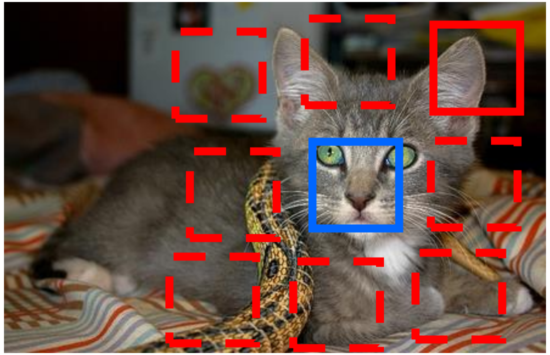
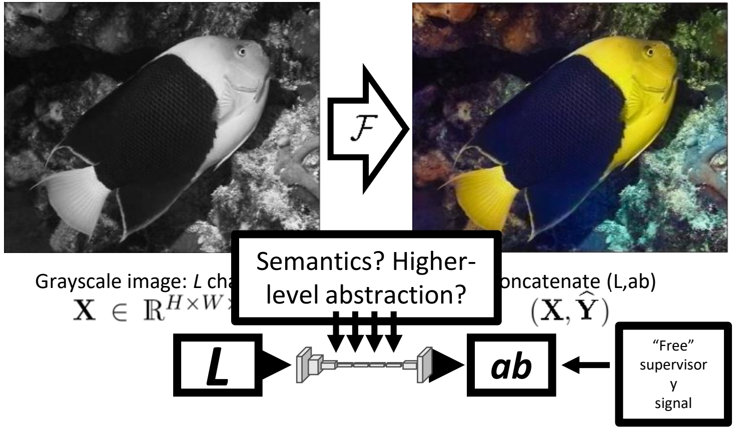
Semantic Segmentation, PSPNet, and MSeg

Many slides by John Lambert

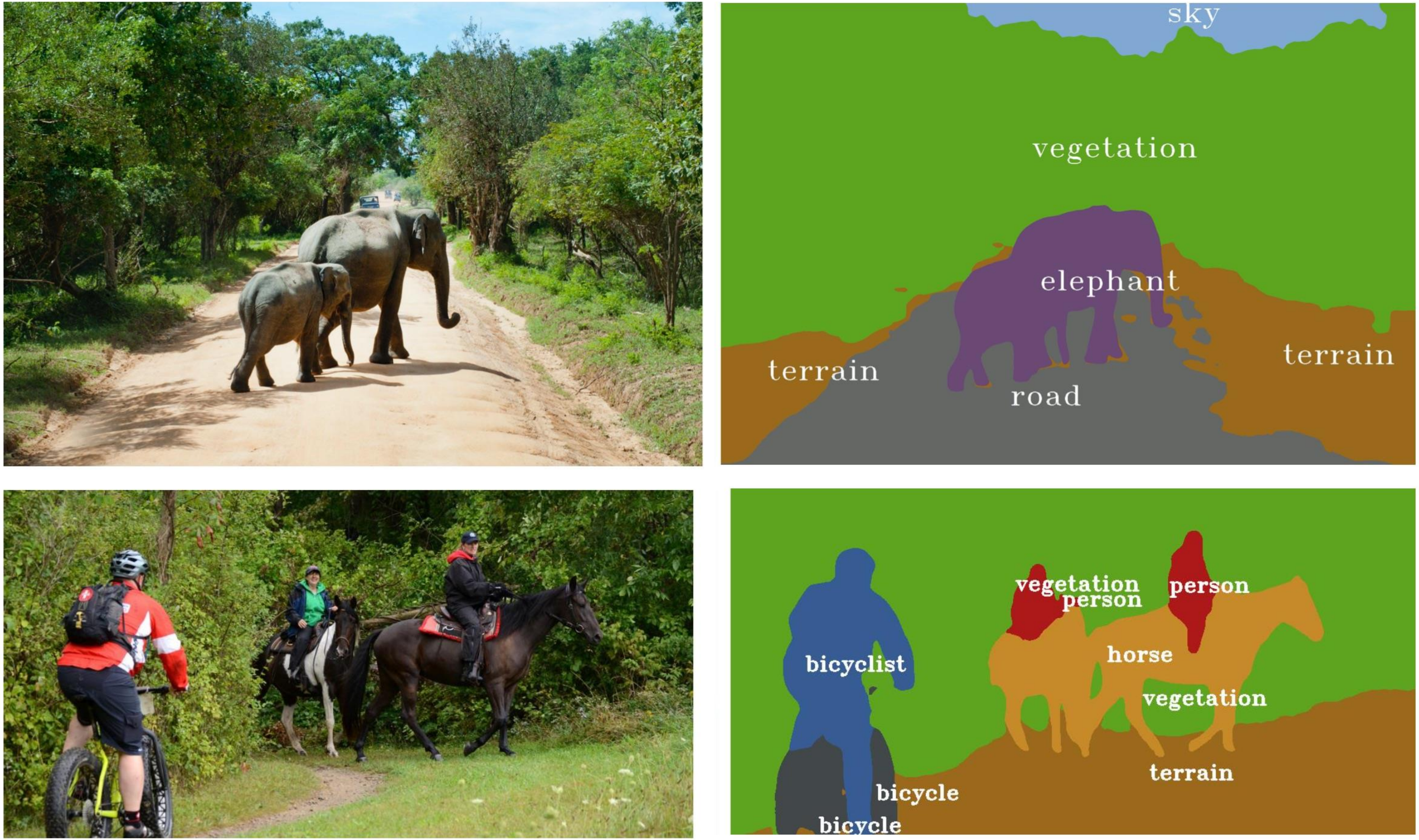
Recap – Self-supervised learning



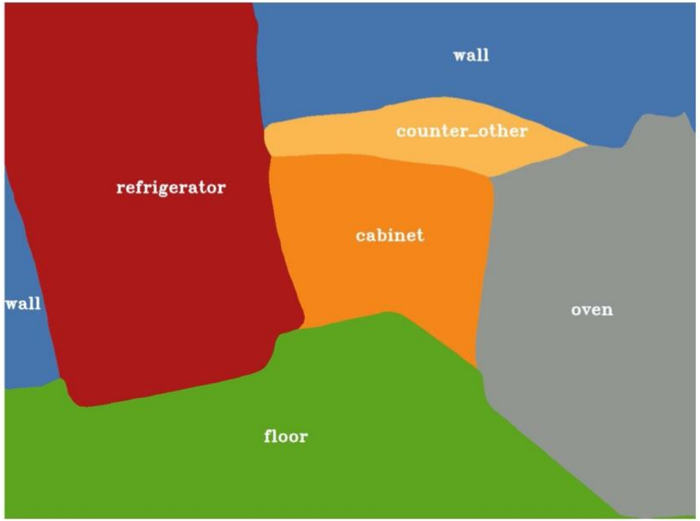


We looked at two of many ways to "self supervised" deep networks. These networks, trained on "pretext" tasks, generalize to other learning problems.

Semantic Segmentation


sky
vegetation
elephant
terrain
road
terrain
vegetation
person
person
bicyclist
horse
vegetation
terrain
bicycle
bicycle


sky
vegetation
truck
building
truck
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

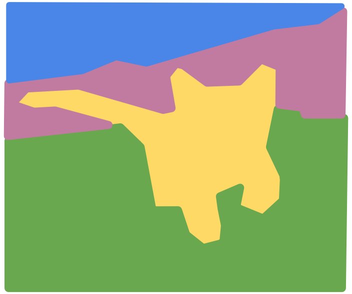
Measuring Performance: Intersection over Union

$$IoU = \frac{\text{area of overlap}}{\text{area of union}}$$

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

Classification	**Semantic Segmentation**	Object Detection	Instance Segmentation
			
CAT	GRASS, CAT, TREE, SKY	DOG, DOG, CAT	DOG, DOG, CAT

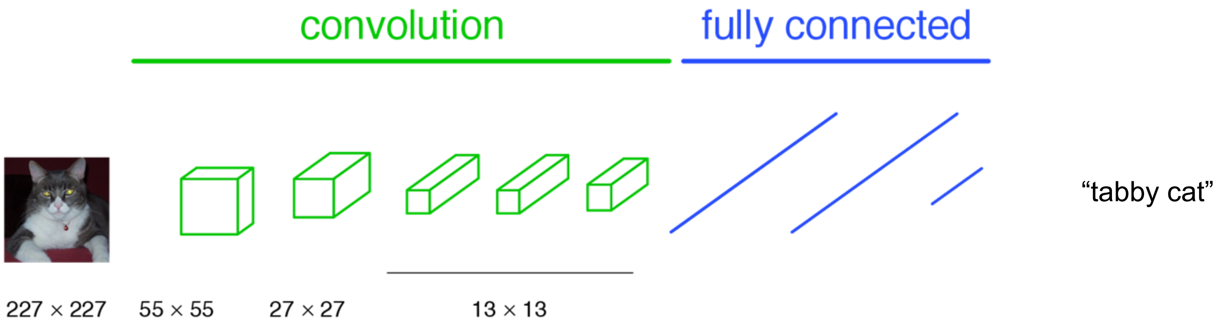
No spatial extent

No objects, just pixels

Multiple Objects

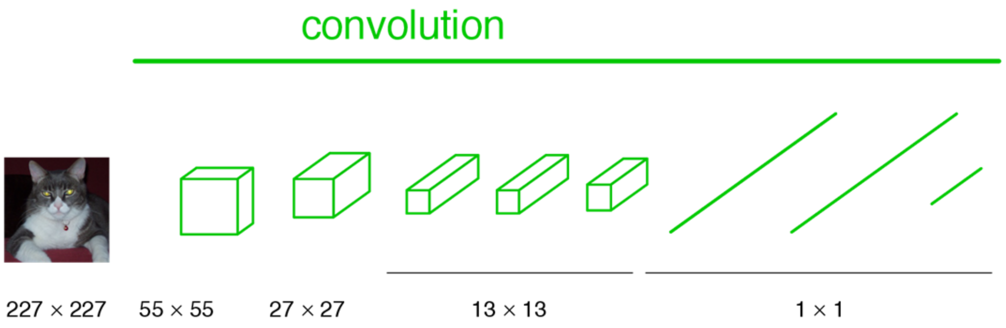
Slide Credit: Justin Johnson and David Fouhey

