Lecture 13:
Spatial Localization and Image Segmentation

Administrivia

This Thursday (10/24) there will be no class.

Instead attend the MLFL seminar:

12-1pm, CS 150

Alex Wong, Yale University

The Know-How of Multimodal Depth Perception

There will be pizza!

https://www.cics.umass.edu/category/machine-learning-and-friends-lunch

Abstract

Training deep neural networks requires tens of thousands to millions of examples, so curating multimodal vision datasets amounts to numerous man-hours; tasks like depth estimation require an even more massive effort. I will introduce an alternative form of supervision that leverages multi-sensor validation as an unsupervised (or self-supervised) training objective for depth estimation. To address its ill-posedness, I will show how one can leverage multimodal inputs in the choice of regularizers, which can play a role in model complexity, speed, generalization, as well as adaptation to test-time (possibly adverse) environments. Additionally, I will discuss the current limitations of data augmentation procedures used during unsupervised training, which involves reconstructing the inputs as the supervision signal, and detail a method that allows one to scale up and introduce previously inviable augmentations to boost performance. Finally, I will show how one can scalably expand the number of modalities supported by multimodal models and demonstrate their use in a number of downstream semantic tasks.

Bio

Alex Wong is an Assistant Professor in the department of Computer Science and the director of the Vision Laboratory at Yale University. He also serves as the Director of AI (consulting capacity) for Horizon Surgical Systems. Prior to joining Yale, he was an Adjunct Professor at Loyola Marymount University (LMU) from 2018 to 2020. He received his Ph.D. in Computer Science from the University of California, Los Angeles (UCLA) in 2019 and was co-advised by Stefano Soatto and Alan Yuille. He was previously a post-doctoral research scholar at UCLA under the guidance of Soatto. His research lies in the intersection of machine learning, computer vision, and robotics and largely focuses on multimodal 3D reconstruction, robust vision under adverse conditions, and unsupervised learning. His work has received the outstanding student paper award at the Conference on Neural Information Processing Systems (NeurIPS) 2011 and the best paper award in robot vision at the International Conference on Robotics and Automation (ICRA) 2019.

UMassAmherst | Home | Lectures | Notes | Assignments | Policies | Project | Office Hours

Lecture	Thursday, Oct 17	Convnets for spatial localization I: Object detection	[slides]
Lecture	Tuesday, Oct 22	Convnets for spatial localization II: Image segmentation	
Lecture	Thursday, Oct 24	Guest Lecture: Alex Wong (Yale)	
Lecture	Tuesday, Oct 29	Understanding and visualizing convnets	
Lecture	Thursday, Oct 31	Guest lecture: Boqing Gong (Google, BU)	
	Tuesday, Nov 5	No class, Election Day	
Lecture	Thursday, Nov 7	Neural texture synthesis and style transfer	
Lecture	Tuesday, Nov 12	Generative AI	
Lecture	Thursday, Nov 14	Recurrent networks and transformers	
Lecture	Tuesday, Nov 19	Self-supervised learning	
Lecture	Thursday, Nov 21	Guest lecture: Chen Sun (Brown)	
Lecture	Tuesday, Nov 26	No regular class, work on projects	
	Thursday, Nov 28	No class, Thanksgiving Break	
Project Presentation	Tuesday, Dec 3	Group 1	
Project Presentation	Thursday, Dec 5	Group 2	
Project Presentation	Tuesday, Dec 10	Group 3	

MLFL talks (12-1pm, CS 150)

Project presentations

Order will be randomized
Attendance required all three days

Upcoming deadlines

gradescope by Turnitin

COMPSCI 682

Neural Networks: A Modern Introduction

- Dashboard
- Assignments
- Roster
- Extensions
- Course Settings

Instructors

- Subhransu Maji
- Chuang Gan

9 Assignments

Name	Points	Released	Due (EDT)
Project Final Report & Presentation	0.0	OCT 18, 2024 1:48 PM	DEC 2, 2024 11:59 PM
Assignment 3 (code)	100.0	OCT 18, 2024 1:53 PM	NOV 26, 2024 11:59 PM Late Due Date: NOV 29, 2024 11:59 PM
Assignment 3	0.0	OCT 18, 2024 1:51 PM	NOV 26, 2024 11:59 PM Late Due Date: NOV 29, 2024 11:59 PM
Project Milestone	0.0	OCT 18, 2024 1:42 PM	OCT 31, 2024 11:59 PM
Assignment 2 (Code)	100.0	OCT 4, 2024 12:30 PM	OCT 24, 2024 11:59 PM Late Due Date: OCT 27, 2024 11:59 PM
Assignment 2	100.0	OCT 4, 2024 12:30 PM	OCT 24, 2024 11:59 PM Late Due Date: OCT 27, 2024 11:59 PM

Next Thursday!

Thursday!

Project milestone

Your project milestone report should be between **2 - 3 pages** following the structure below (standard conference format). The course website has a latex template.

- **Title, Author(s)**
- **Abstract:** A short summary of the approach and expected results.
- **Introduction:** This section introduces your problem, motivation, and the overall plan. It should describe your problem precisely specifying the dataset to be used, expected results and evaluation.
- **Related work:** A literature survey of past work on this topic. Introduce the baselines you will compare to and the weakness you plan to address. *This section should be nearly complete.*
- **Technical approach**: Describe the methods you intend to apply to solve the given problem.
- **Intermediate/Preliminary Results:** State and evaluate your results upto the milestone.

Submission: Please upload a PDF file to Gradescope. Please coordinate with your teammates and **submit only under ONE of your accounts**, and add your teammates on Gradescope.

Your final report can continue to flesh out these sections.

