E

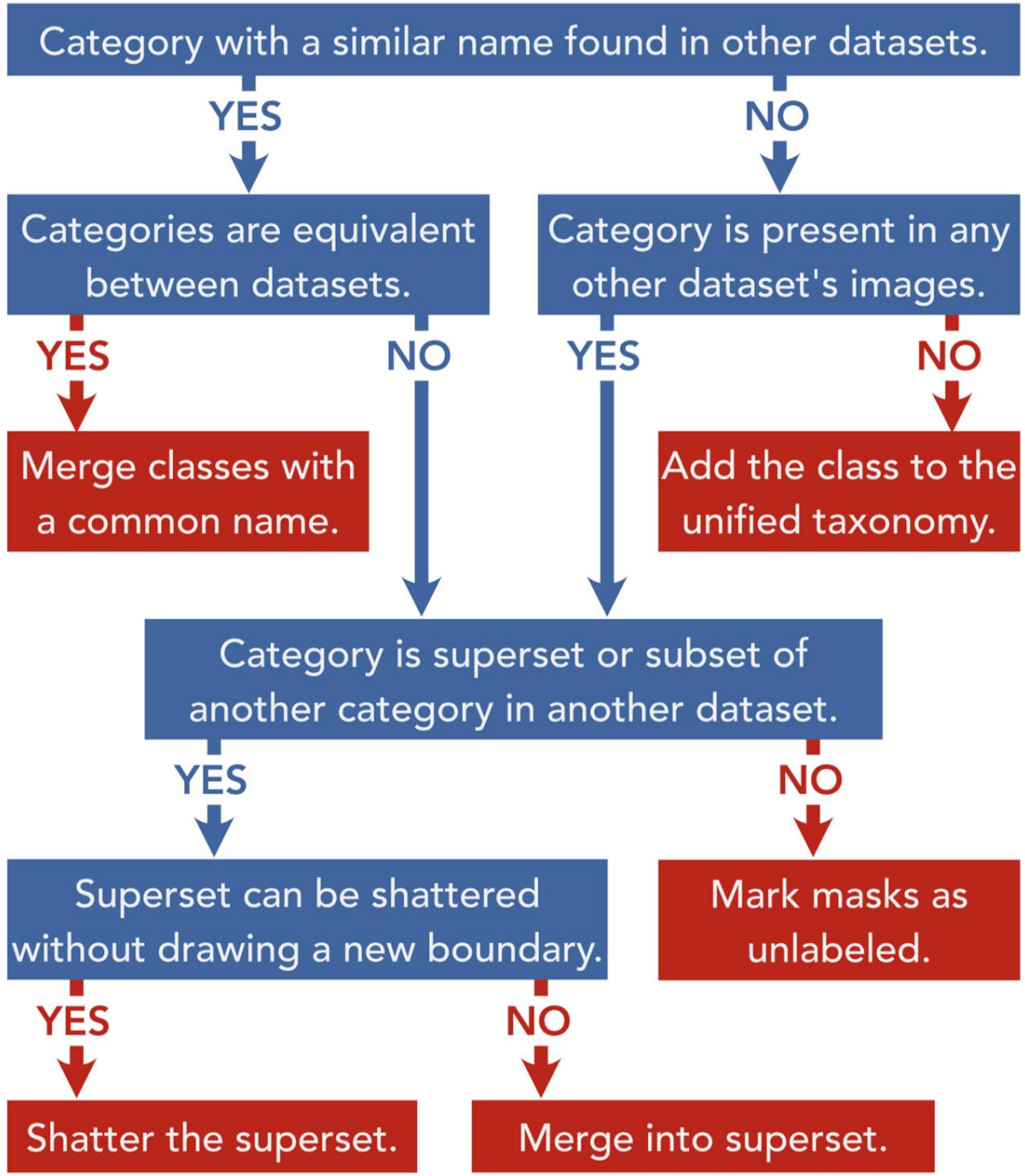

F
Curtain
Other Curtain
Shower Curtain


Building
Skyscraper
Booth
Hovel
Building


Pothole
Parking
Road
Bike Lane
Service Lane
Crosswalk
Lane Marking
Crosswalk
Road
C