a classification network



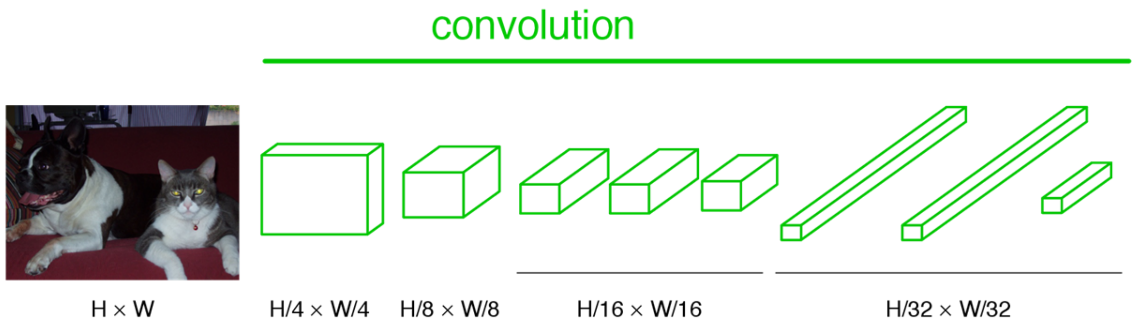
Fully Convolutional Networks for Semantic Segmentation.
Jon Long, Evan Shelhamer, Trevor Darrell. CVPR 2015

becoming fully convolutional

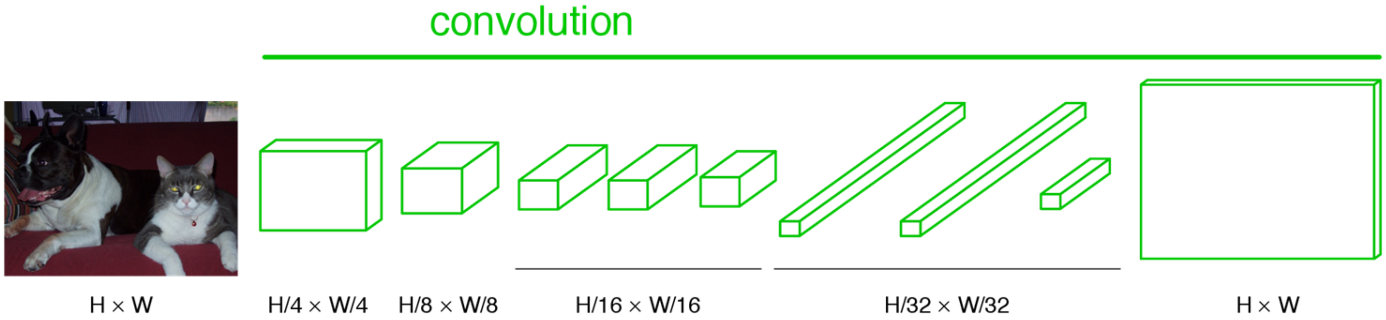


Note: “Fully Convolutional” and “Fully Connected” aren’t the same thing.
They’re almost opposites, in fact.

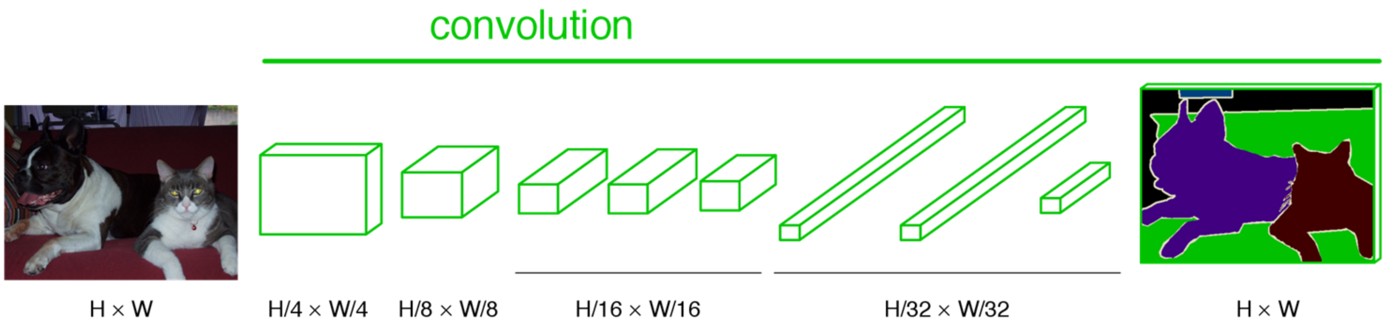
becoming fully convolutional



upsampling output

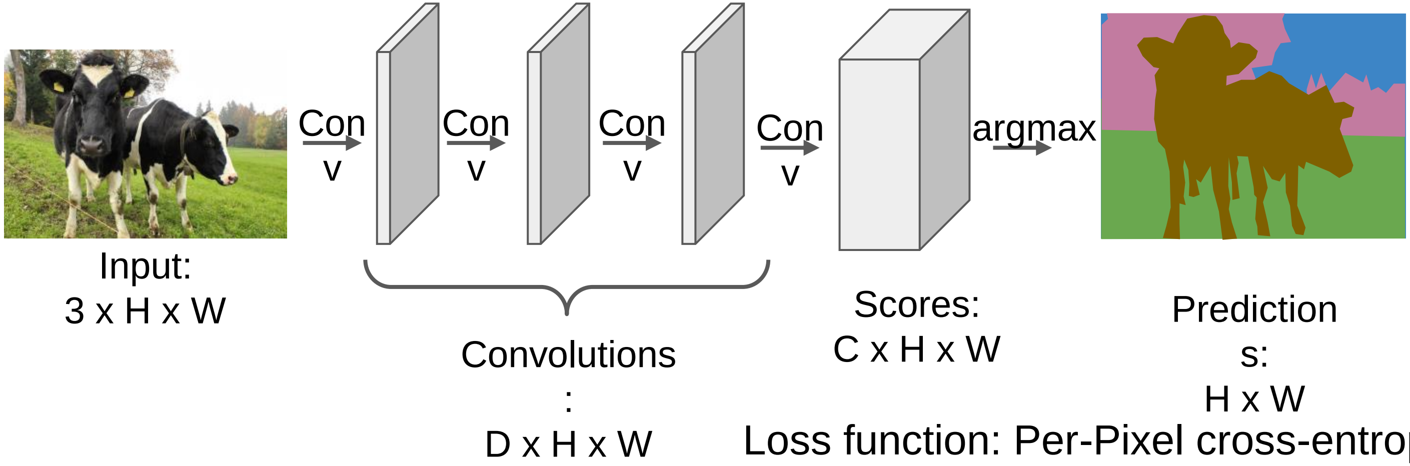


end-to-end, pixels-to-pixels network



Fully Convolutional Network

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



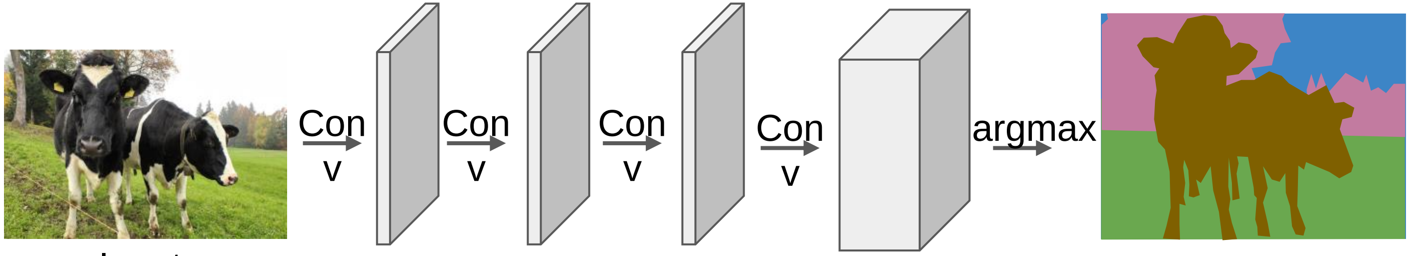
Loss function: Per-Pixel cross-entropy

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!