Computer Vision Tasks

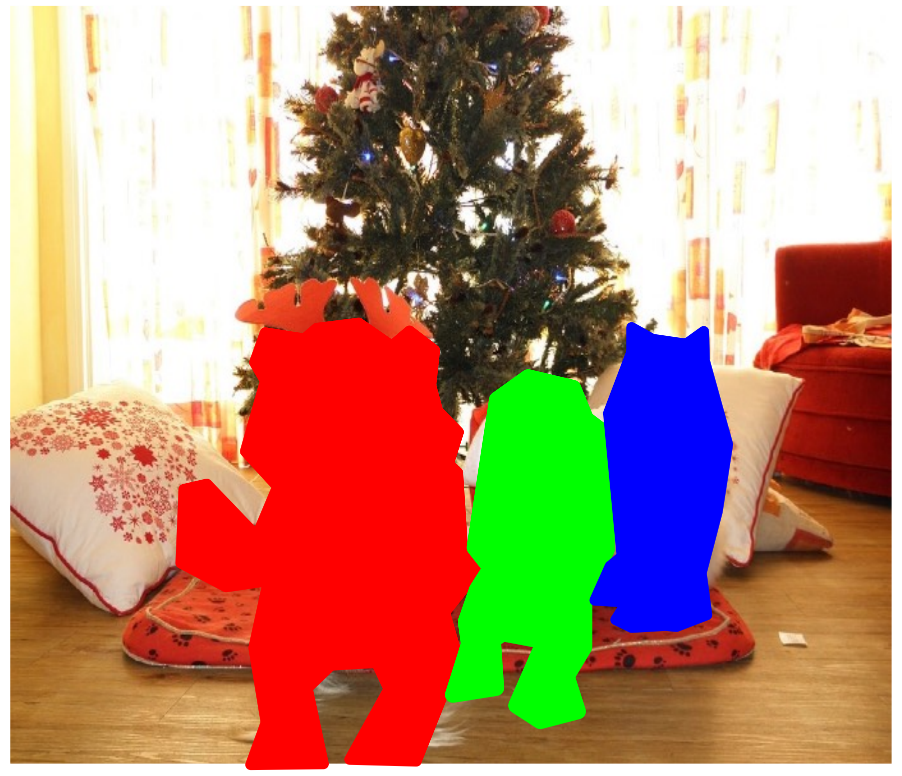
Classification

CAT

Object Detection

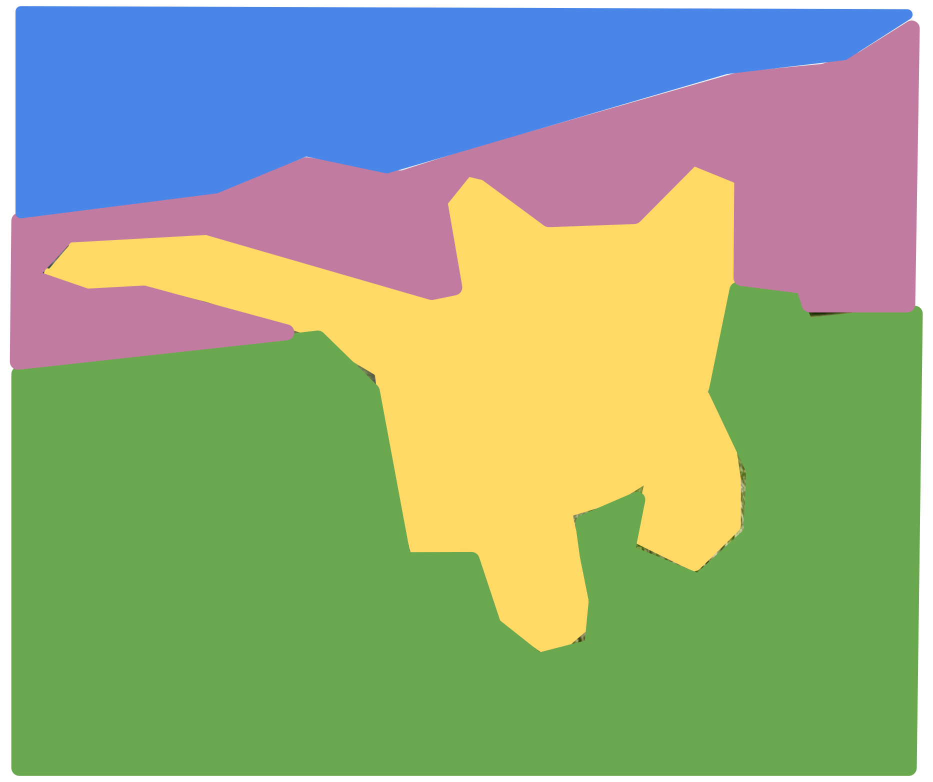
DOG, DOG, CAT

Instance Segmentation



DOG, DOG, CAT

Semantic Segmentation



GRASS, CAT, TREE, SKY

No spatial extent

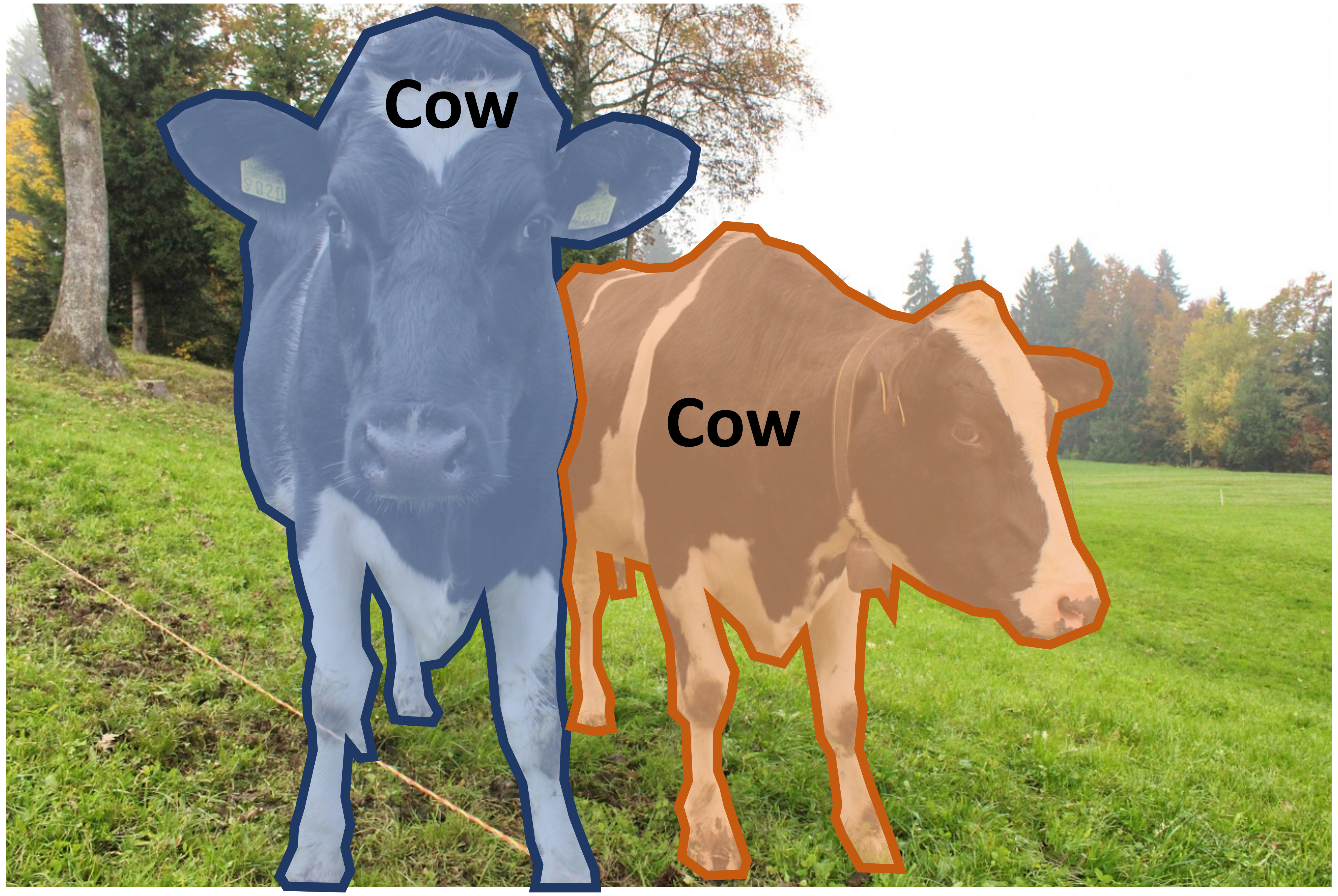
Multiple Objects

No objects, just pixels

Slide adapted from: D Fouhey & J Johnson

Instance segmentation

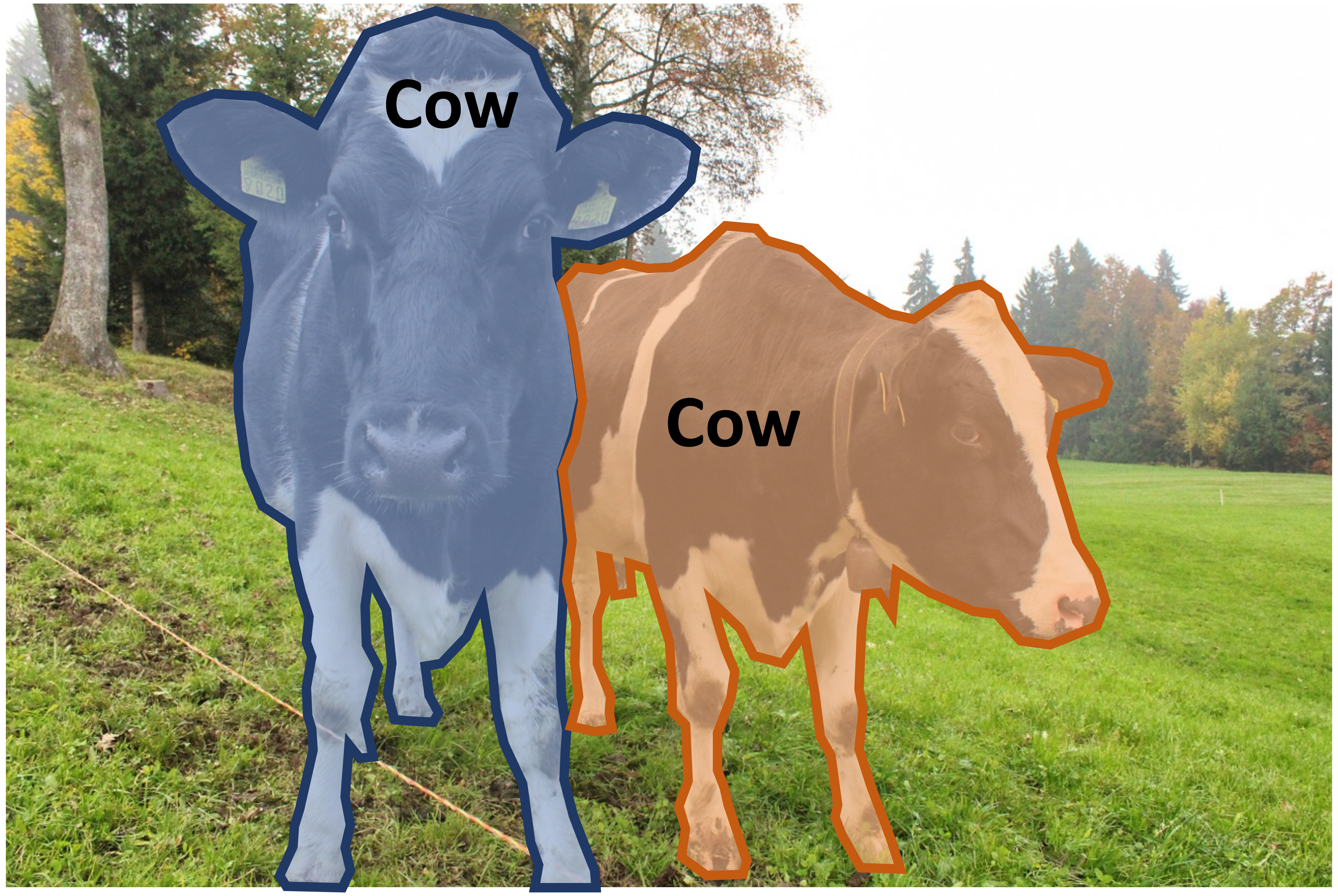
Instance Segmentation: Detect all objects in the image, and identify the pixels that belong to each object



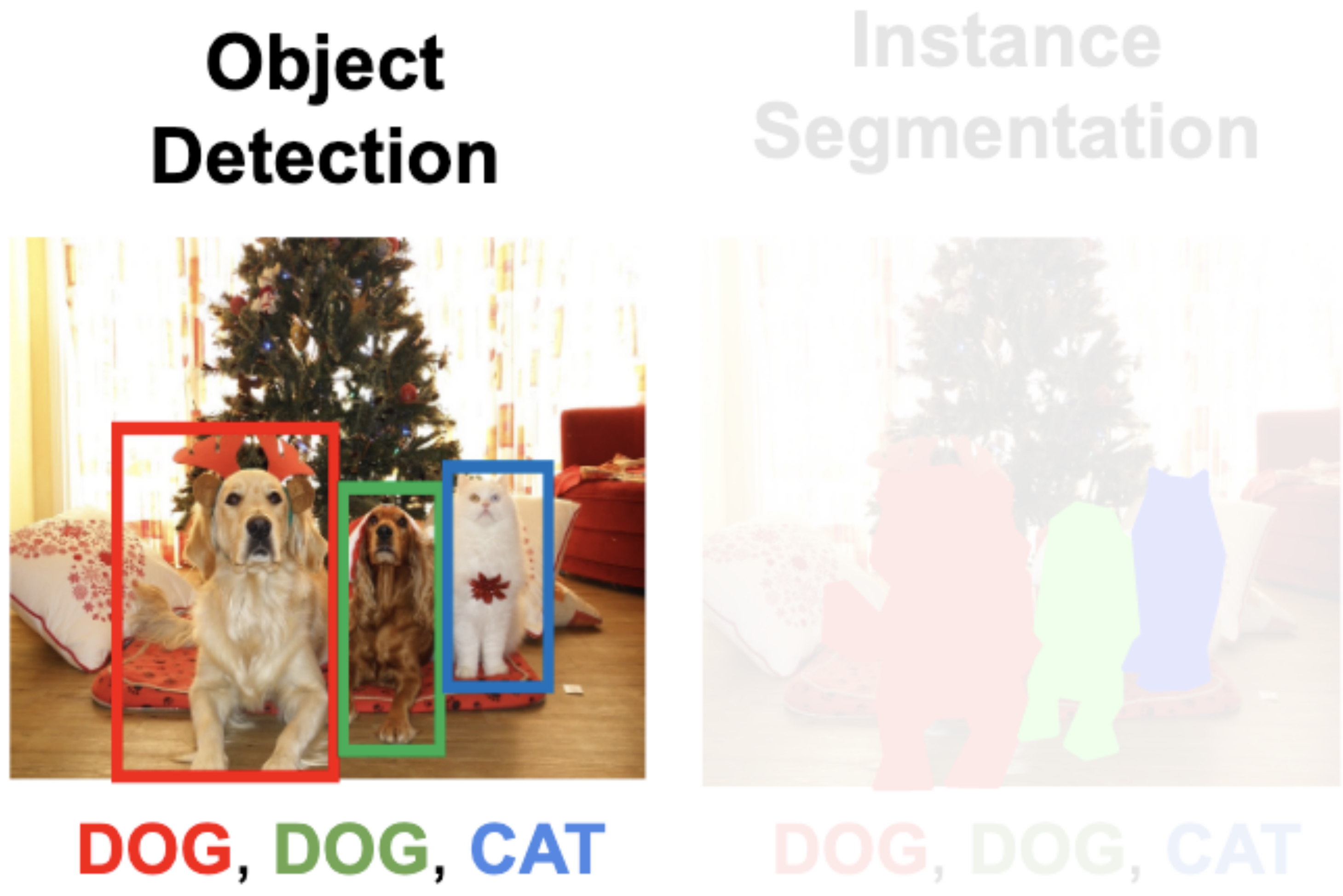
Instance segmentation

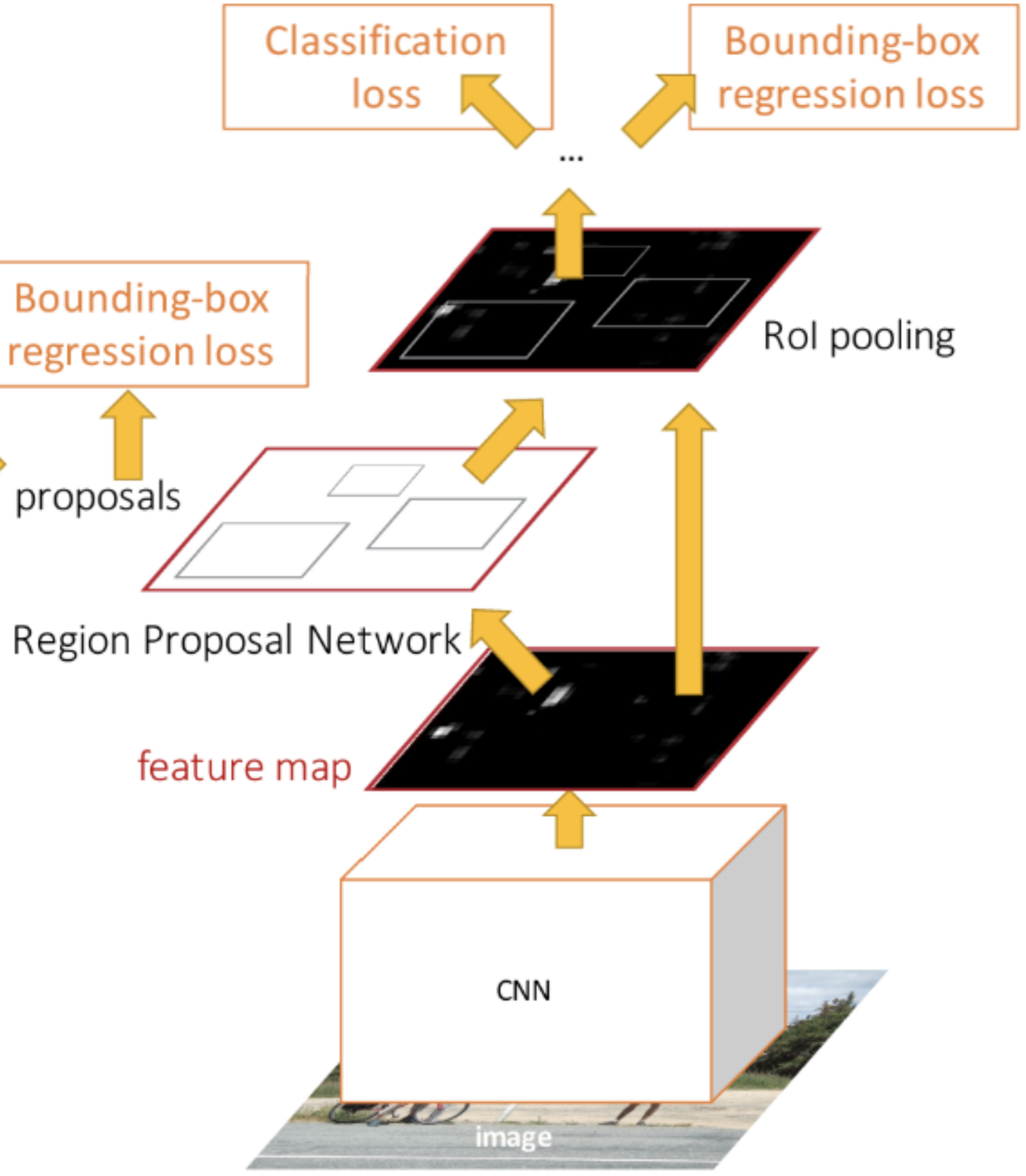
Instance Segmentation: Detect all objects in the image, and identify the pixels that belong to each object

Approach: Perform object detection, then predict a segmentation mask for each object!