Chair
Armchair
Basket
Bench
Chair-Other
Ottoman
Seat
Slow Wheeled Object
Stool
Swivel Chair
B


Counter
Cabinet
Table
Dining Table
Bath. Counter
Counter - other
Kitchen Island
Desk
Nightstand
Table
Pool Table
G


Category with a similar name found in other datasets.
YES
NO
Categories are equivalent between datasets.
Category is present in any other dataset's images.
YES
NO
YES
NO
Merge classes with a common name.
Add the class to the unified taxonomy.
Category is superset or subset of another category in another dataset.
YES
NO
Superset can be shattered without drawing a new boundary.
Mark masks as unlabeled.
YES
NO
Shatter the superset.
Merge into superset.

Generality and Robustness

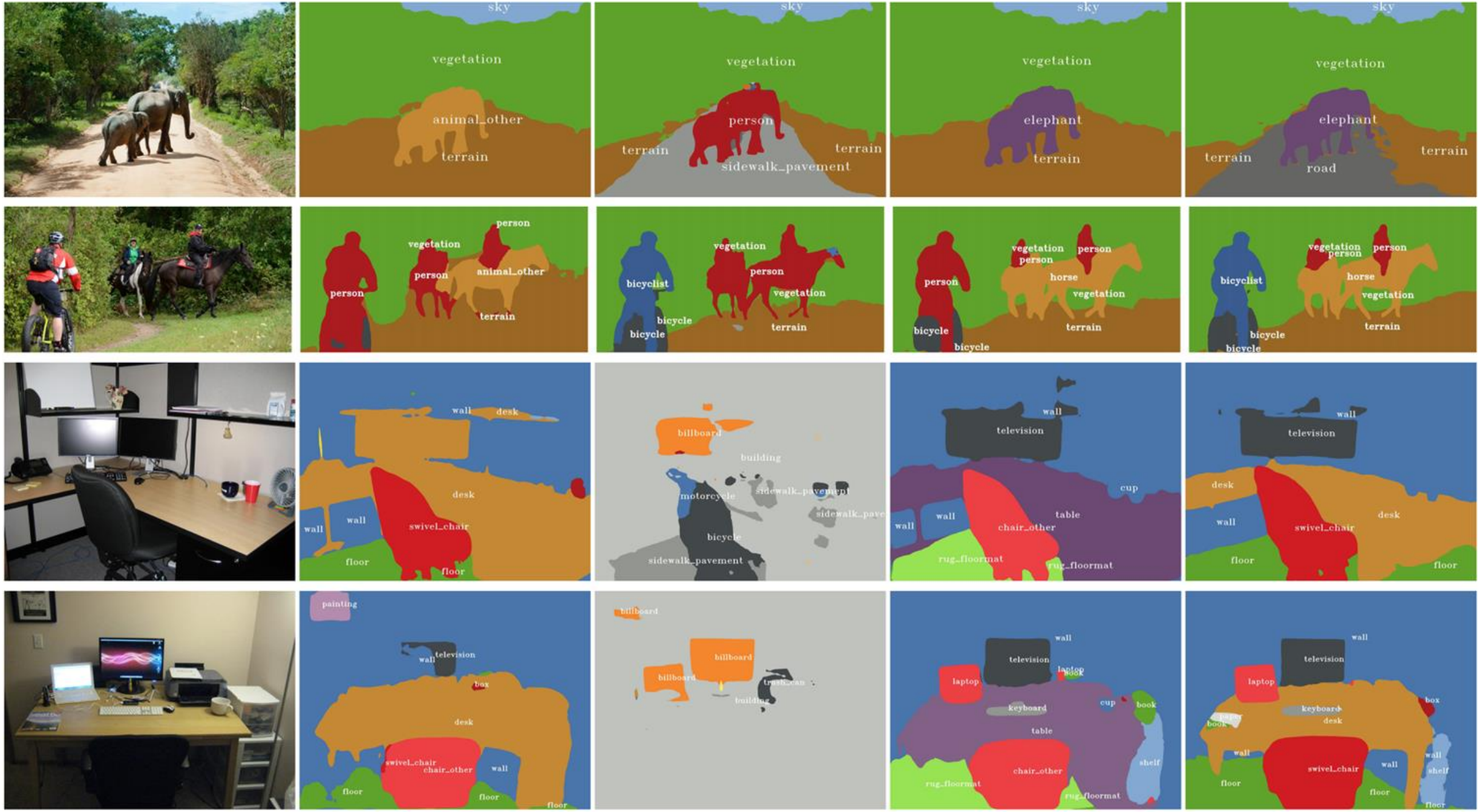
Train/Test	COCO	ADE20K	Mapillary	IDD	BDD	Cityscapes	SUN	*h. mean*
COCO	**52.7**	19.1	28.4	31.1	44.9	46.9	29.6	32.4
ADE20K	14.6	**45.6**	24.2	26.8	40.7	44.3	36.0	28.7
Mapillary	7.0	6.2	**53.0**	50.6	59.3	71.9	0.3	1.7
IDD	3.2	3.0	24.6	**64.9**	42.4	48.0	0.4	2.3
BDD	3.8	4.2	23.2	32.3	63.4	58.1	0.3	1.6
Cityscapes	3.4	3.1	22.1	30.1	44.1	77.5	0.2	1.2
SUN RGBD	3.4	7.0	1.1	1.0	2.2	2.6	43.0	2.1
MSeg-w/o relabeling	50.4	**45.4**	**53.1**	**65.1**	66.5	**79.5**	**49.9**	**56.6**
MSeg	50.7	**45.7**	**53.1**	**65.3**	**68.5**	**80.4**	**50.3**	**57.1**