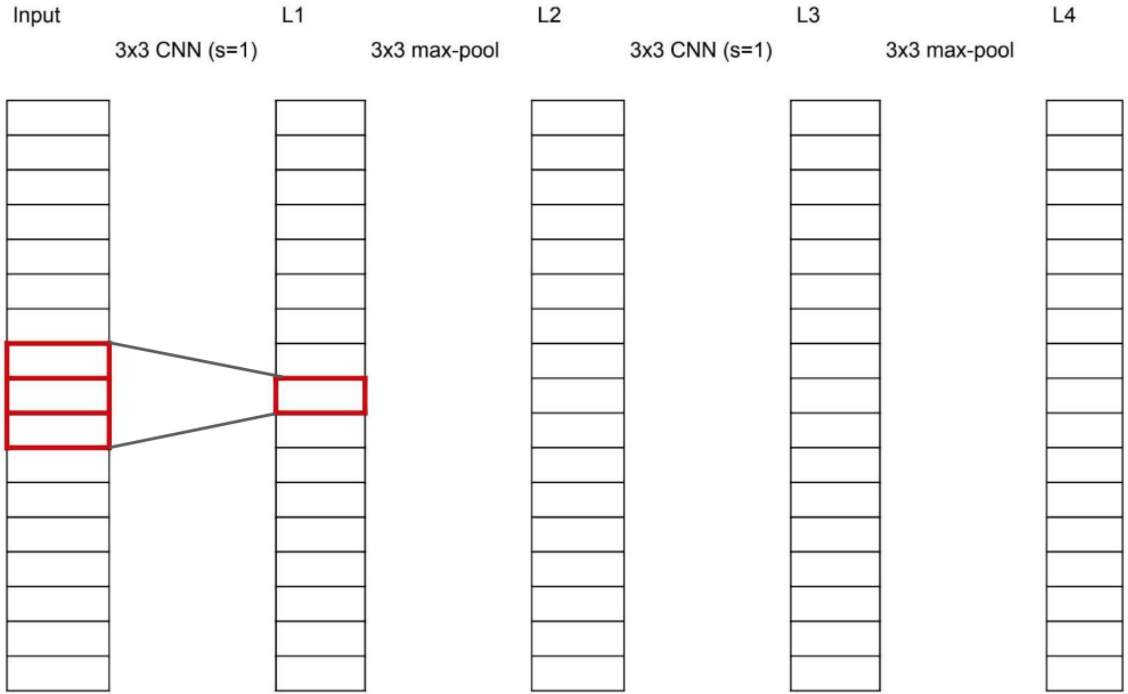
Input:
3 x H x W

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

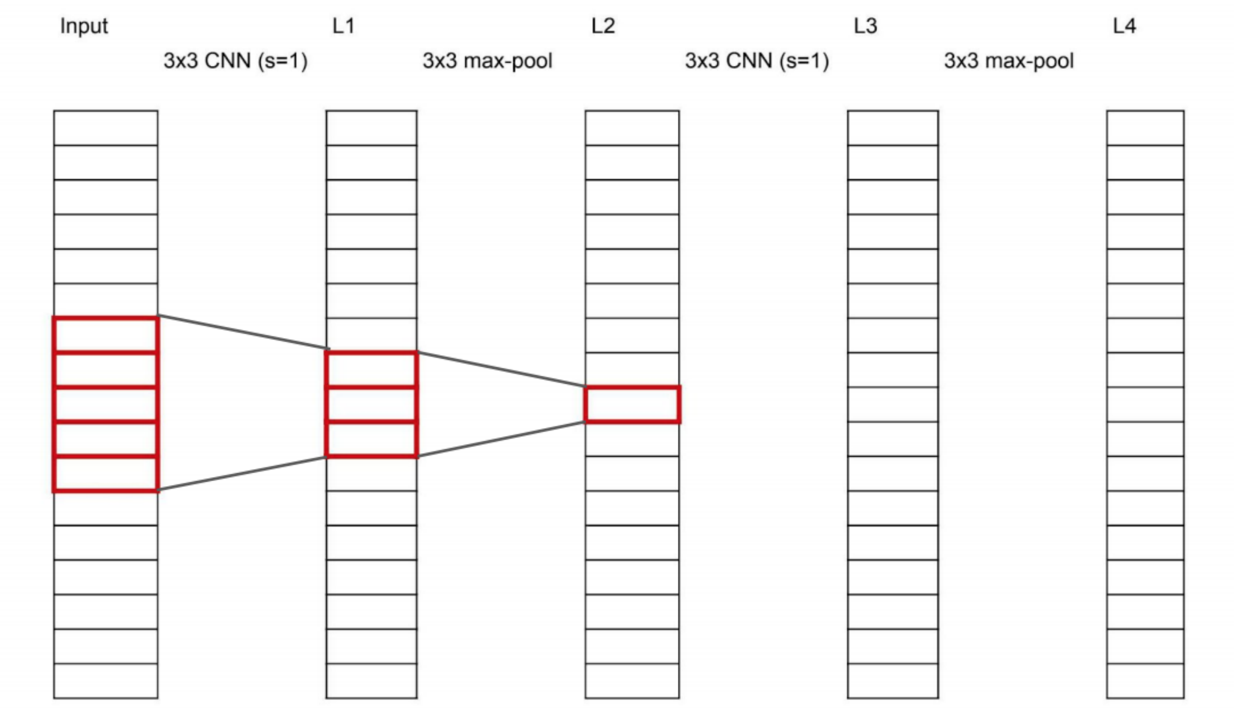
Slide Credit: Justin Johnson and David Fouhey

Receptive field



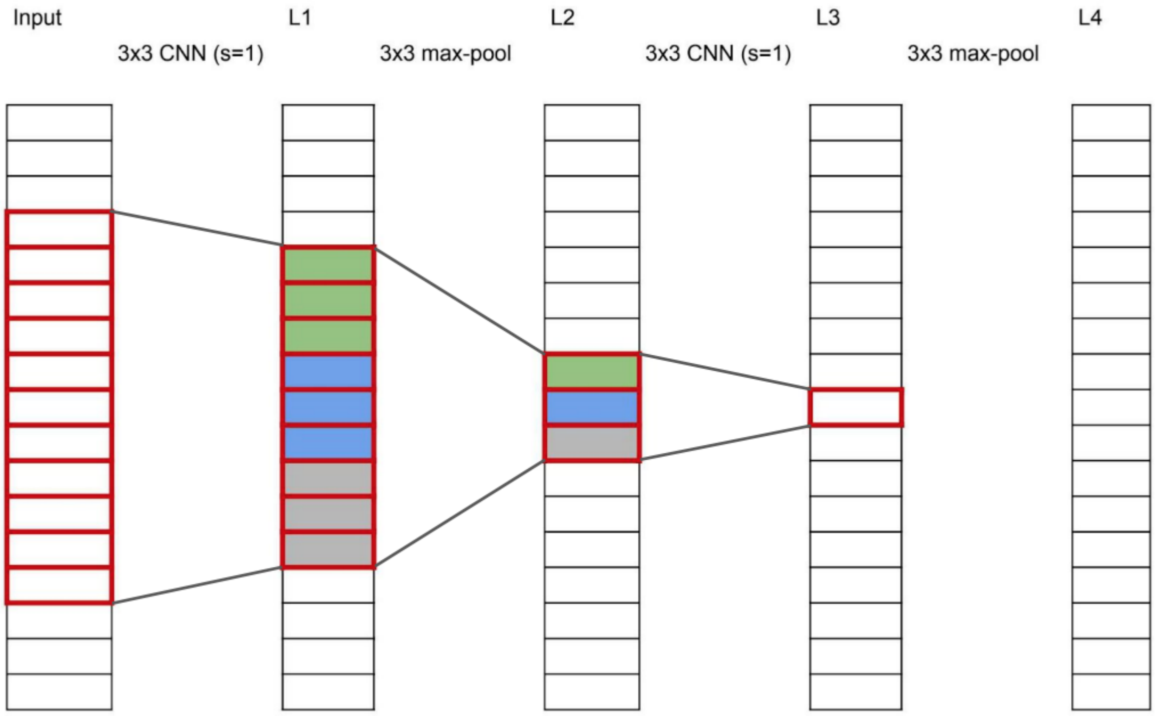
Slide Credit: Frank Dellaert https://dellaert.github.io/19F-4476/resources/receptiveField.pdf