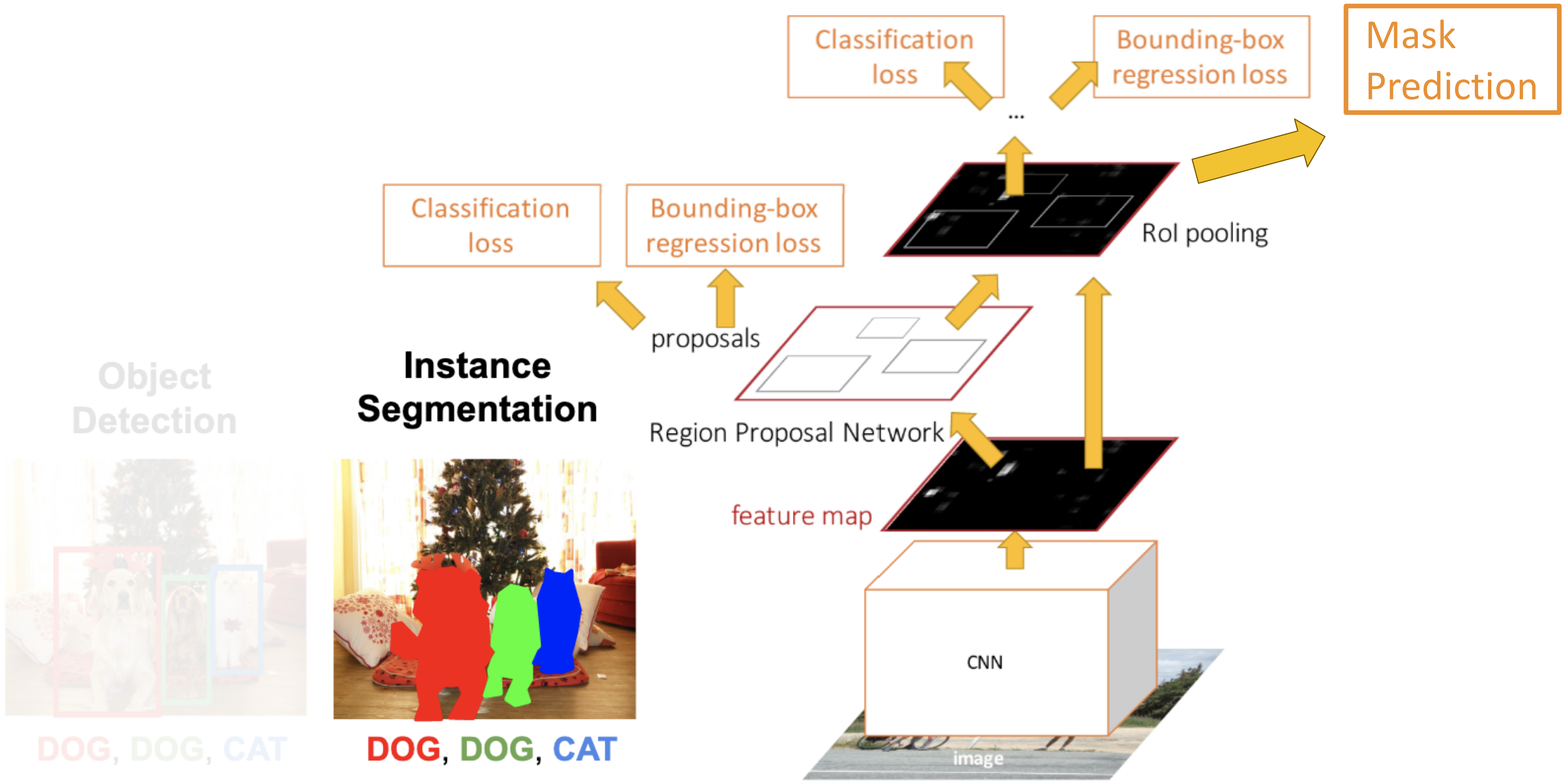
Object Detection: Faster R-CNN





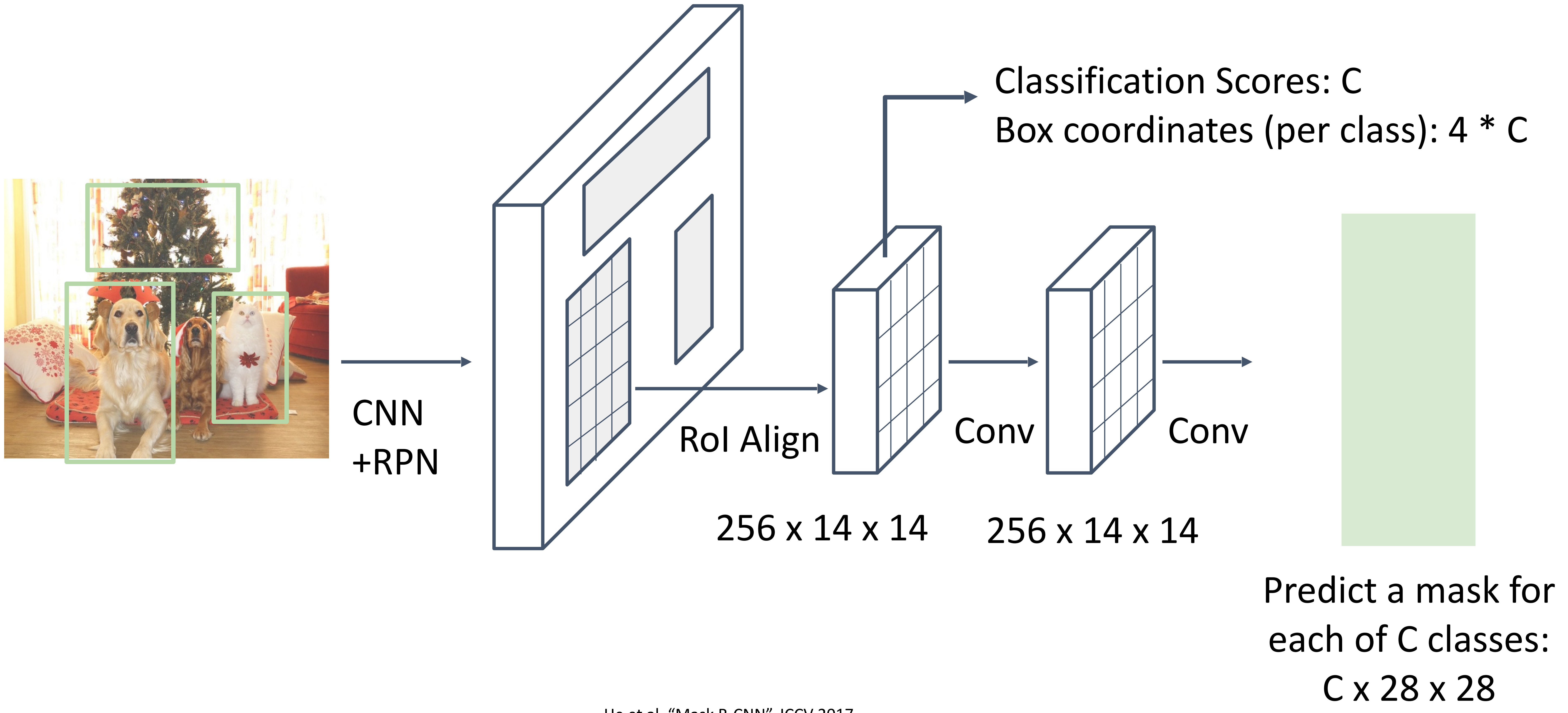
Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NeurIPS 2015