Method	Mean IoU(%)	Pixel Acc.(%)
FCN [26]	29.39	71.32
SegNet [2]	21.64	71.00
DilatedNet [40]	32.31	73.55
CascadeNet [43]	34.90	74.52
ResNet50-Baseline	34.28	76.35
ResNet50+DA	35.82	77.07
ResNet50+DA+AL	37.23	78.01
ResNet50+DA+AL+PSP	**41.68**	**80.04**
ResNet269+DA+AL+PSP	43.81	80.88
ResNet269+DA+AL+PSP+MS	**44.94**	**81.69**

Accuracy on MSeg *training* datasets

Train/Test	VOC	Context	CamVid	WildDash	KITTI	ScanNet	*h. mean*
COCO	**73.4**	**43.3**	58.7	38.2	47.6	33.4	45.8
ADE20K	35.4	23.9	52.6	38.6	41.6	42.9	36.9
Mapillary	22.5	13.6	82.1	55.4	**67.7**	2.1	9.3
IDD	14.6	6.5	72.1	41.2	51.0	1.6	6.5
BDD	14.4	7.1	70.7	52.2	54.5	1.4	6.1
Cityscapes	13.3	6.8	76.1	30.1	57.6	1.7	6.8
SUN RGBD	10.0	4.3	0.1	1.9	1.1	42.6	0.3
MSeg-1m	70.7	**42.7**	**83.3**	**62.0**	**67.0**	**48.2**	**59.2**
MSeg-1m-w/o relabeling	70.2	42.7	82.0	62.7	65.5	43.2	57.6
Oracle	77.8	45.8	78.8	–	58.4	62.3	–