Receptive field



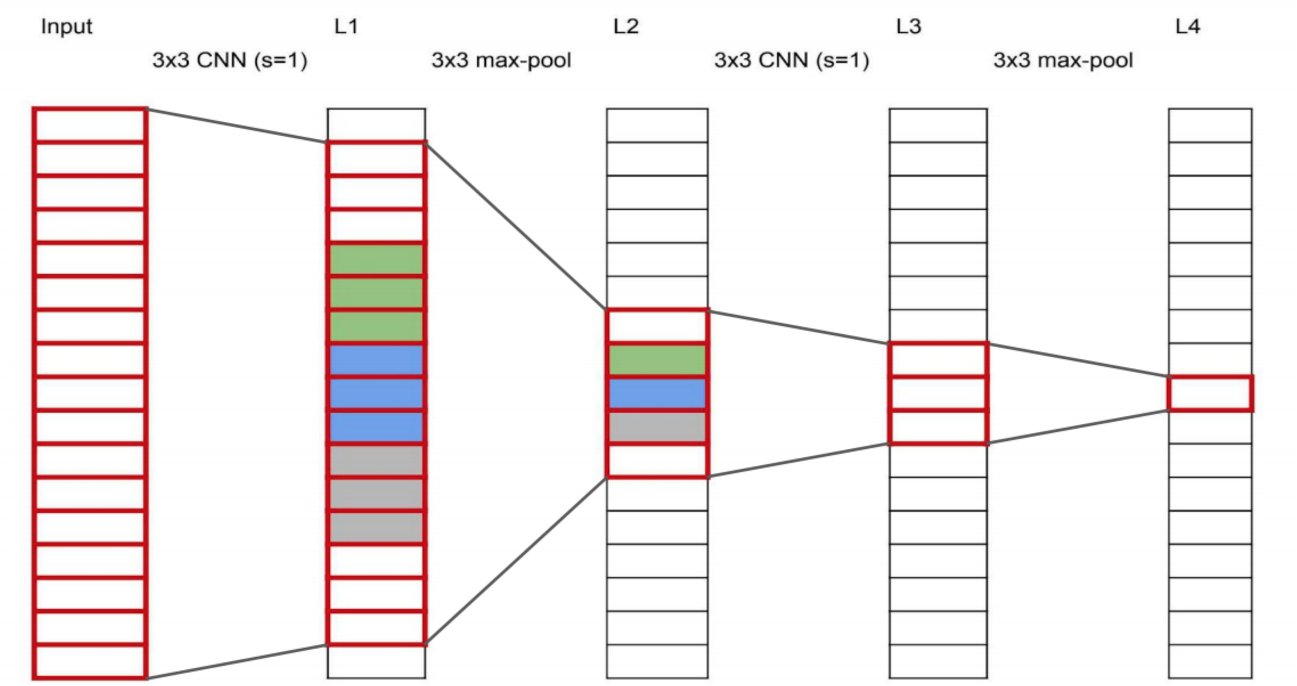
Slide Credit: Frank Dellaert https://dellaert.github.io/19F-4476/resources/receptiveField.pdf

Receptive field



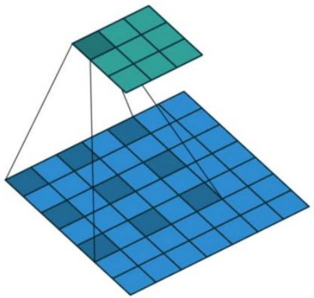
Slide Credit: Frank Dellaert https://dellaert.github.io/19F-4476/resources/receptiveField.pdf

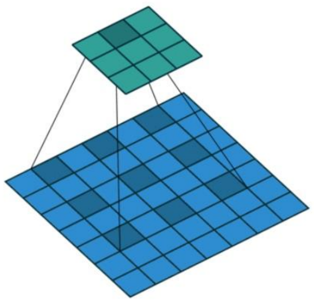
Receptive field



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

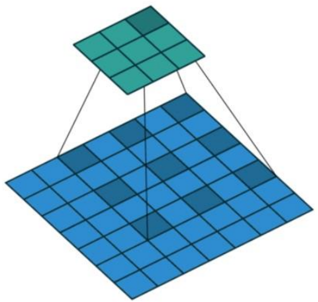

No padding, no stride, dilation

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

2 Dilated Convolutions

Let $F : \mathbb{Z}^2 \rightarrow \mathbb{R}$ be a discrete function. Let $\Omega_r = [-r, r]^2 \cap \mathbb{Z}^2$ and let $k : \Omega_r \rightarrow \mathbb{R}$ be a discrete filter of size $(2r + 1)^2$. The discrete convolution operator $*$ can be defined as

$$(F * k)(\mathbf{p}) = \sum_{\mathbf{s}+\mathbf{t}=\mathbf{p}} F(\mathbf{s})\, k(\mathbf{t}). \tag{1}$$

We now generalize this operator. Let l be a dilation factor and let $*_l$ be defined as

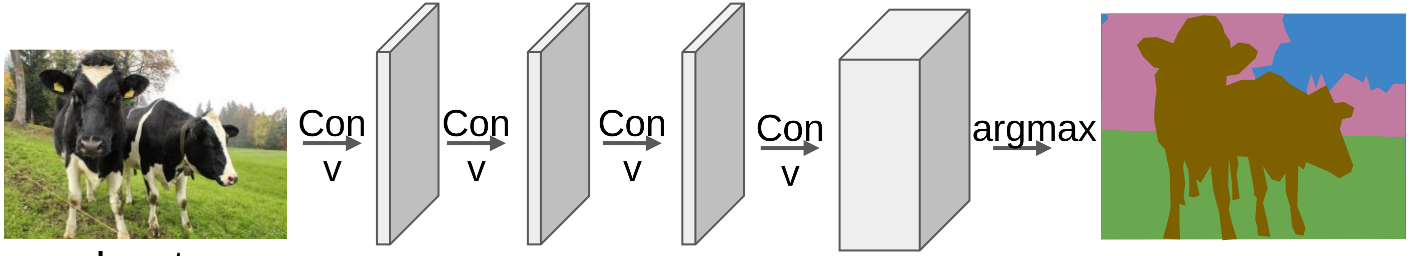
$$(F *_l k)(\mathbf{p}) = \sum_{\mathbf{s}+l\mathbf{t}=\mathbf{p}} F(\mathbf{s})\, k(\mathbf{t}). \tag{2}$$

We will refer to $*_l$ as a dilated convolution or an l-dilated convolution. The familiar discrete convolution $*$ is simply the 1-dilated convolution.

Fisher Yu and Vladlen Koltun. Multi-Scale Context Aggregation by Dilated Convolutions. ICLR, 2016.

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!

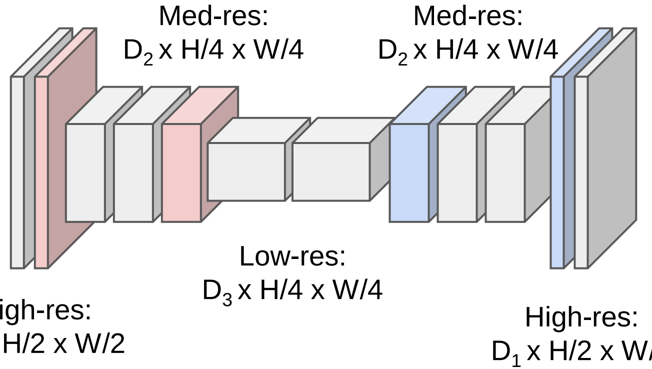
Long et al, “Fully convolutional networks for semantic segmentation”, CVPR 2015

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”

Bed of Nails

1	2
3	4

→

1	0	2	0
0	0	0	0
3	0	4	0
0	0	0	0

Input
C x 2 x 2

Output
C x 4 x 4

Nearest Neighbor

1	2
3	4

→

1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

Input
C x 2 x 2

Output
C x 4 x 4

Slide Credit: Justin Johnson and David Fouhey

Upsampling: Bilinear Interpolation

1	2
3	4

Input: C x 2 x 2

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

Output: C x 4 x 4

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

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

Slide Credit: Justin Johnson and David Fouhey

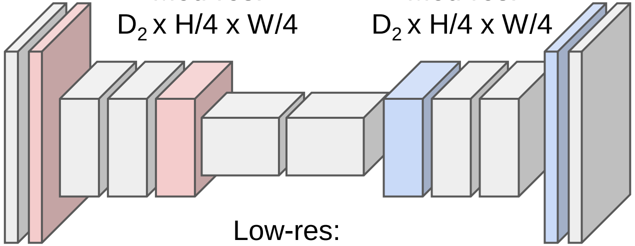
Fully Convolutional Network

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

Input:
3 x H x W

Med-res:
D_2 x H/4 x W/4

Med-res:
D_2 x H/4 x W/4



High-res:
D_1 x H/2 x W/2

Low-res:
D_3 x H/4 x W/4

High-res:
D_1 x H/2 x W/2

Prediction
s:
H x W

Downsampling:
Pooling, strided convolution

Upsampling:
???