Instance Segmentation: Mask R-CNN



He et al, "Mask R-CNN", ICCV 2017

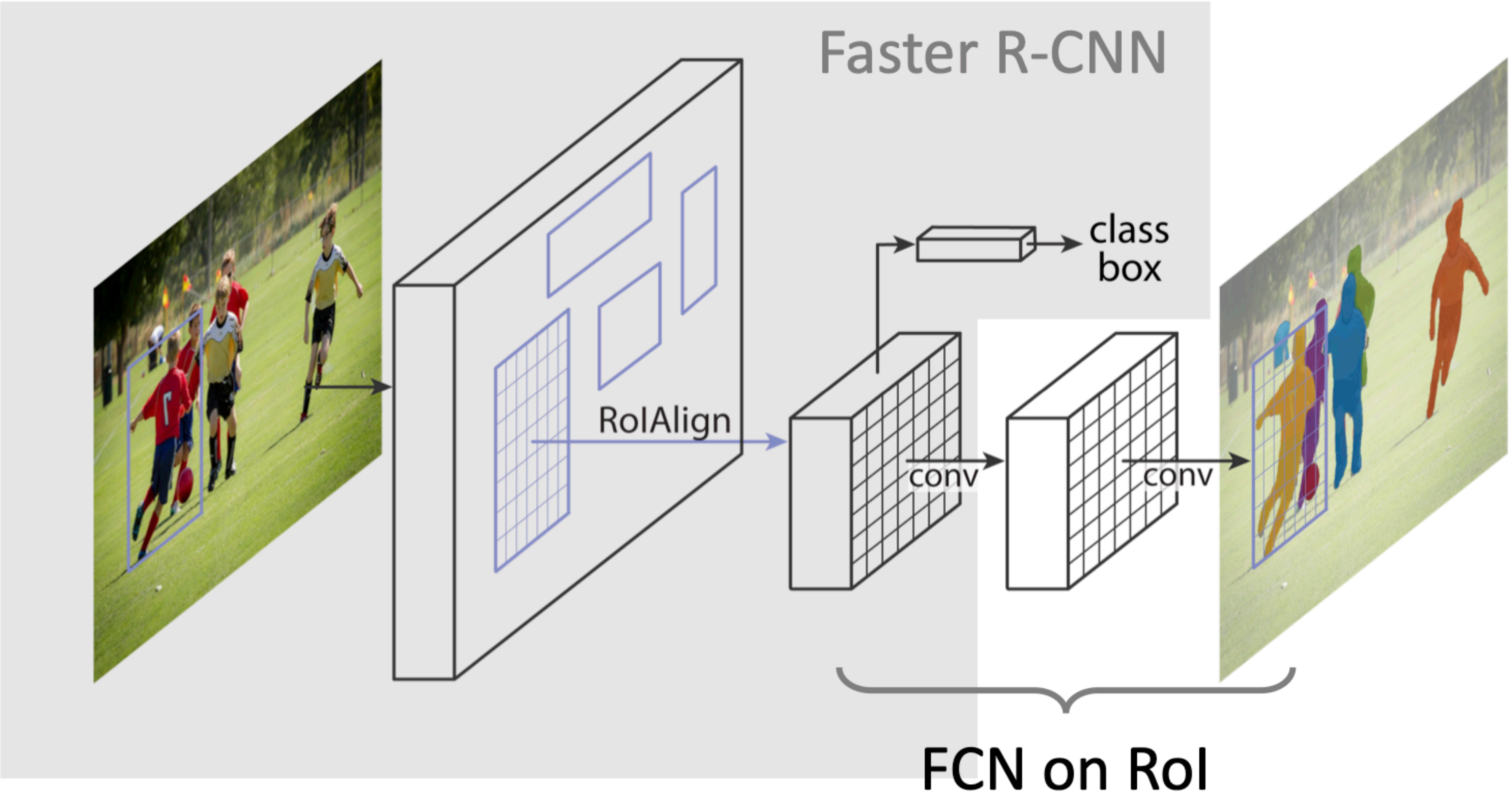
Mask R-CNN



He et al, "Mask R-CNN", ICCV 2017

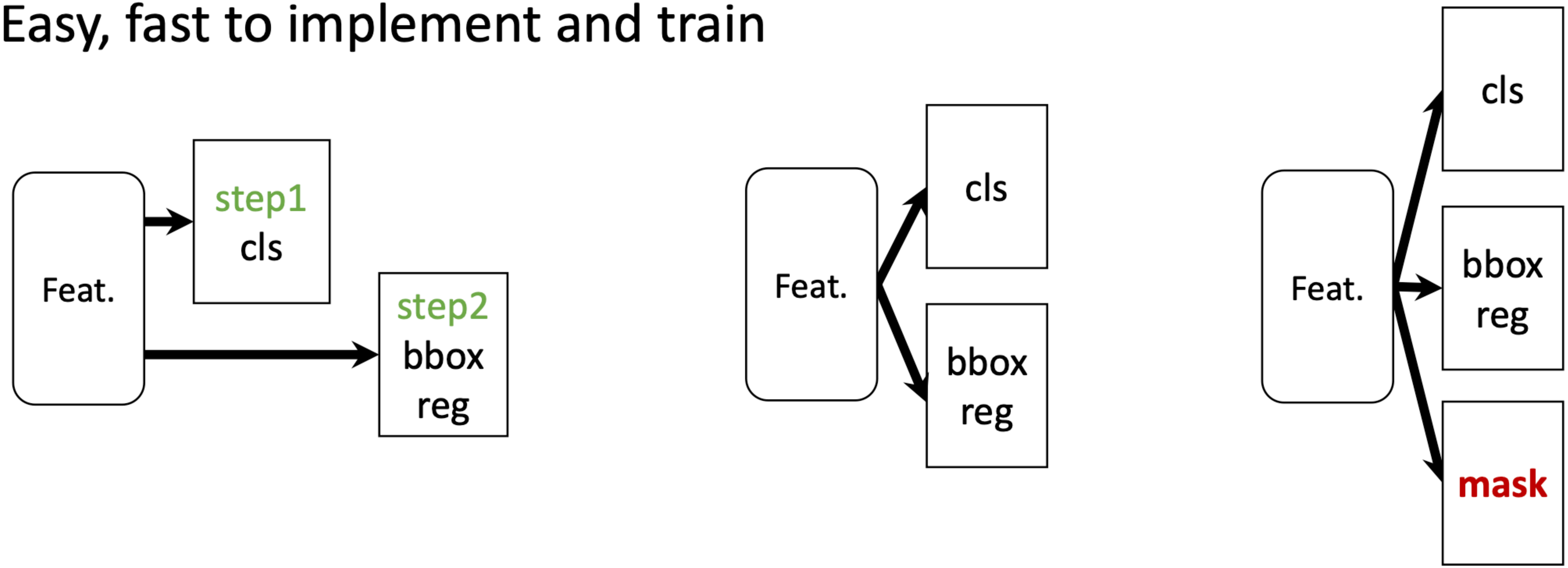
Mask R-CNN

- Mask R-CNN = **Faster R-CNN** with **FCN** on RoIs



Parallel Heads

- Easy, fast to implement and train

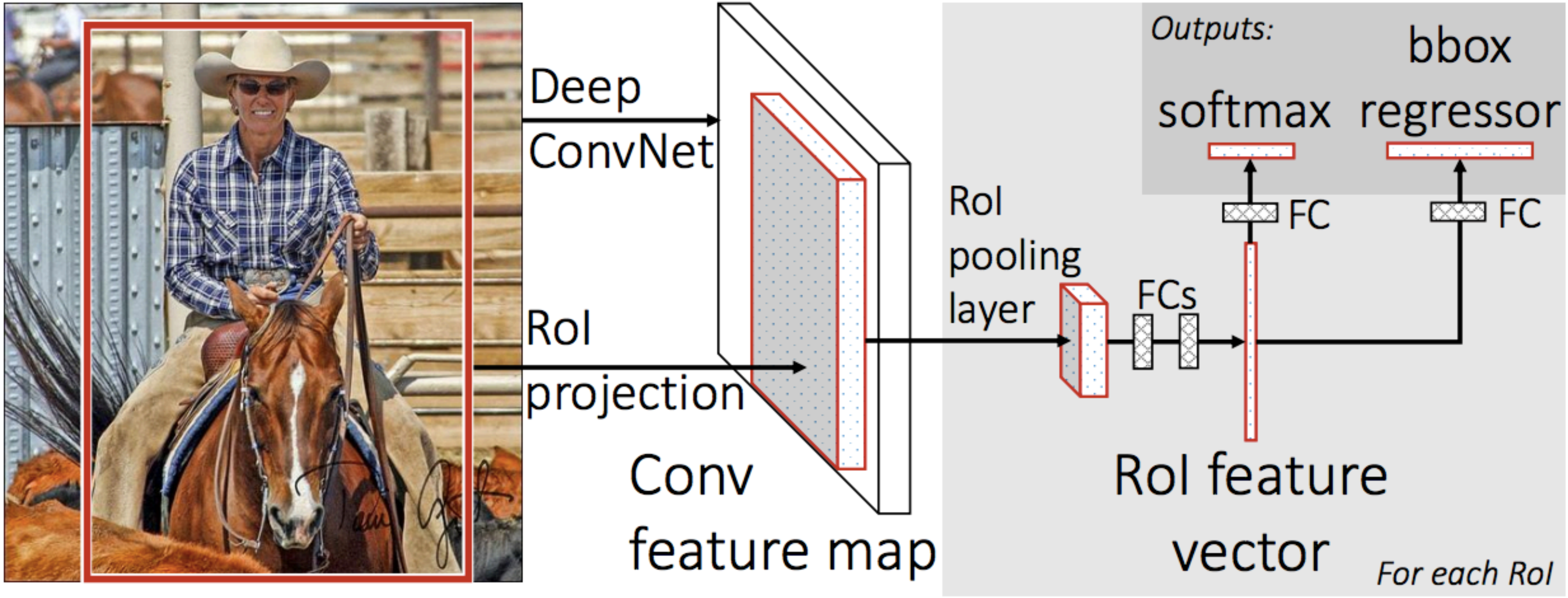


(slow) R-CNN

Fast/er R-CNN

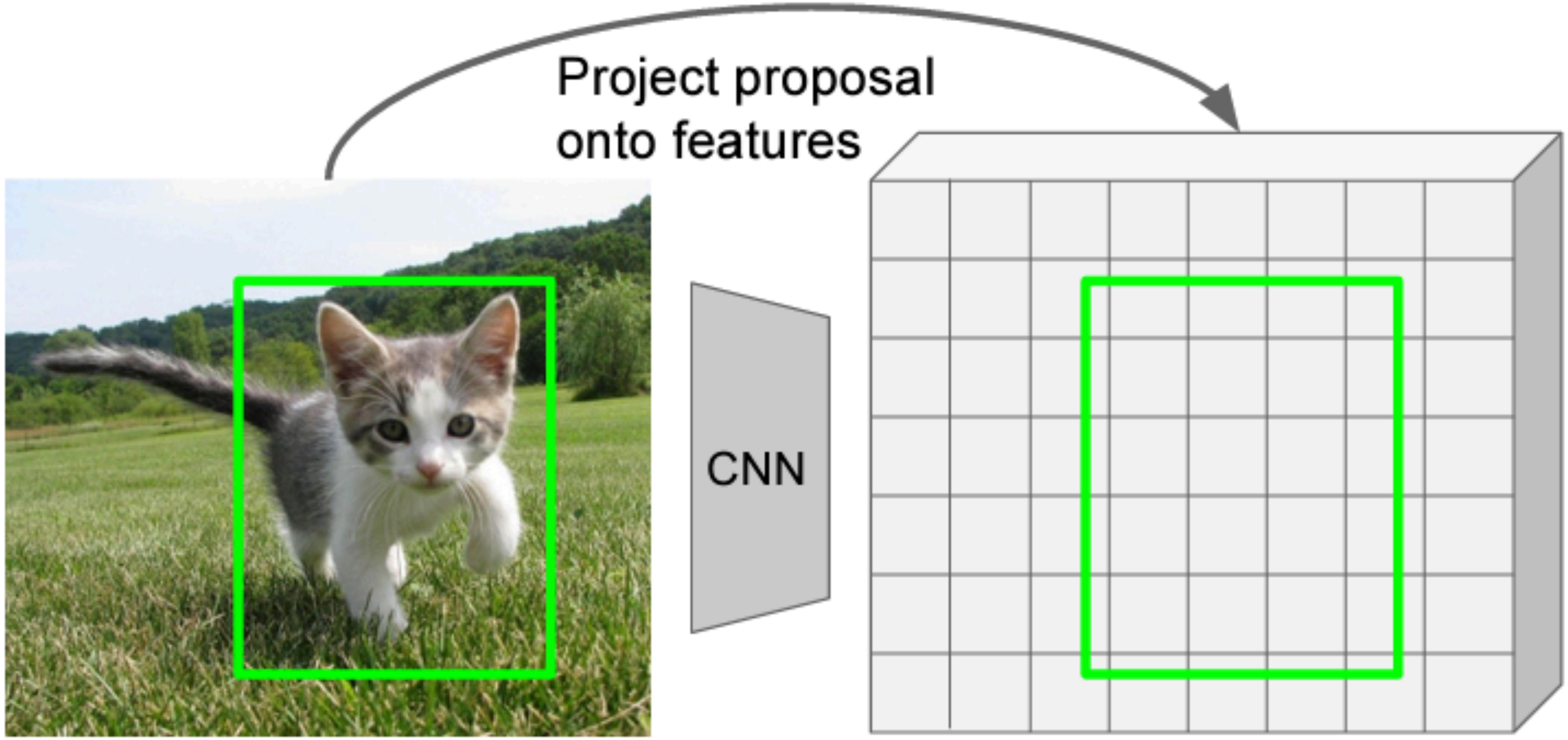
Mask R-CNN

RoIPool and RoIAlign



R. Girshick, Fast R-CNN, ICCV 2015

Cropping Features: RoI Pool



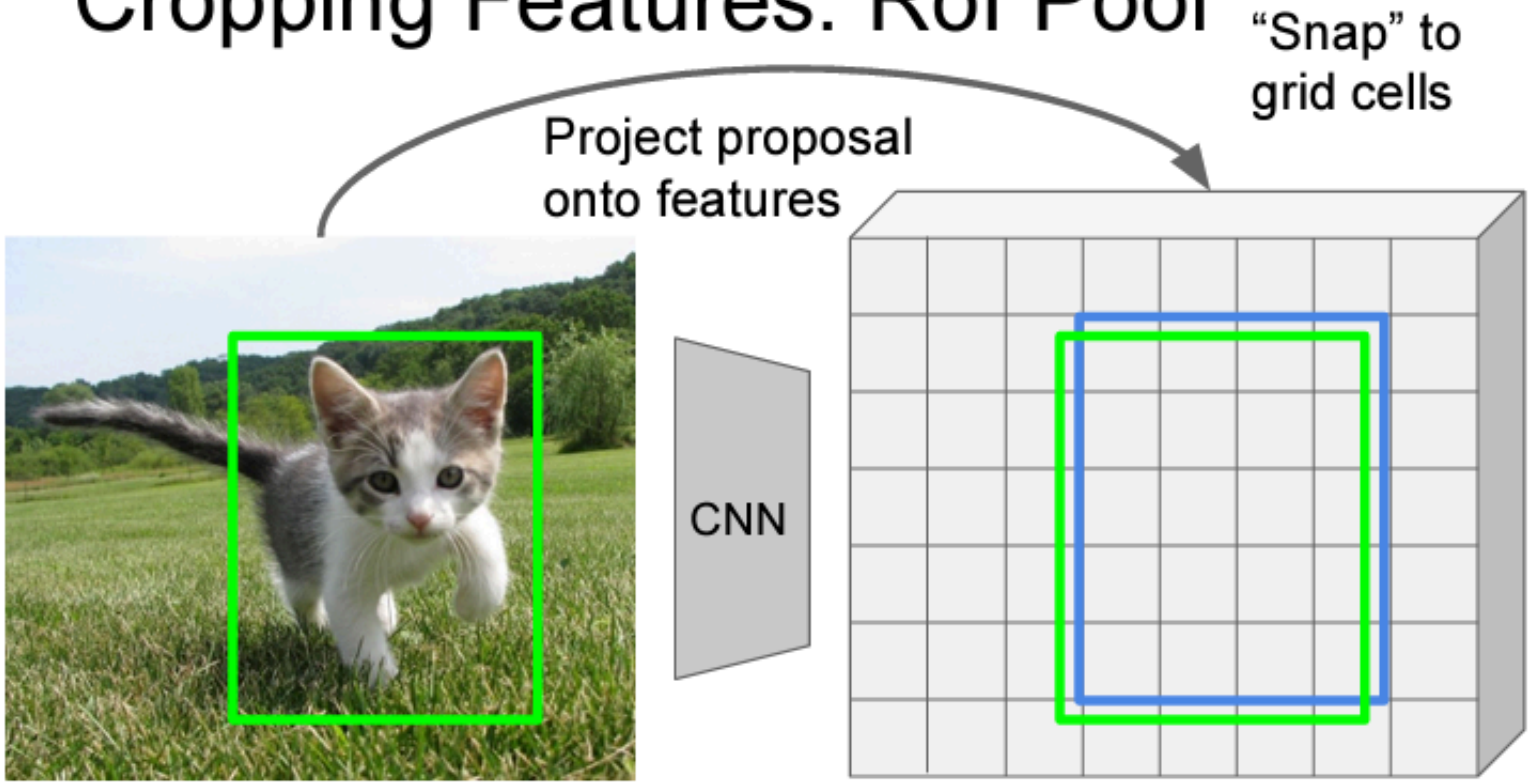
Input Image
(e.g. 3 x 640 x 480)

Image features: C x H x W
(e.g. 512 x 20 x 15)

Girshick, "Fast R-CNN", ICCV 2015.

Girshick, "Fast R-CNN", ICCV 2015.

Cropping Features: RoI Pool



Input Image
(e.g. 3 x 640 x 480)

Image features: C x H x W
(e.g. 512 x 20 x 15)

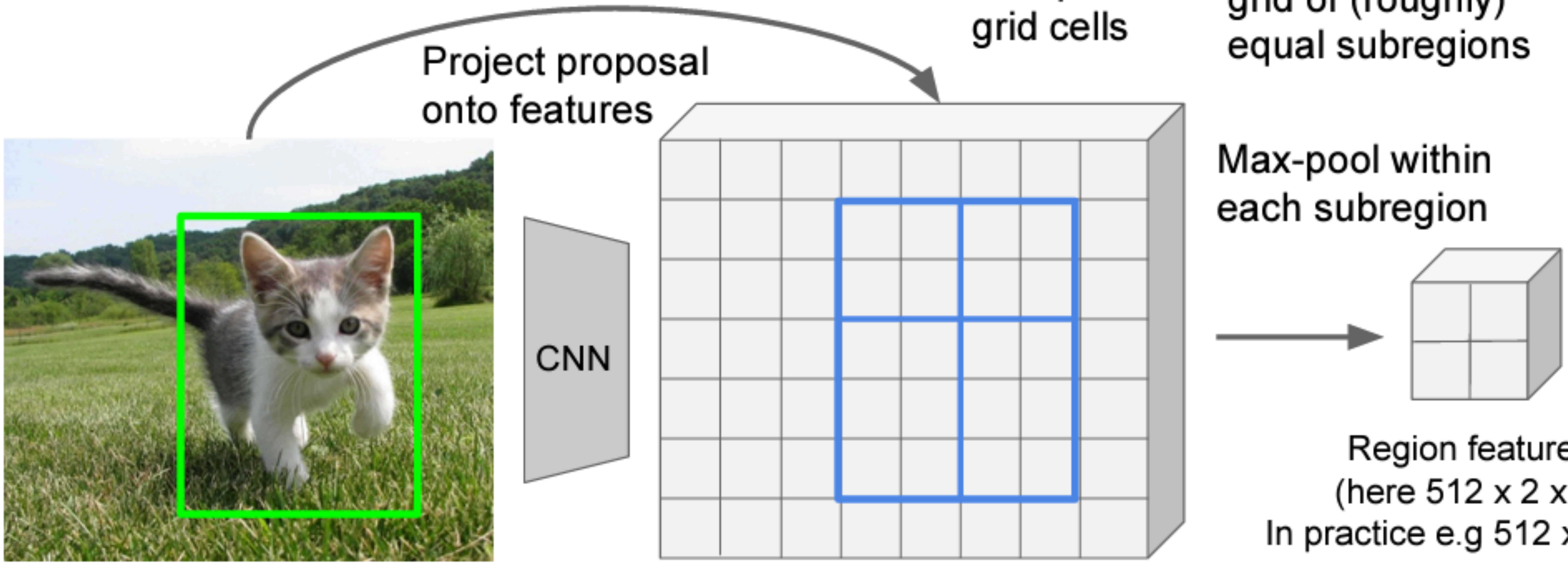
Girshick, "Fast R-CNN", ICCV 2015.

Cropping Features: RoI Pool

"Snap" to grid cells

Divide into 2x2 grid of (roughly) equal subregions

Max-pool within each subregion



Input Image
(e.g. 3 x 640 x 480)

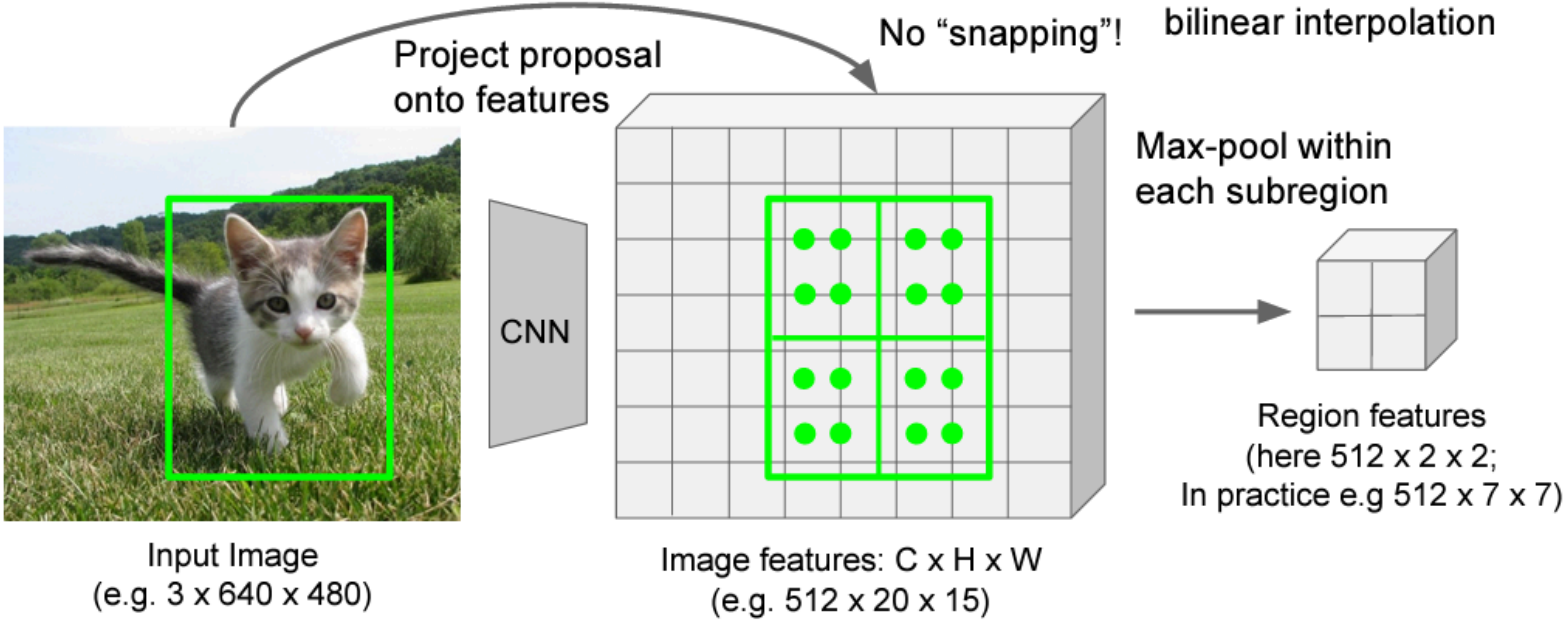
Image features: C x H x W
(e.g. 512 x 20 x 15)

Region features
(here 512 x 2 x 2;
In practice e.g 512 x 7 x 7)

Region features always the same size even if input regions have different sizes!