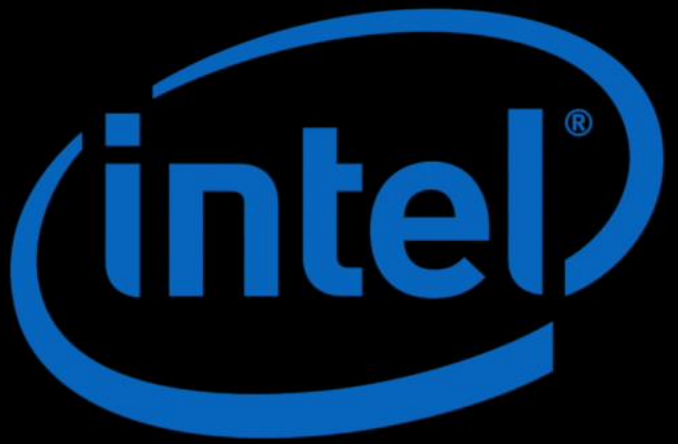
Accuracy on MSeg test datasets



Input image | ADE20K model | Mapillary model | COCO model | MSeg model

MSeg: A Composite Dataset for Multi-domain Semantic Segmentation

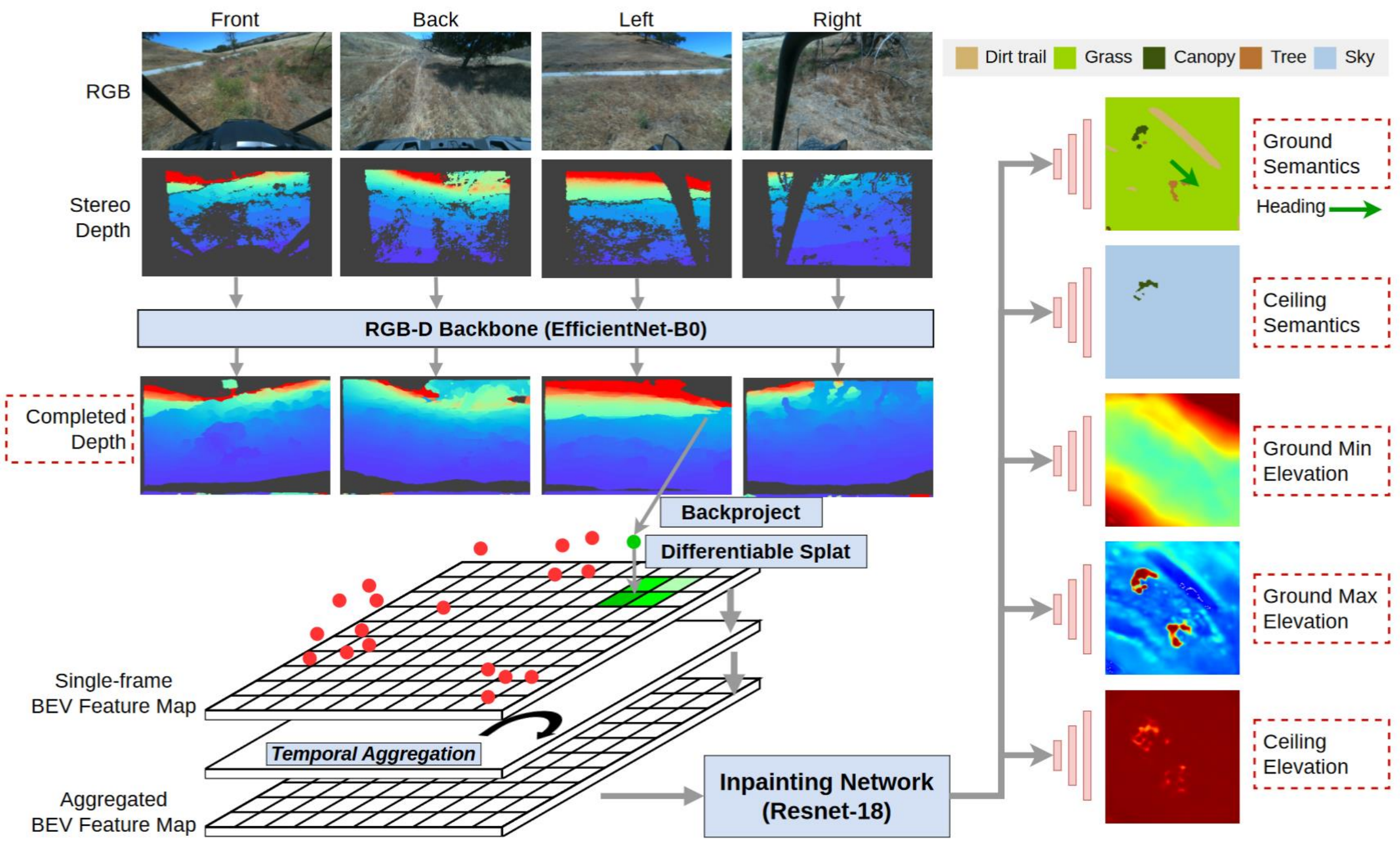
John Lambert*, Zhuang Liu*, Ozan Sener,
James Hays, Vladlen Koltun