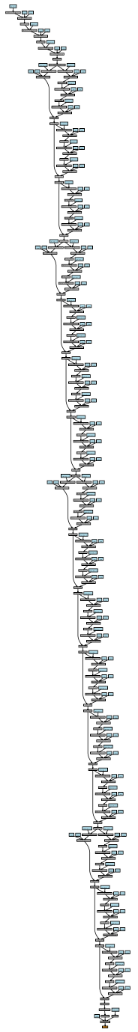
Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

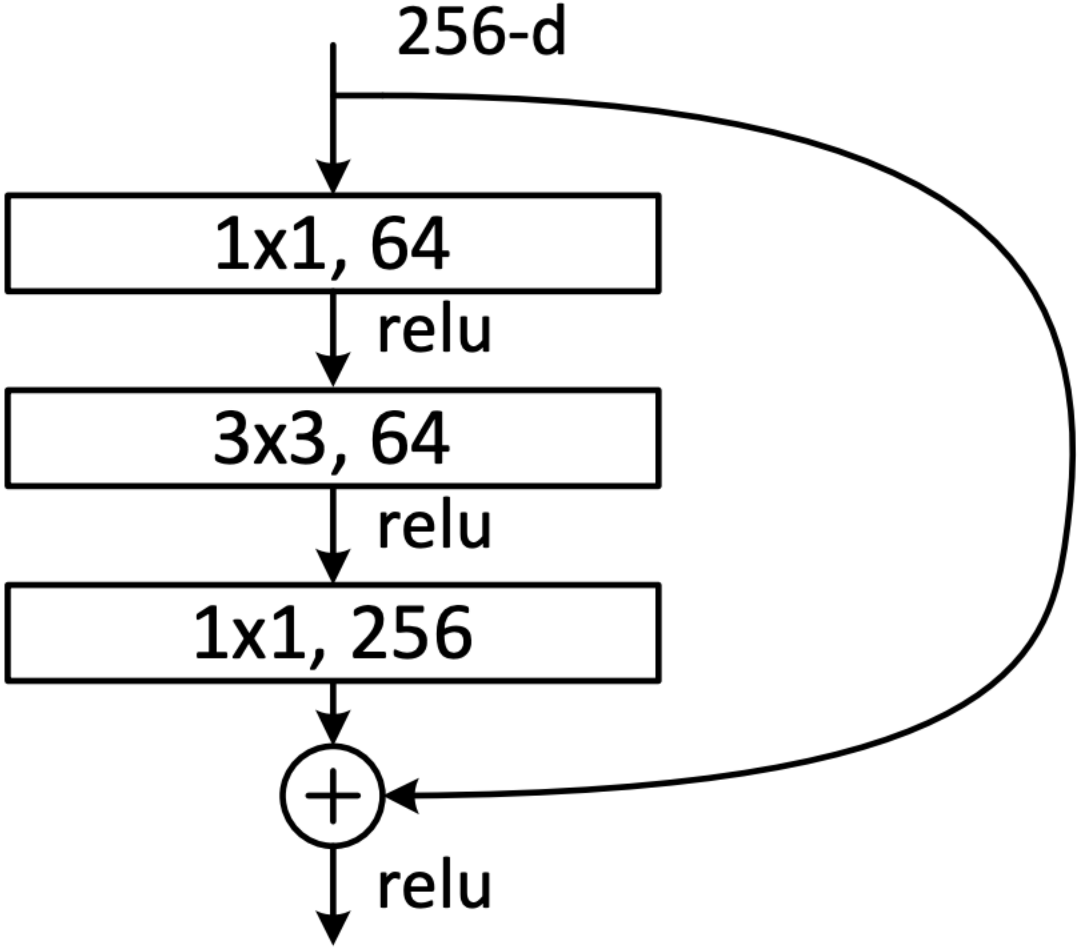
Slide Credit: Justin Johnson and David Fouhey

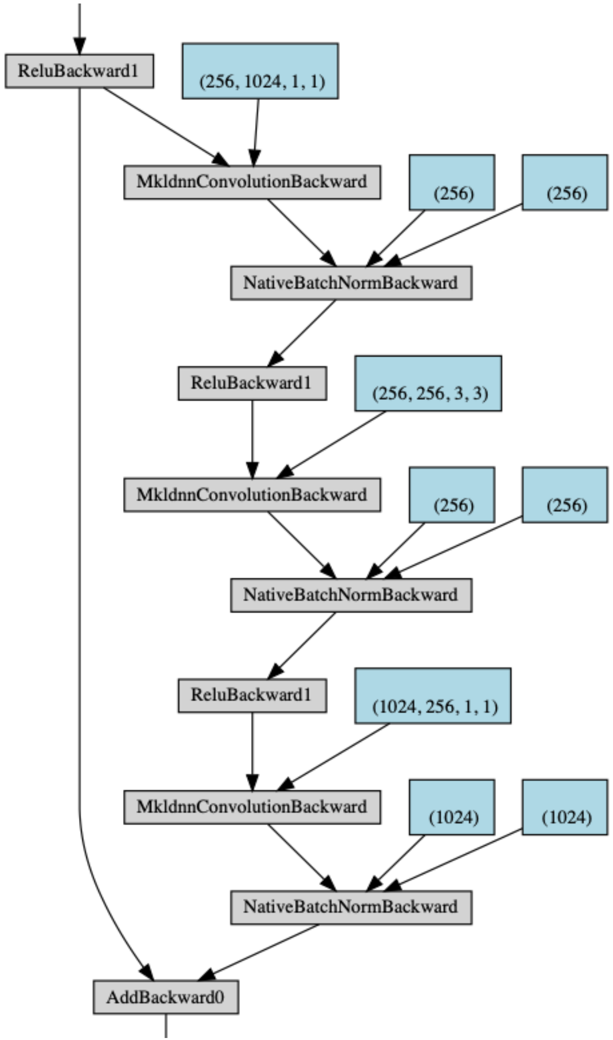
PSPNet

PSPNet uses a ResNet backbone

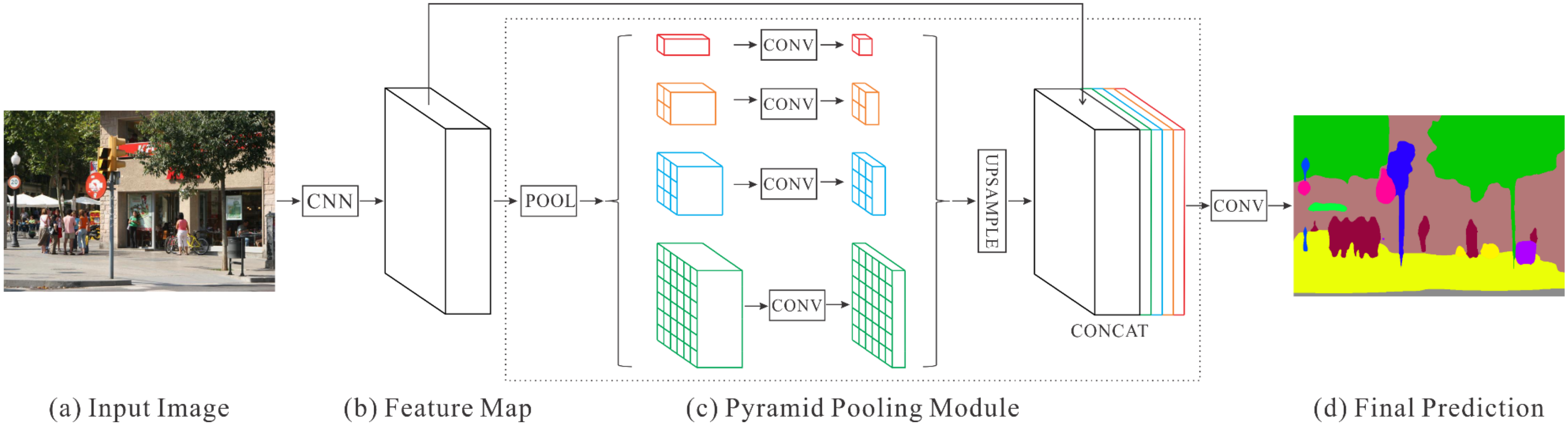
- 50, 101, or 152 Layers
- 50 Layers is already quite deep!




256-d
1x1, 64
relu
3x3, 64
relu
1x1, 256
relu


ReluBackward1
(256, 1024, 1, 1)
MkldnnConvolutionBackward
(256)
(256)
NativeBatchNormBackward
ReluBackward1
(256, 256, 3, 3)
MkldnnConvolutionBackward
(256)
(256)
NativeBatchNormBackward
ReluBackward1
(1024, 256, 1, 1)
MkldnnConvolutionBackward
(1024)
(1024)
NativeBatchNormBackward
AddBackward0

Pyramid Scene Parsing Network

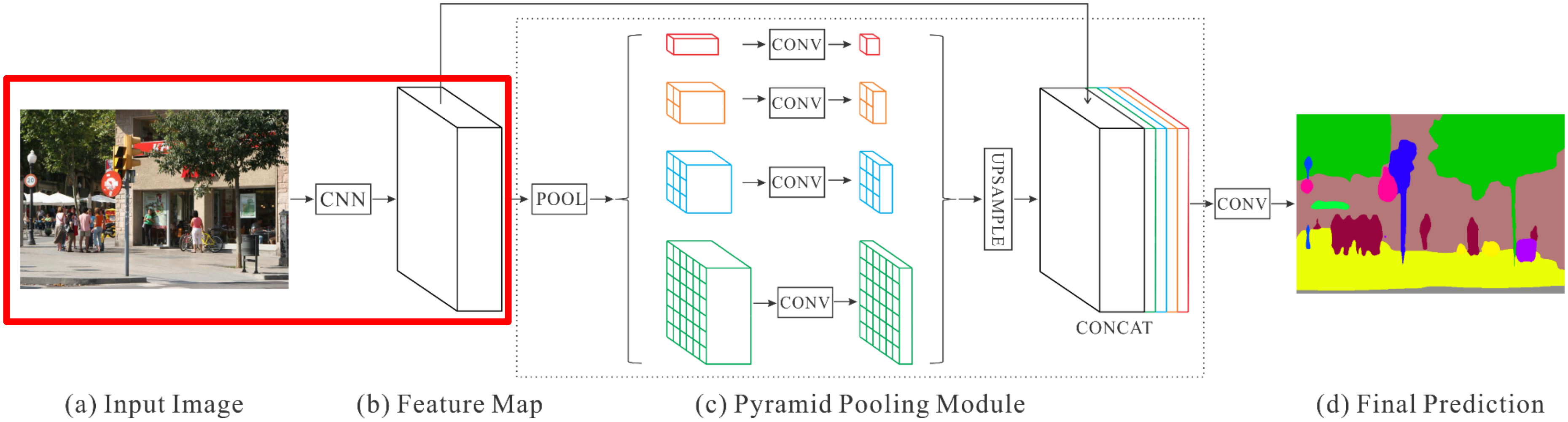


Framework overview of PSPNet

"Pyramid Scene Parsing Network", Zhao et al. CVPR 2017 [4,000+ citation]