Girshick, "Fast R-CNN", ICCV 2015.

Cropping Features: RoI Align



He et al, "Mask R-CNN", ICCV 2017

Ablation: RoIPool vs RoIAlign

baseline: ResNet-50-Conv5 backbone, **stride=32**

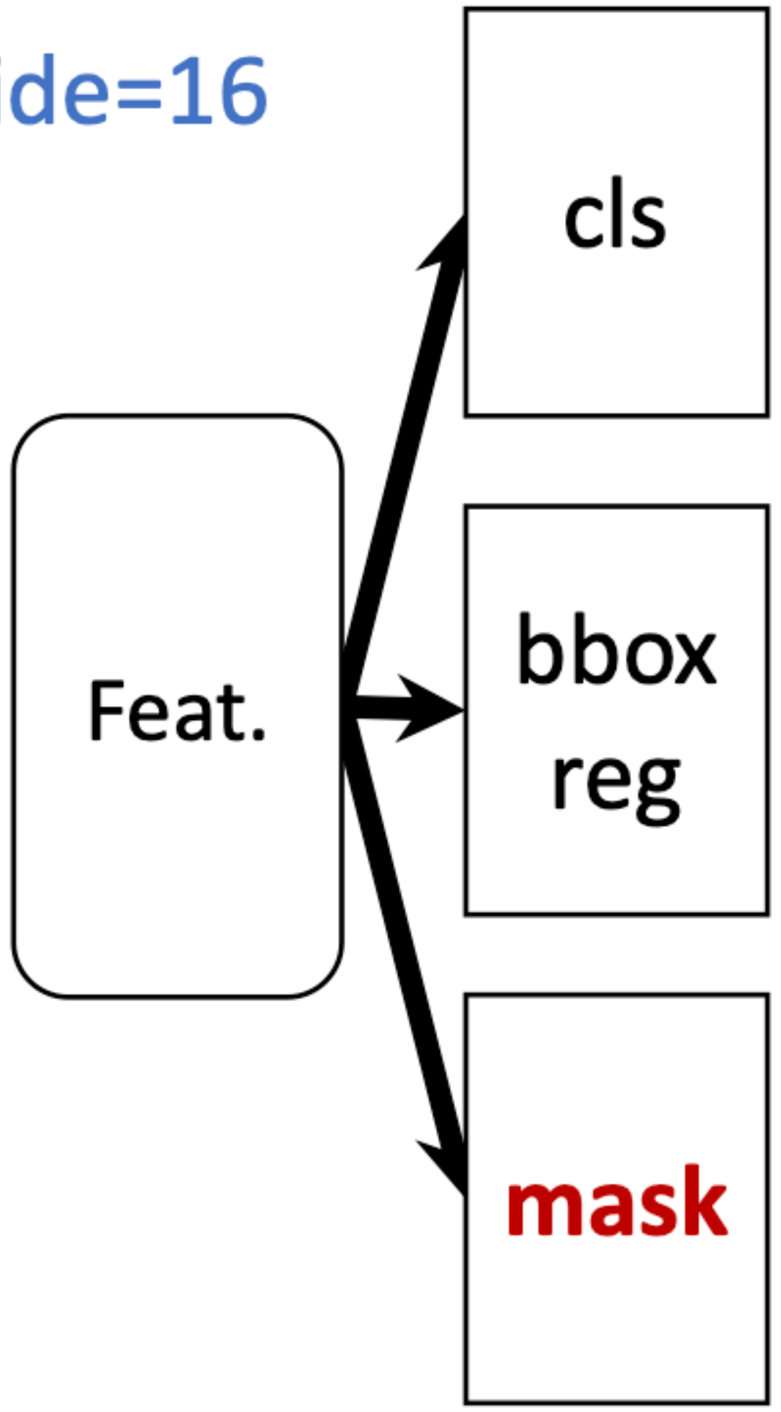
	mask AP			box AP		
	AP	AP_{50}	AP_{75}	AP^{bb}	AP^{bb}_{50}	AP^{bb}_{75}
RoIPool	23.6	46.5	21.6	28.2	52.7	26.9
RoIAlign	**30.9**	**51.8**	**32.1**	**34.0**	**55.3**	**36.4**
	+7.3	*+ 5.3*	$\boxed{+10.5}$	*+5.8*	*+2.6*	*+9.5*

- huge gain at high IoU, in case of big stride (32)

Ablation: Multinomial vs Binary Segmentation

baseline: ResNet-50-Conv4 backbone, stride=16

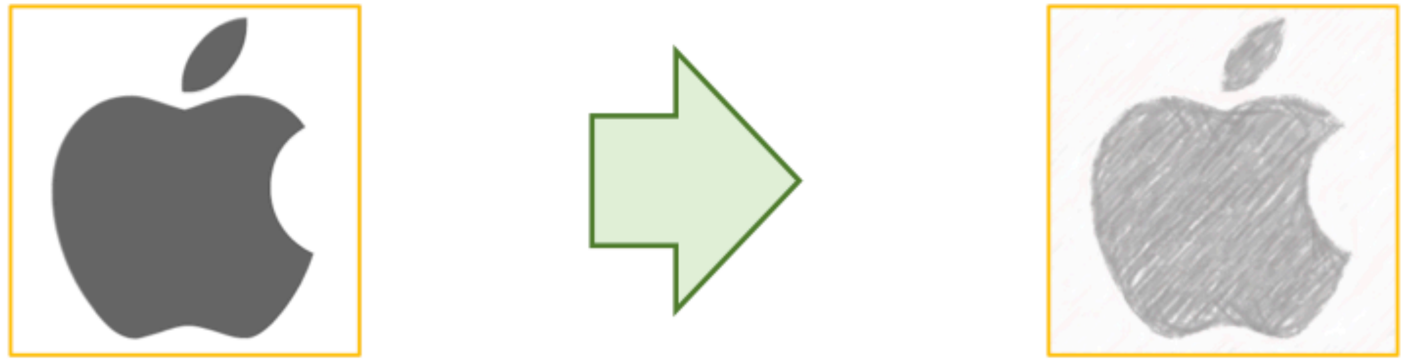
	AP	AP_{50}	AP_{75}
softmax	24.8	44.1	25.1
sigmoid	**30.3**	**51.2**	**31.5**
	+5.5	*+7.1*	*+6.4*



- cls head: did recognition

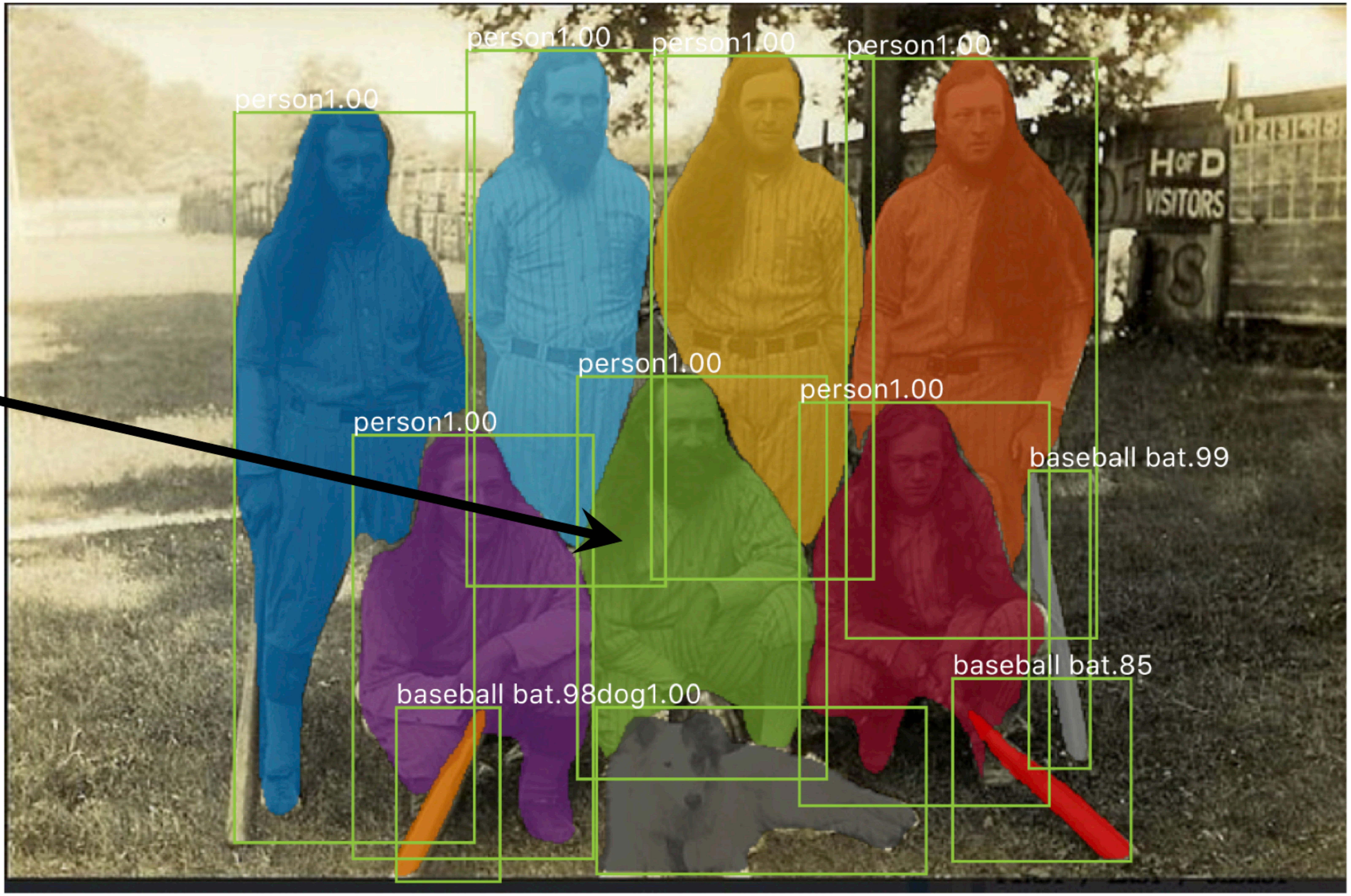
“apple”

- mask head: no need to recognize again



Mask R-CNN: Very Good Results!

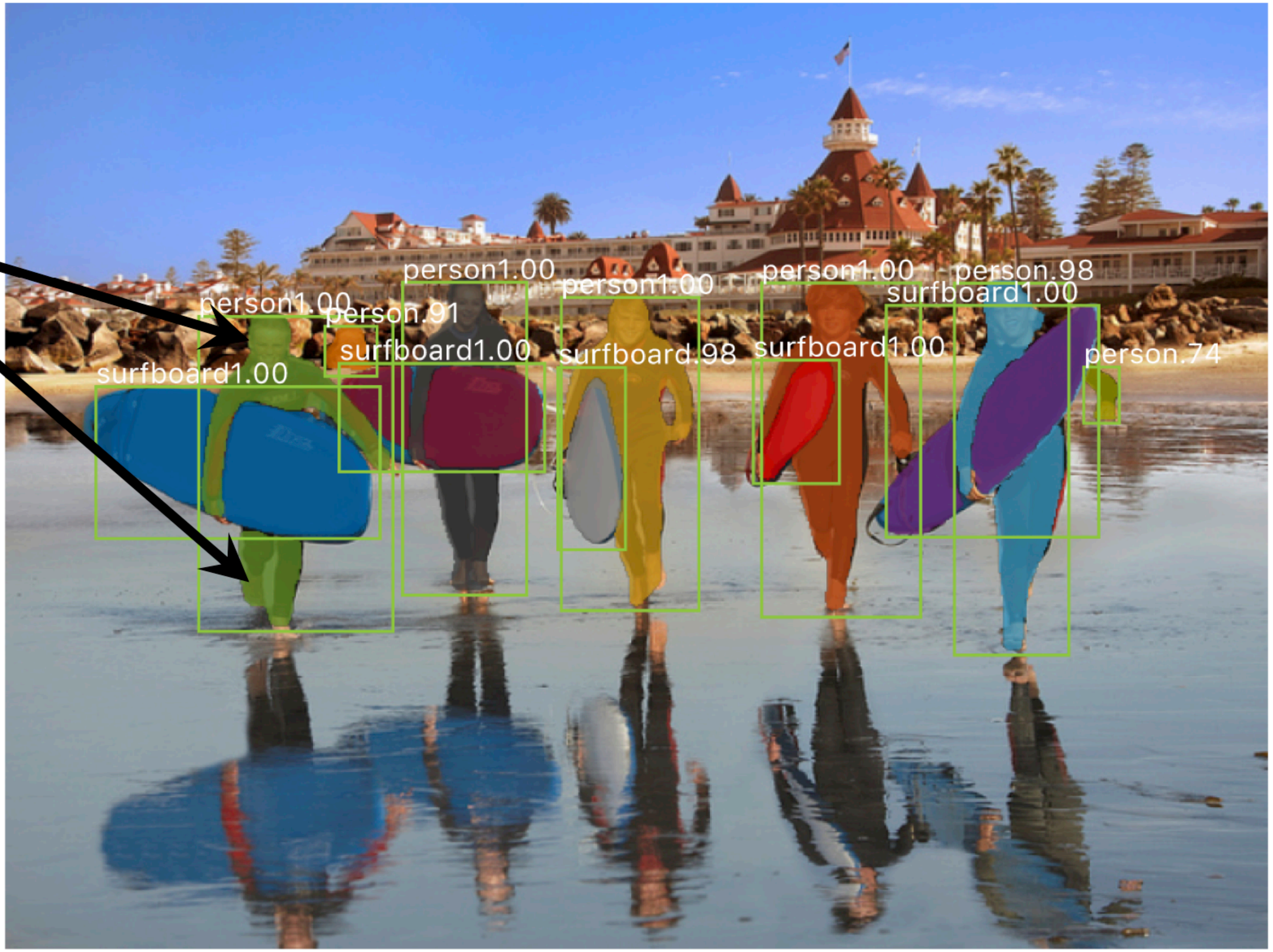
object surrounded by same-category objects



Mask R-CNN results on COCO

Mask R-CNN: Very Good Results!