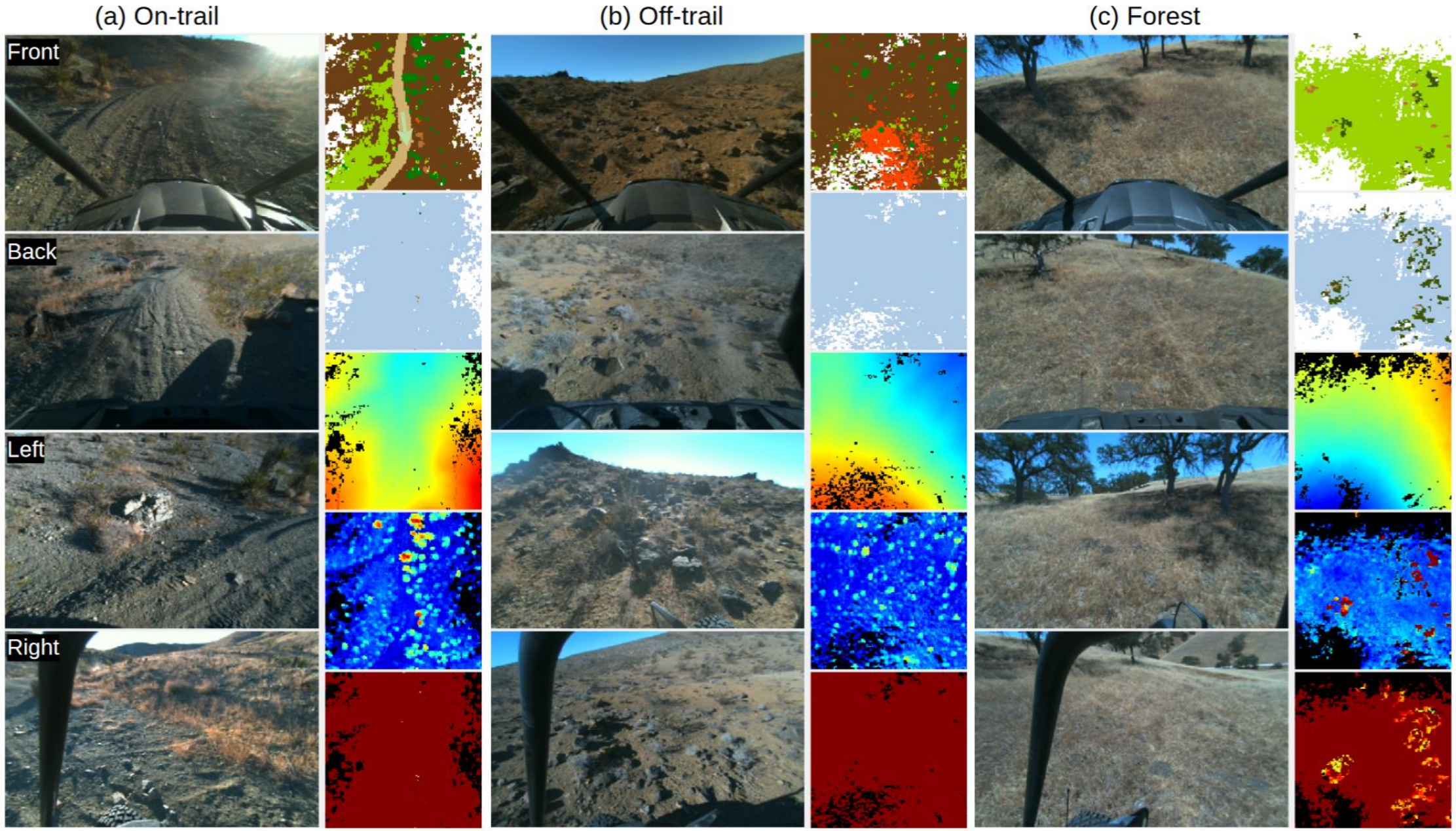


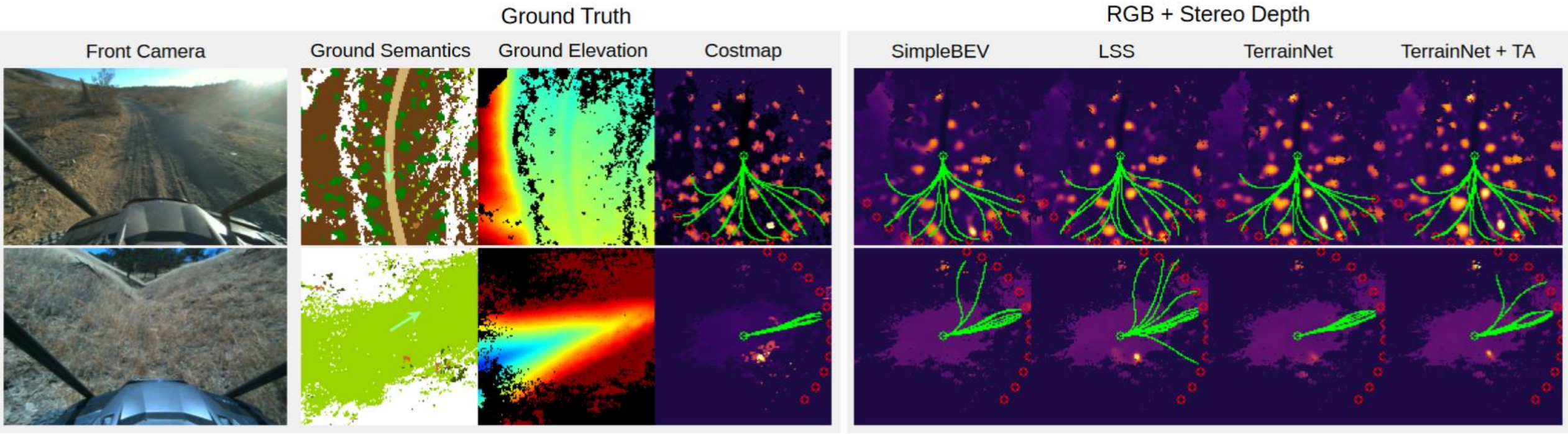
“Bird’s eye” Semantic Segmentation for Robots

TerrainNet: Visual Modeling of Complex Terrain for High-speed, Off-road Navigation

Xiangyun Meng, Nathan Hatch, Alexander Lambert, Anqi Li, Nolan Wagener, Matthew Schmittle, JoonHo Lee, Wentao Yuan, Zoey Chen, Samuel Deng, Greg Okopal, Dieter Fox, Byron Boots, Amirreza Shaban


Front
Back
Left
Right
RGB
Stereo Depth
RGB-D Backbone (EfficientNet-B0)
Completed Depth
Backproject
Differentiable Splat
Single-frame BEV Feature Map
Temporal Aggregation
Aggregated BEV Feature Map
Inpainting Network (Resnet-18)
Dirt trail
Grass
Canopy
Tree
Sky
Ground Semantics
Heading
Ceiling Semantics
Ground Min Elevation
Ground Max Elevation
Ceiling Elevation


(a) On-trail
(b) Off-trail
(c) Forest
Front
Back
Left
Right
Average speed: 7.8 m/s
Max speed: 13.4 m/s
Average speed: 2.8 m/s
Max speed: 7.4 m/s
Average speed: 4.2 m/s
Max speed: 10.9 m/s
Ontology
Dirt
Dirt trail
Grass
Bush
Canopy
Tree
Rock
Sky


Ground Truth
RGB + Stereo Depth
Front Camera
Ground Semantics
Ground Elevation
Costmap
SimpleBEV
LSS
TerrainNet
TerrainNet + TA

Project 4