Slide Credit: Hengshuang Zhao and Jiaya Jia

Pyramid Scene Parsing Network

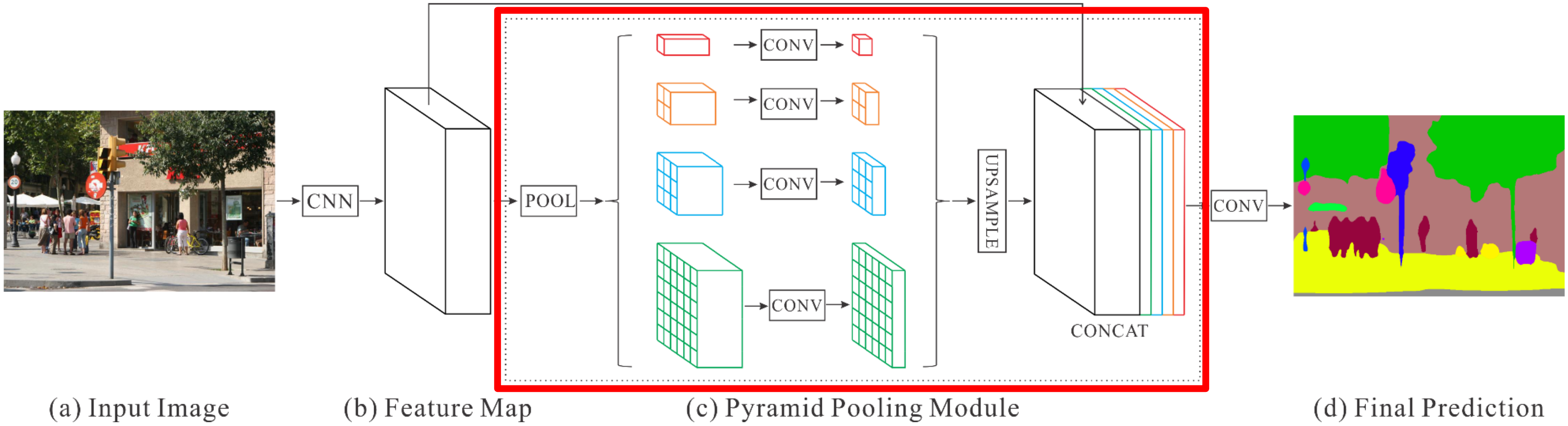


Regular feature extractor

"Pyramid Scene Parsing Network", Zhao et al. CVPR 2017 [4,000+ citation]

Slide Credit: Hengshuang Zhao and Jiaya Jia

Pyramid Scene Parsing Network

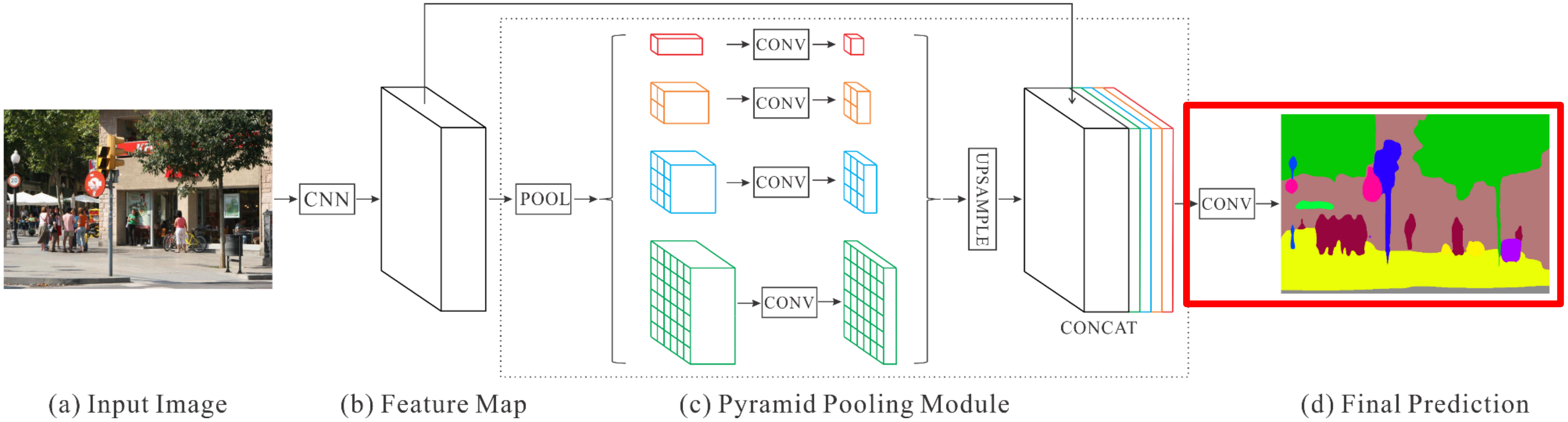


Context modeling: pyramid pooling module

"Pyramid Scene Parsing Network", Zhao et al. CVPR 2017 [4,000+ citation]

Slide Credit: Hengshuang Zhao and Jiaya Jia

Pyramid Scene Parsing Network

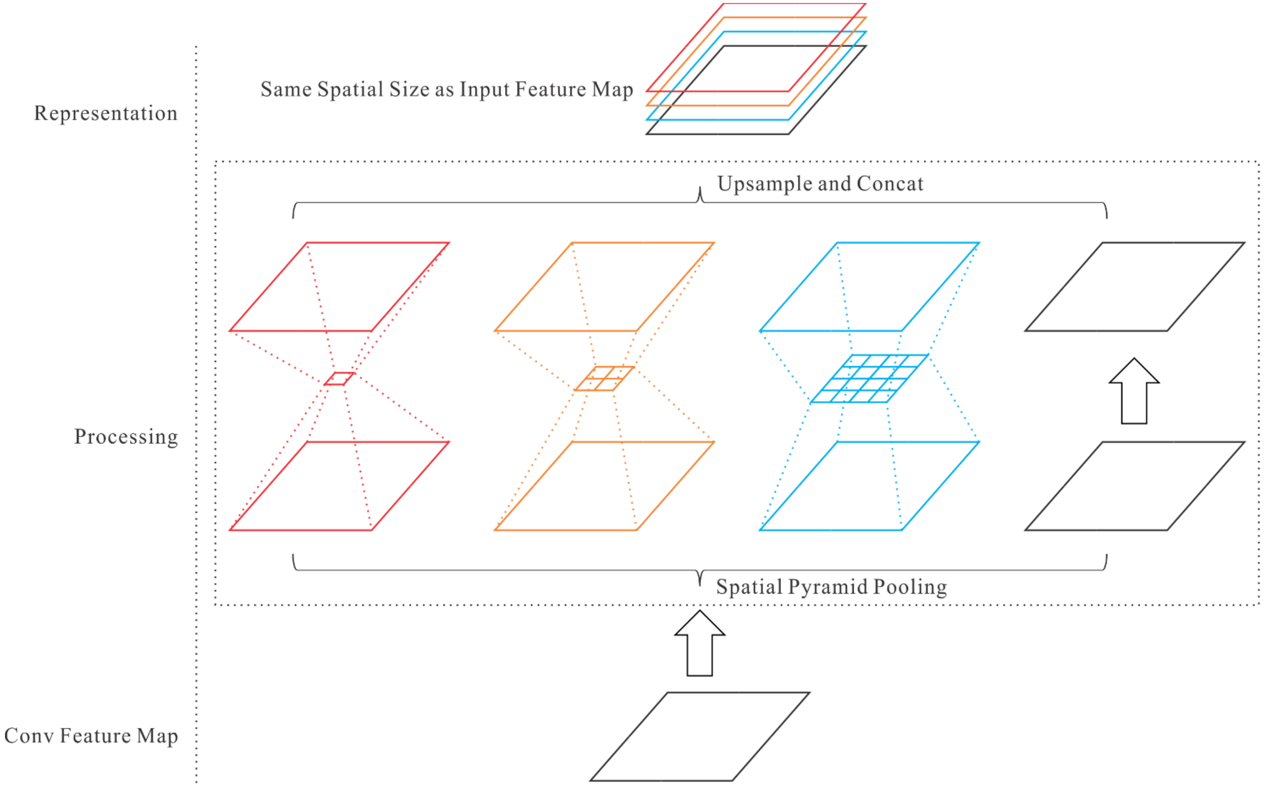


Convolutional classifier for pixel-wise prediction

"Pyramid Scene Parsing Network", Zhao et al. CVPR 2017 [4,000+ citation]