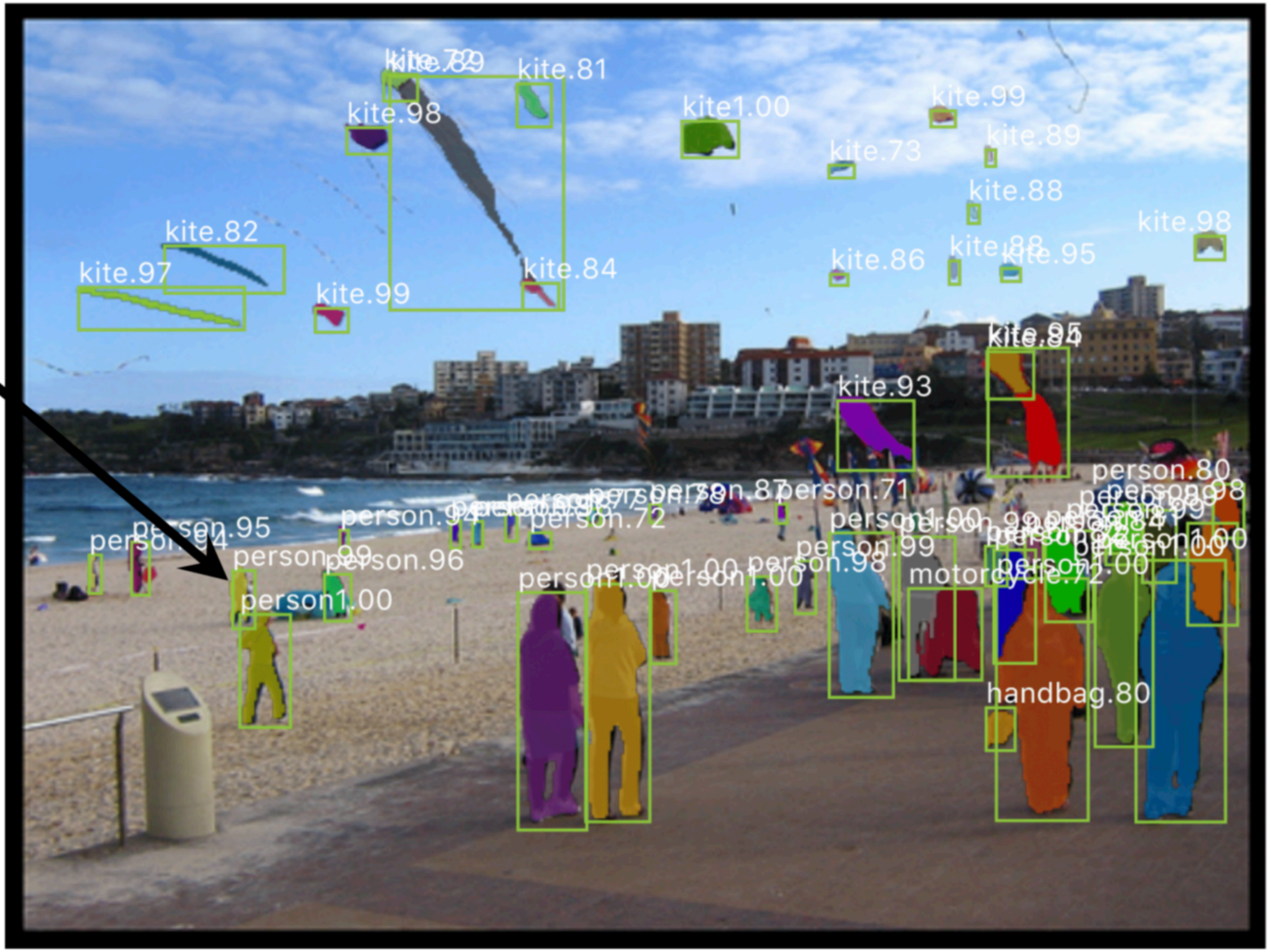
disconnected object



Mask R-CNN results on COCO

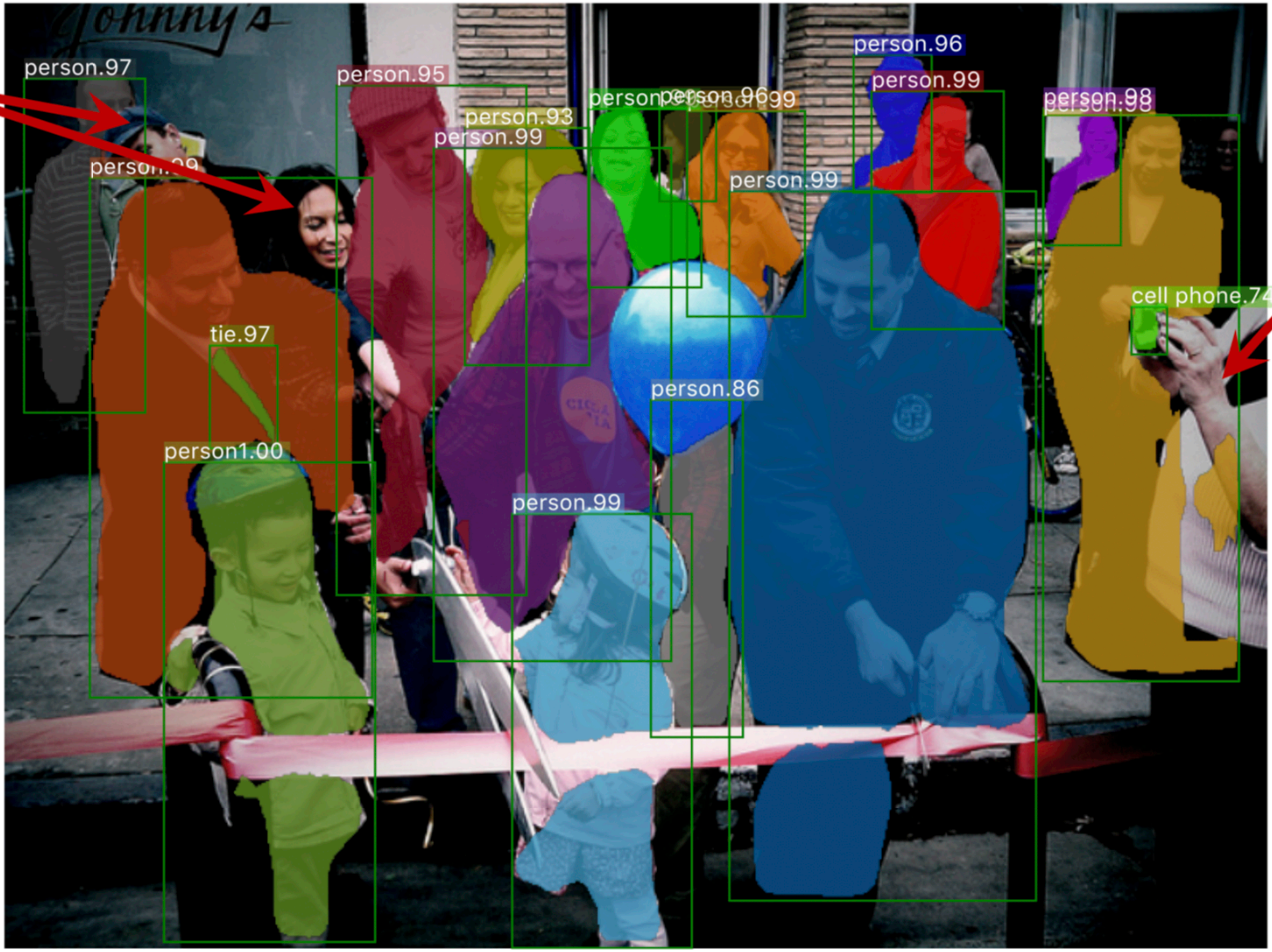
Mask R-CNN: Very Good Results!

small
objects



Mask R-CNN results on COCO

Mask R-CNN: Failure Case

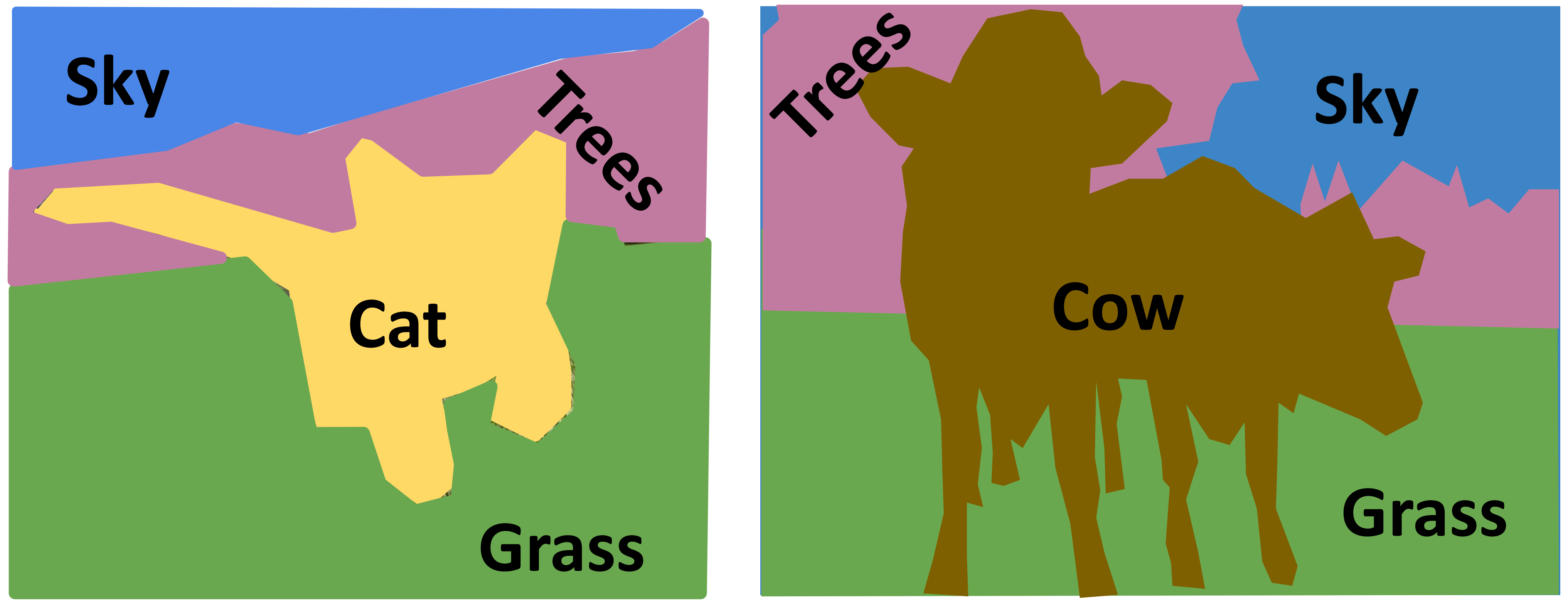


Mask R-CNN results on COCO

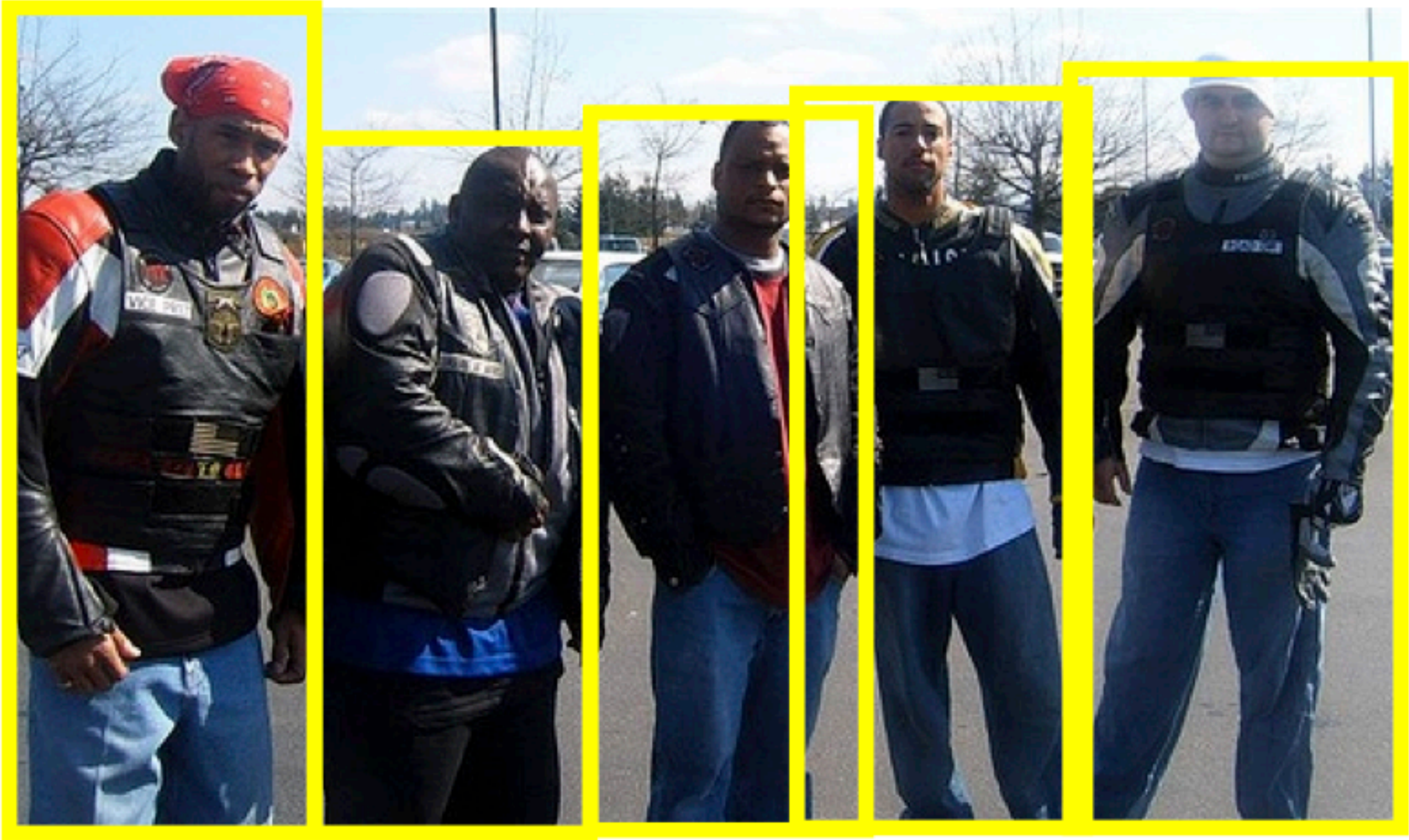
Semantic Segmentation

Label each pixel in the image with a category label

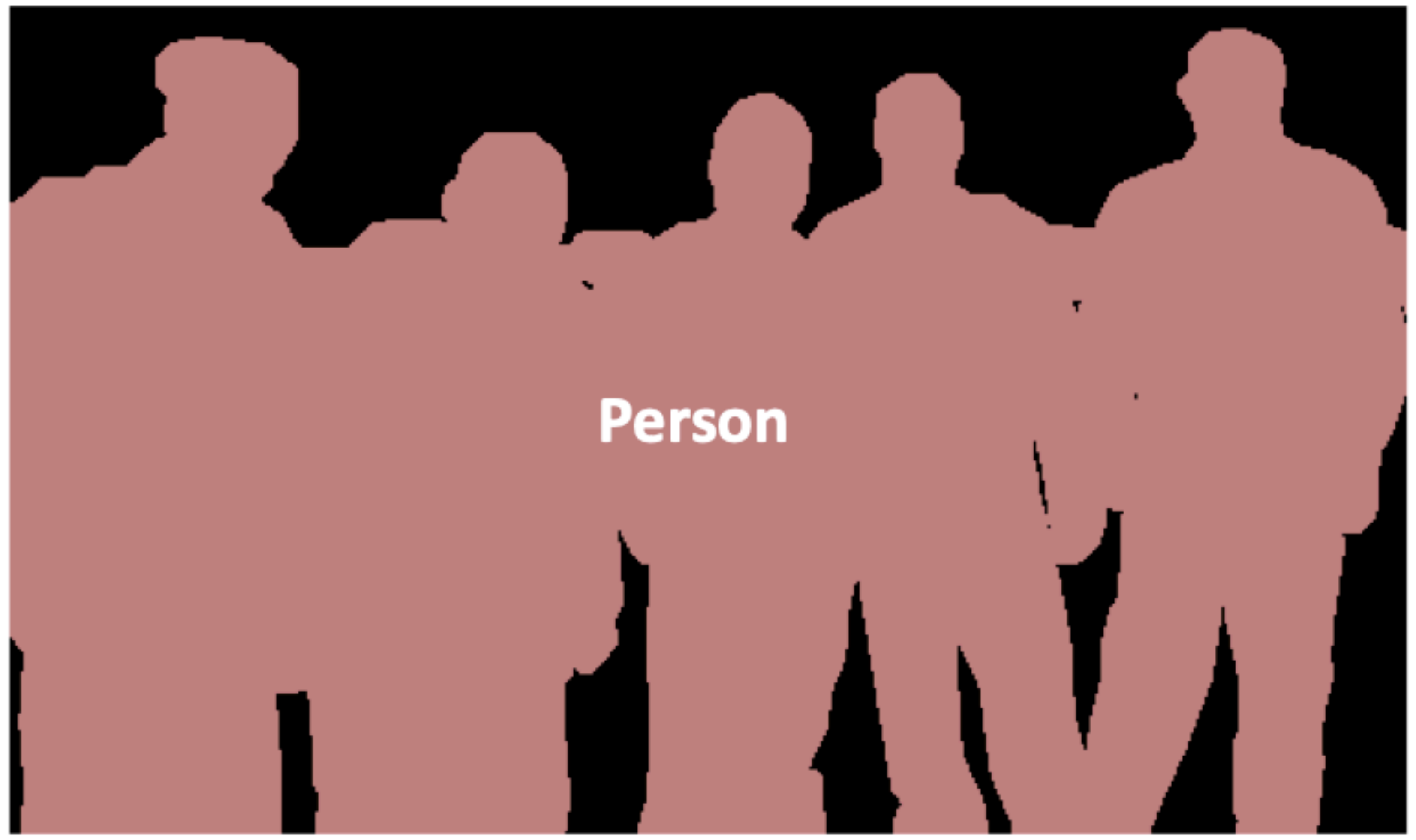
Don't differentiate instances, only care about pixels



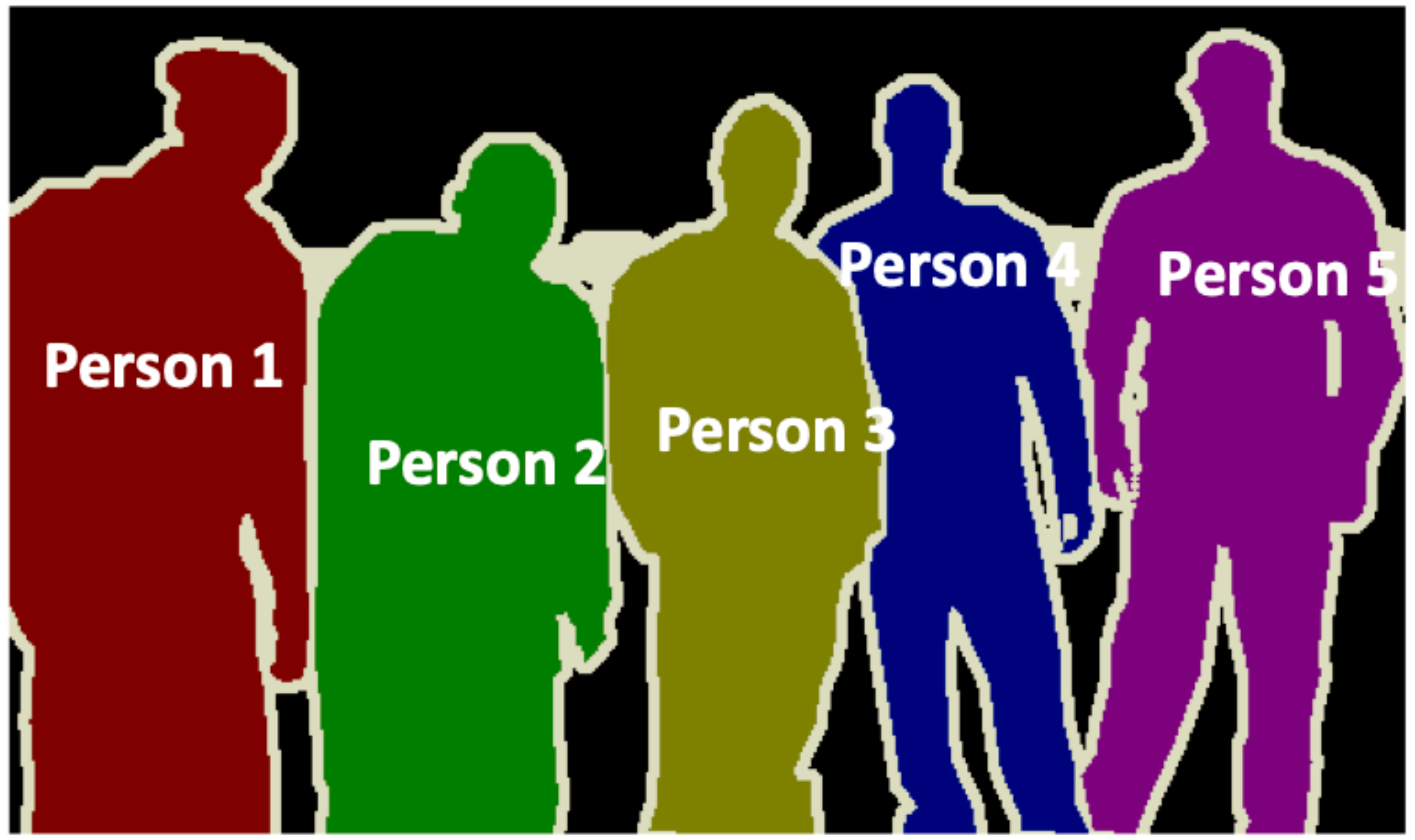
Semantic vs Instance Segmentation



Object Detection



Semantic Segmentation



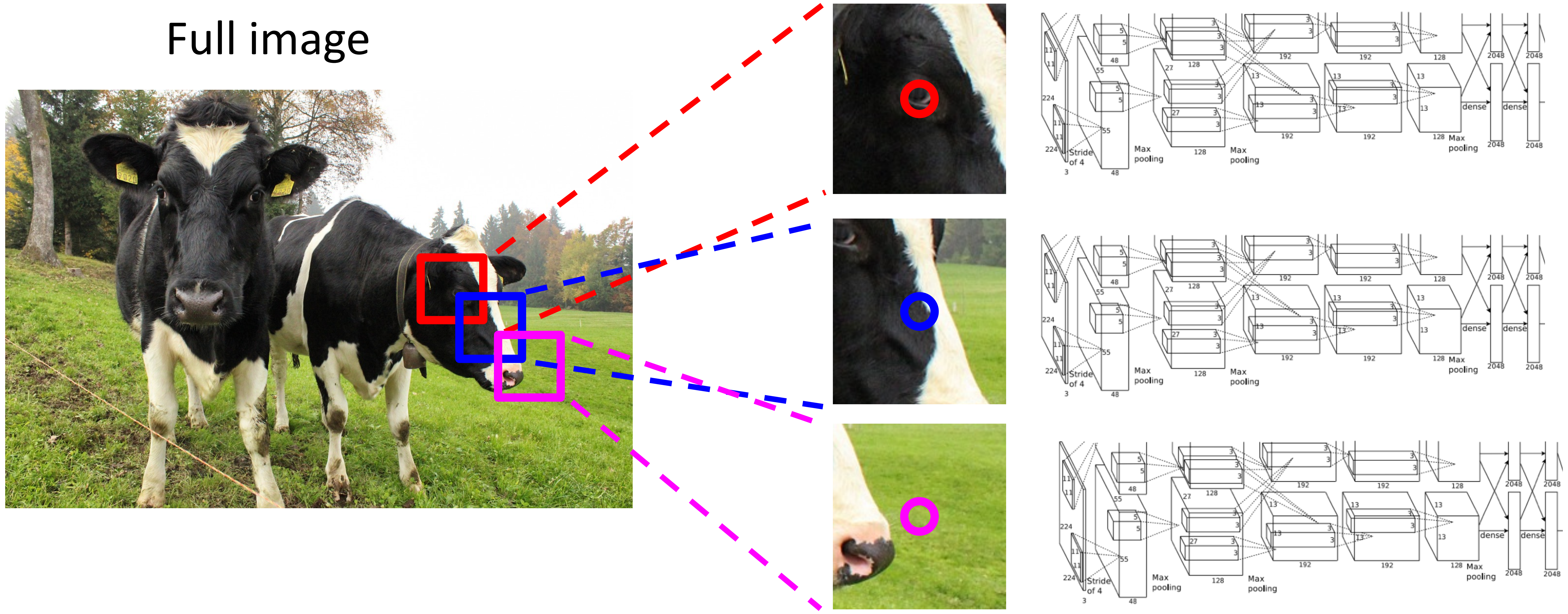
Instance Segmentation

Segmentation: Sliding Window

Extract patch

Classify center pixel with CNN

Full image



Cow

Cow

Grass

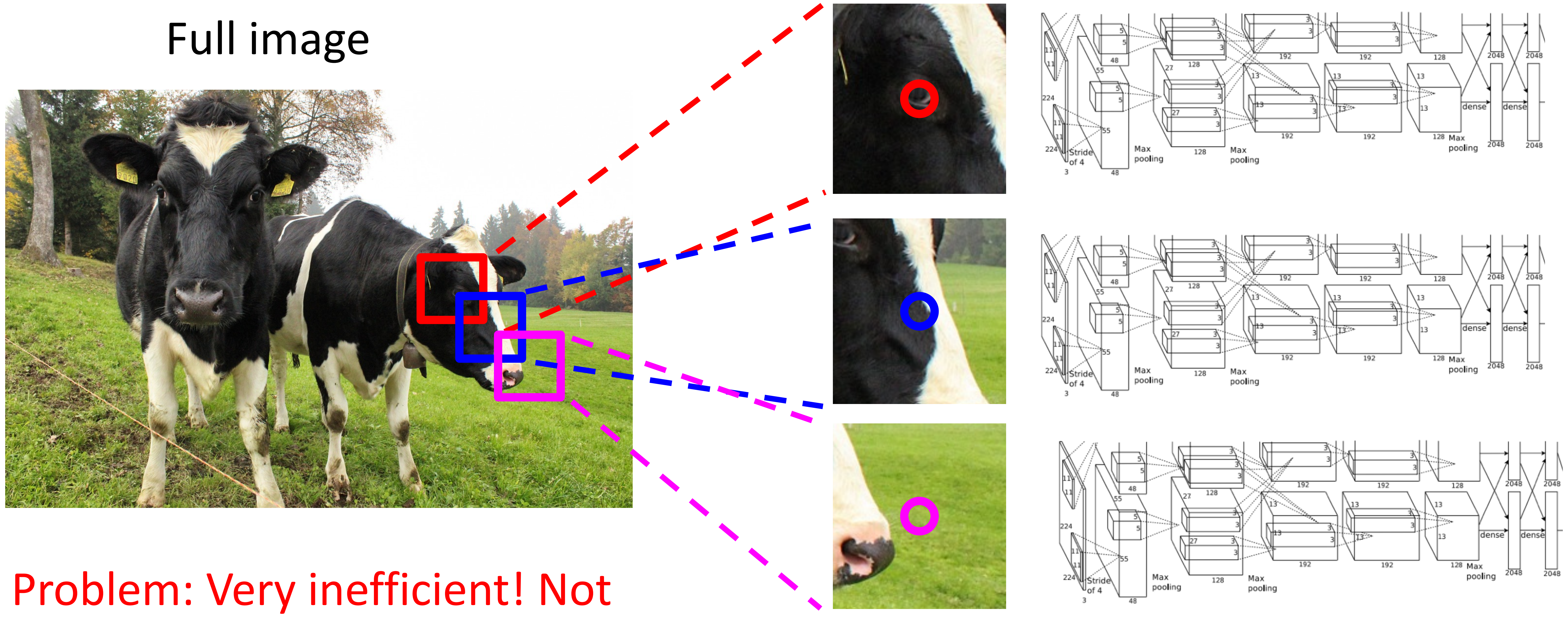
Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013
Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014