Slide Credit: Hengshuang Zhao and Jiaya Jia

Pyramid Pooling Module



PPM: spatial illustration

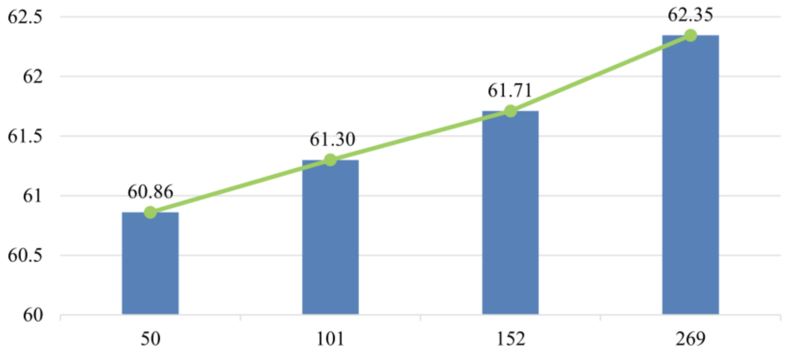
Slide Credit: Hengshuang Zhao and Jiaya Jia

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

Exceed by a large margin



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**

Get the highest accuracy on all 20 classes

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**

Outperform previous state-of-the-art by 8.4 points

Slide Credit: Hengshuang Zhao and Jiaya Jia

Algorithm Impact

Pyramid Scene Parsing Network 4071 2017
H Zhao, J Shi, X Qi, X Wang, J Jia
Computer Vision and Pattern Recognition (CVPR), 2017.

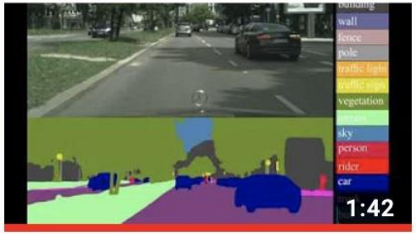
hszhao / PSPNet

Watch 63 | Star 1,134 | Fork 471

<> Code | Issues 75 | Pull requests 0 | Projects 0 | Security | Insights

Pyramid Scene Parsing Network, CVPR2017. https://hszhao.github.io/projects/pspnet

Pyramid Scene Parsing Network (CVPR 2017)
13K views • 2 years ago



Pyramid Scene Parsing Network (CVPR 2017)
13K views • 2 years ago

Pyramid Scene Parsing Network (CVPR 2017)
7.9K views • 2 years ago


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.

MSeg: A Composite Dataset for Multi-Domain Semantic Segmentation

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

intel®

Berkeley
UNIVERSITY OF CALIFORNIA

Georgia Tech

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

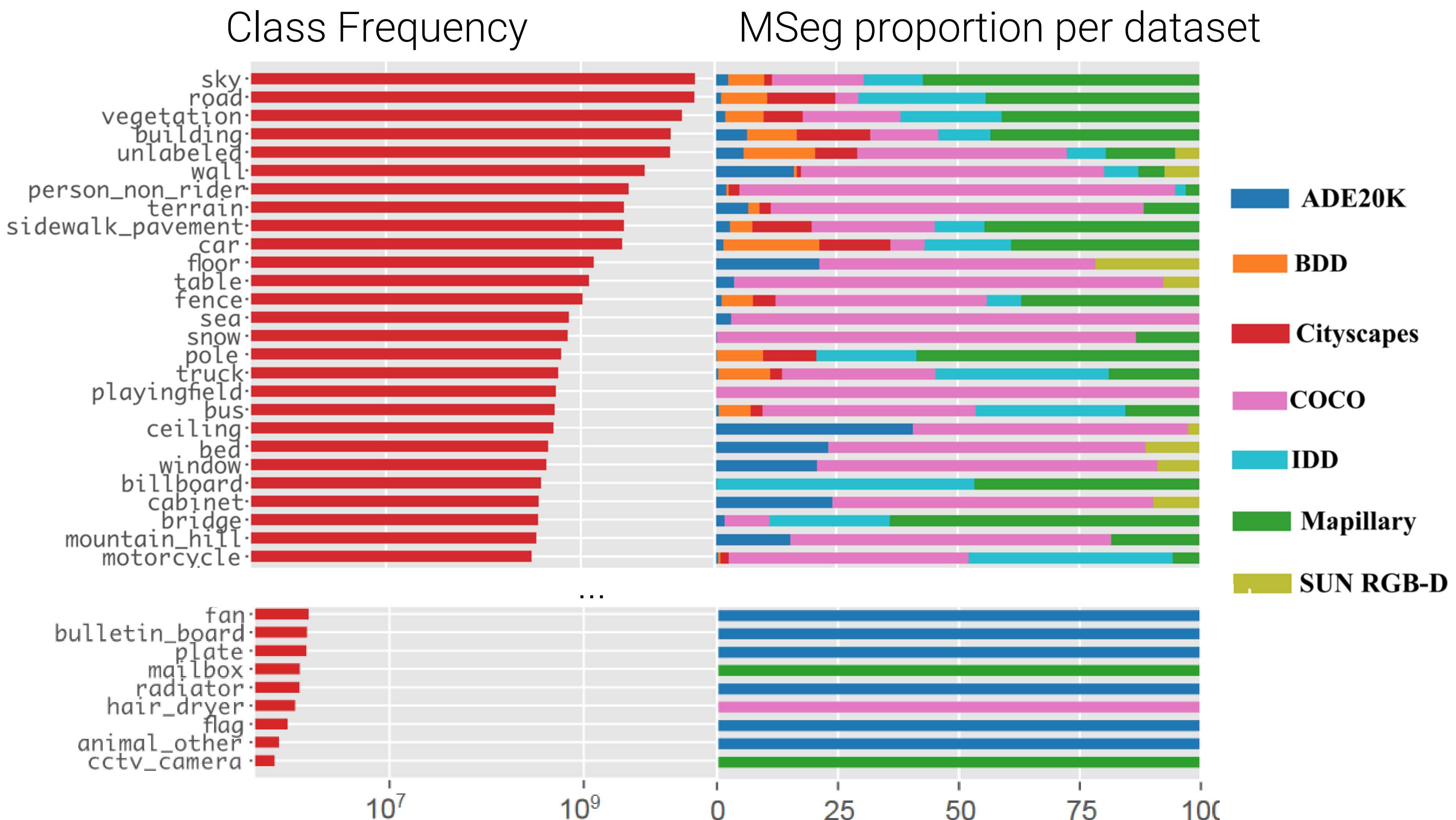
MSeg: A Composite Dataset for Multi-domain Semantic Segmentation

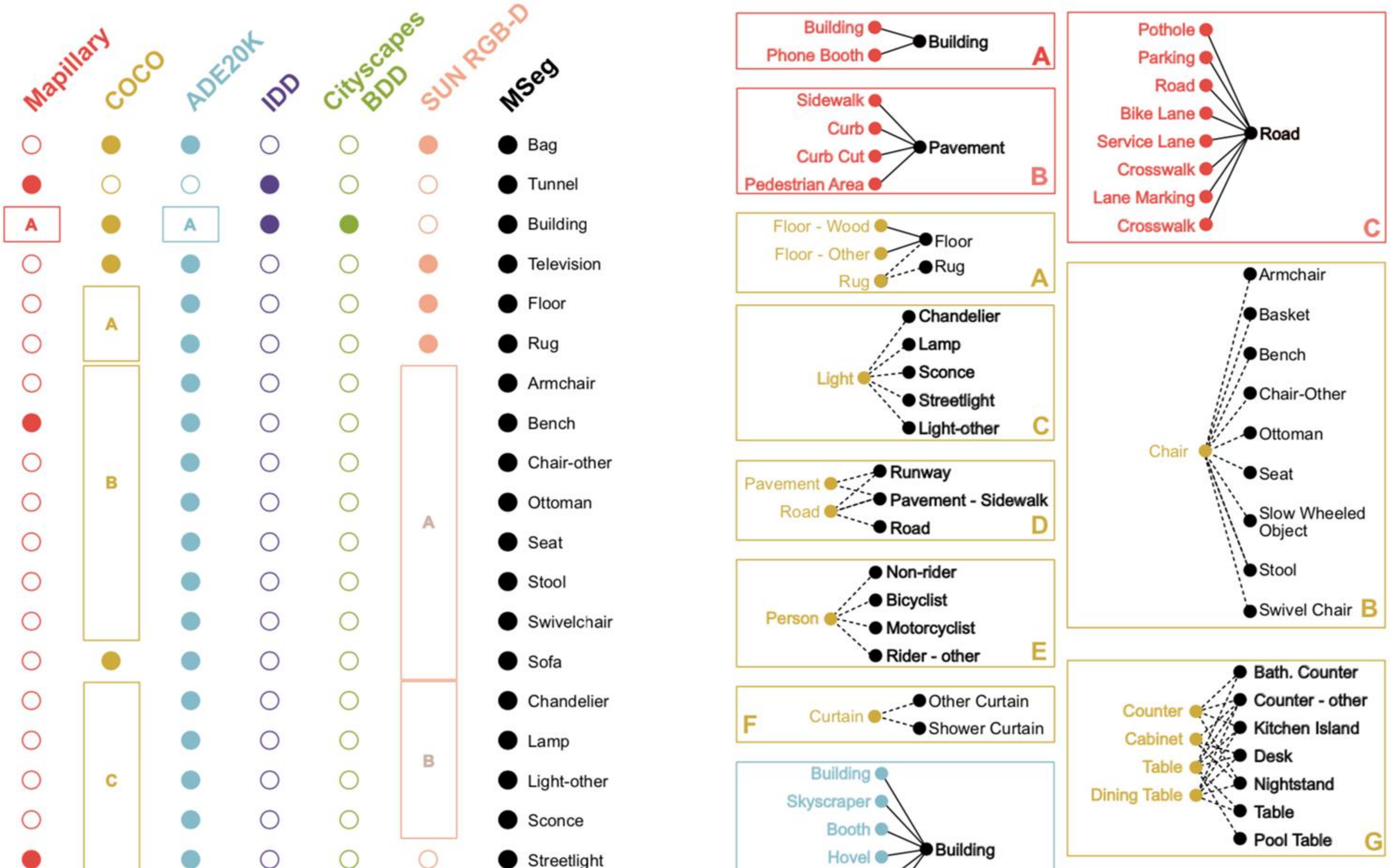
John Lambert*, Zhuang Liu*, Ozan Sener,
James Hays, Vladlen Koltun

intel

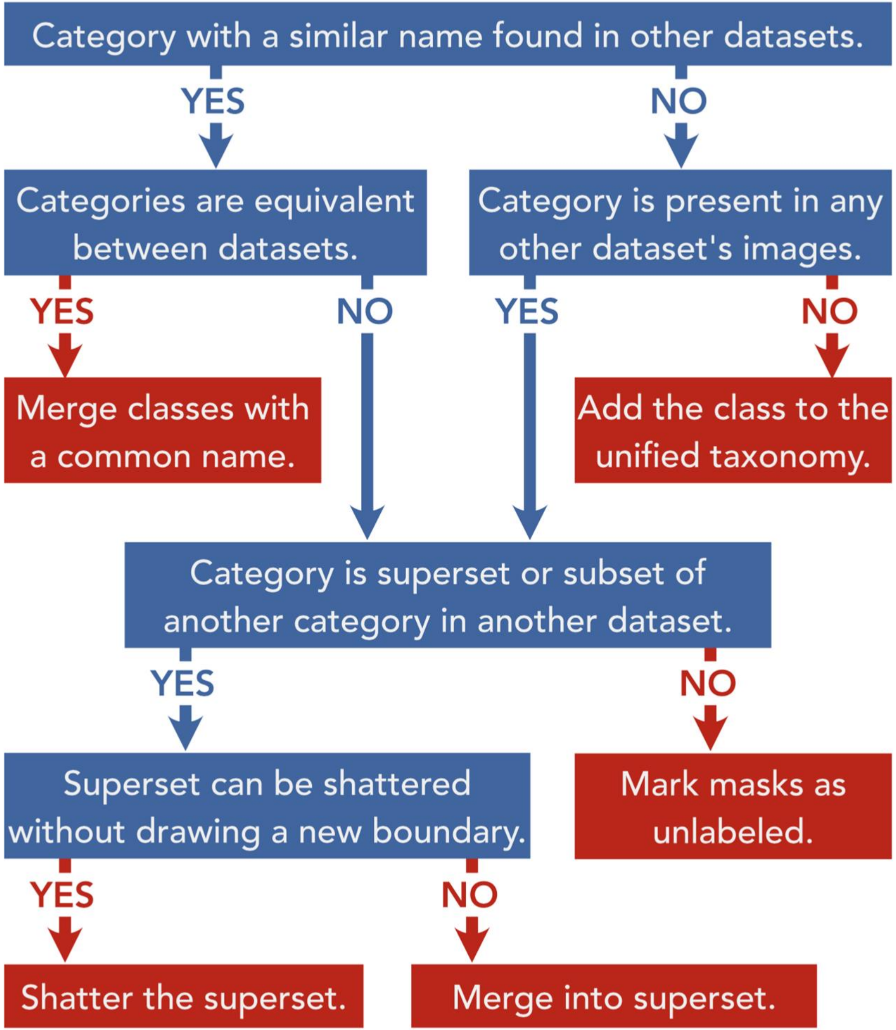
Berkeley
UNIVERSITY OF CALIFORNIA

Georgia Tech


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
...
fan
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
Sidewalk
Curb
Curb Cut
Pedestrian Area
Pavement
Pothole
Parking
Road
Bike Lane
Service Lane
Crosswalk
Lane Marking
Crosswalk
Road
Floor - Wood
Floor - Other
Rug
Floor
Rug
Light
Chandelier
Lamp
Sconce
Streetlight
Light-other
Pavement
Road
Runway
Pavement - Sidewalk
Road
D
Person
Non-rider
Bicyclist
Motorcyclist
Rider - other
E
Curtain
Other Curtain
Shower Curtain
F
Chair
Armchair
Basket
Bench
Chair-Other
Ottoman
Seat
Slow Wheeled Object
Stool
Swivel Chair
Counter
Cabinet
Table
Dining Table
Bath. Counter
Counter - other
Kitchen Island
Desk
Nightstand
Table
Pool Table
G
Building
Skyscraper
Booth
Hovel
Building

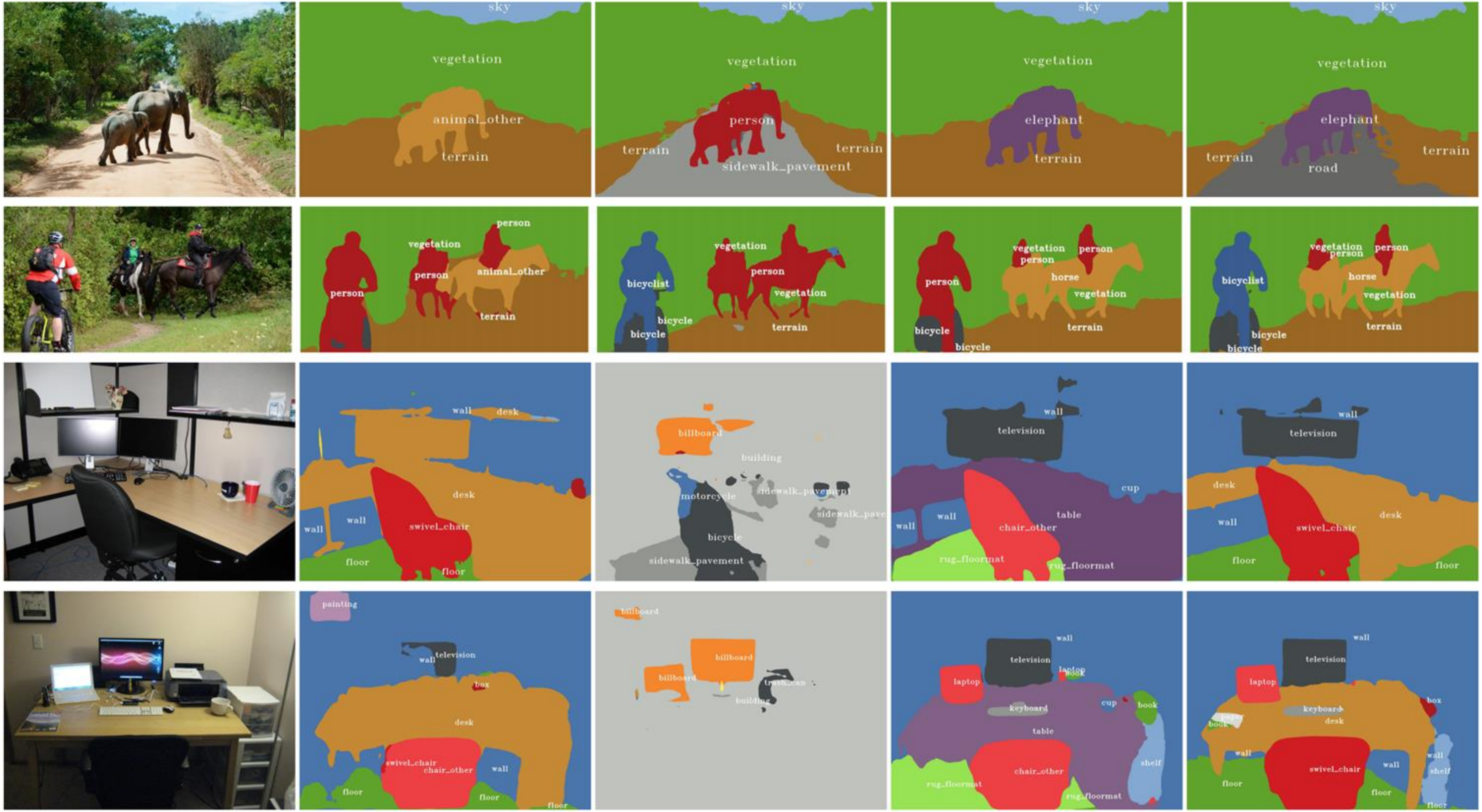
Georgia Tech



Generality and Robustness

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

WildDash benchmark

	Meta AVG mIoU	Seen WildDash data?
MSeg-1080 (Ours)	**48.3**	✗
LDN BIN-768 [4]	46.9	✓
LDN OE [4]	42.7	✓
DN169-CAT-DUAL	41.0	✓
AHiSS [34]	39.0	✗

Project 6