Segmentation: Sliding Window

Extract patch

Classify center pixel with CNN

Full image



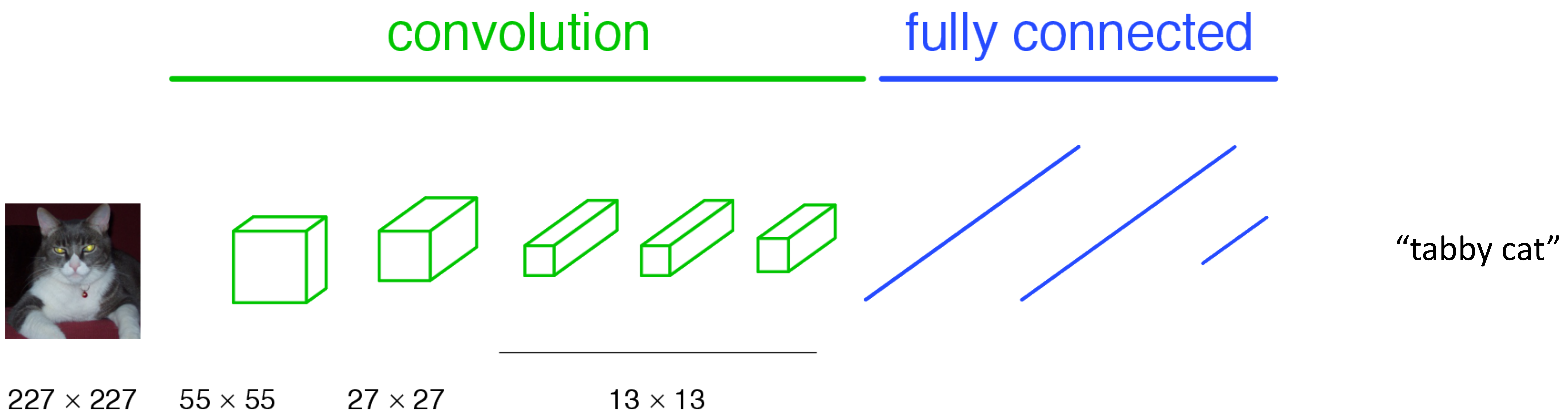
Cow

Cow

Grass

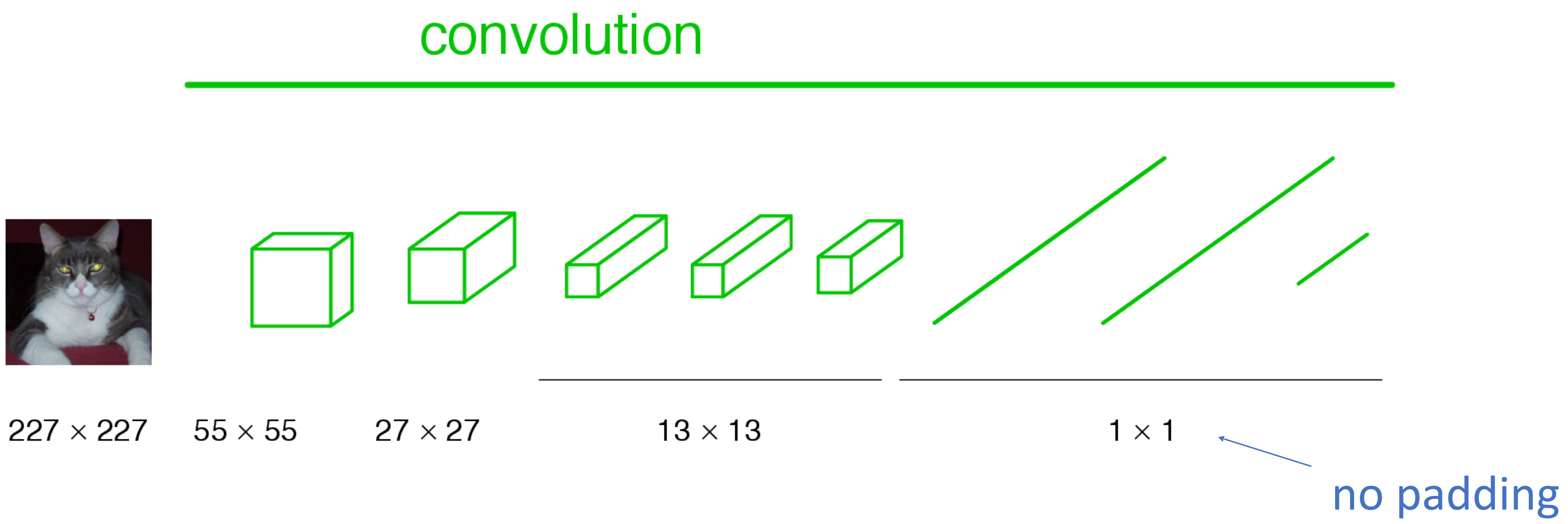
Problem: Very inefficient! Not reusing shared features between overlapping patches

A Classification Network



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

Becoming Fully Convolutional

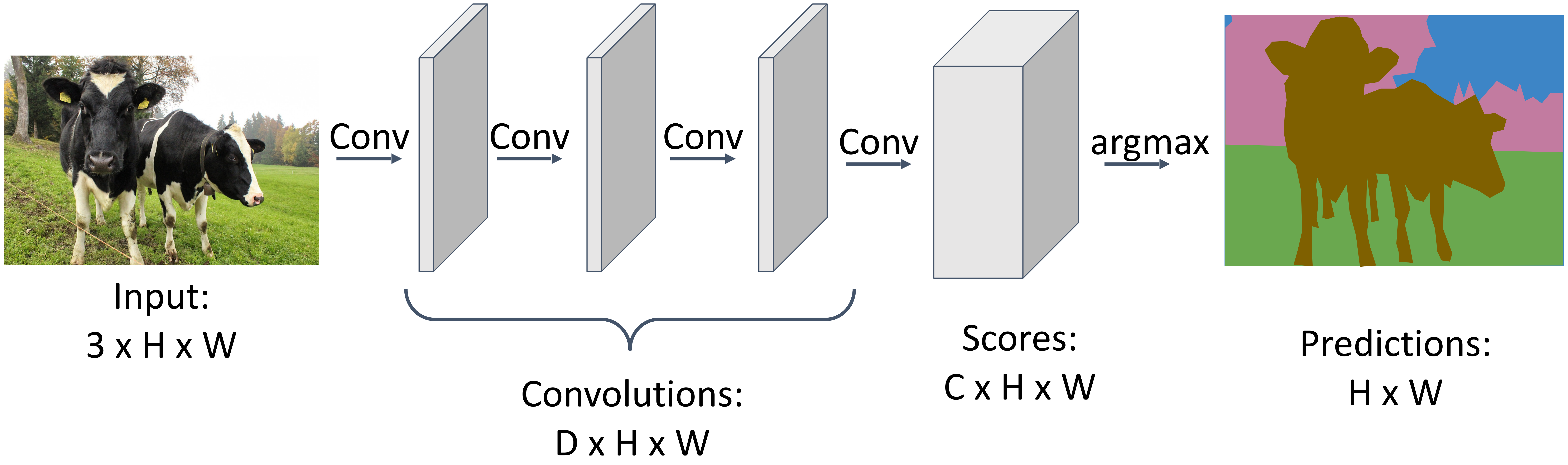


A fully-connected layer is equivalent to a convolution layer.

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

Fully Convolutional Network

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

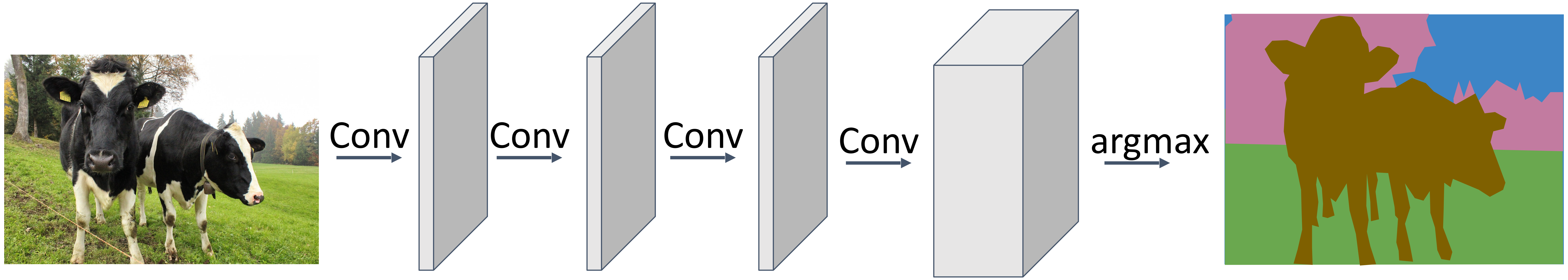


Loss function: Per-Pixel cross-entropy

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

Fully Convolutional Network

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



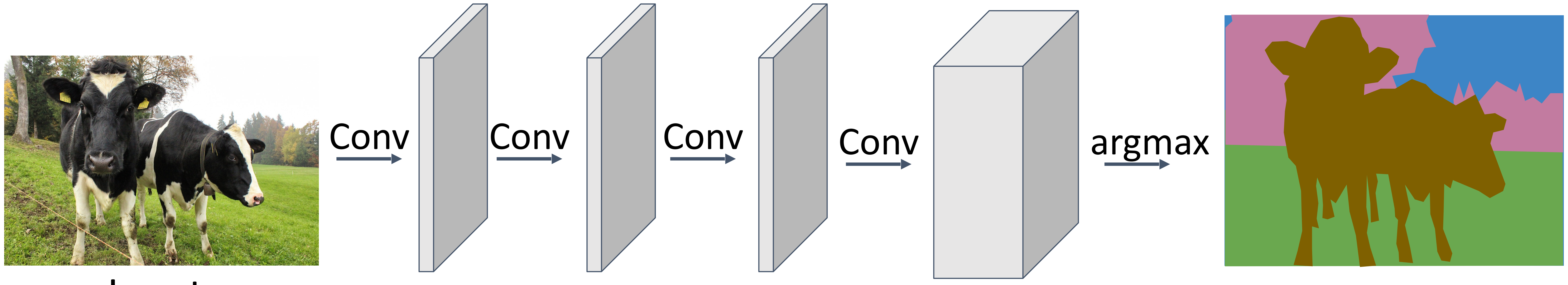
Input:
3 x H x W

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

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

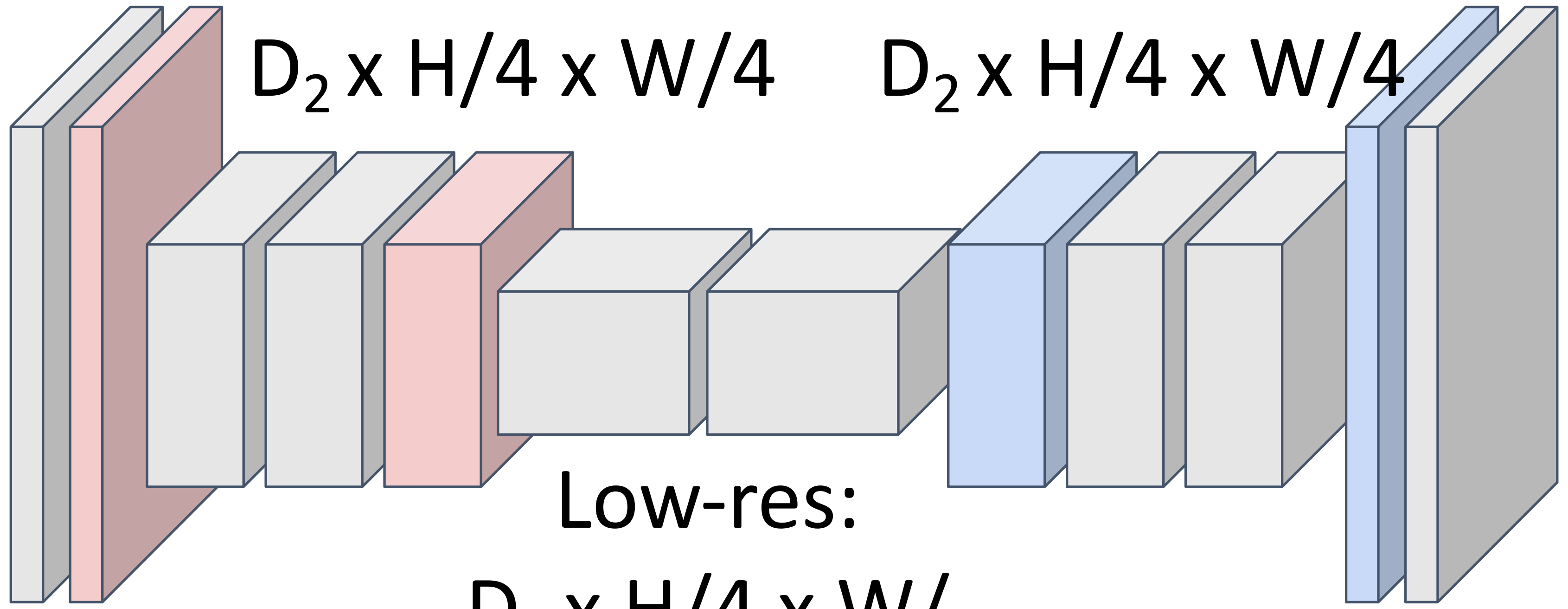
Problem #1: Effective receptive field size is linear in number of conv layers: With L 3x3 conv layers, receptive field is 1+2L

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

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

Predictions:
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

In-Network Upsampling: “Unpooling”

Bed of Nails

Input

C x 2 x 2

1	2
3	4

→

Output

C x 4 x 4

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

Nearest Neighbor

Input

C x 2 x 2

1	2
3	4

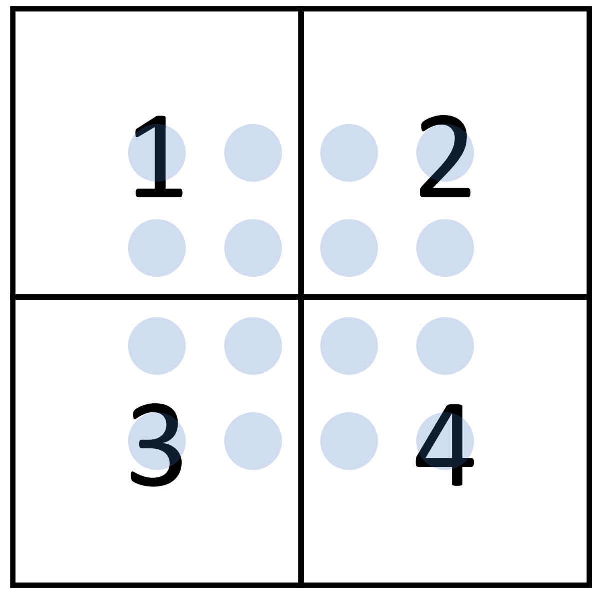
→

Output

C x 4 x 4

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

Upsampling: Bilinear Interpolation



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

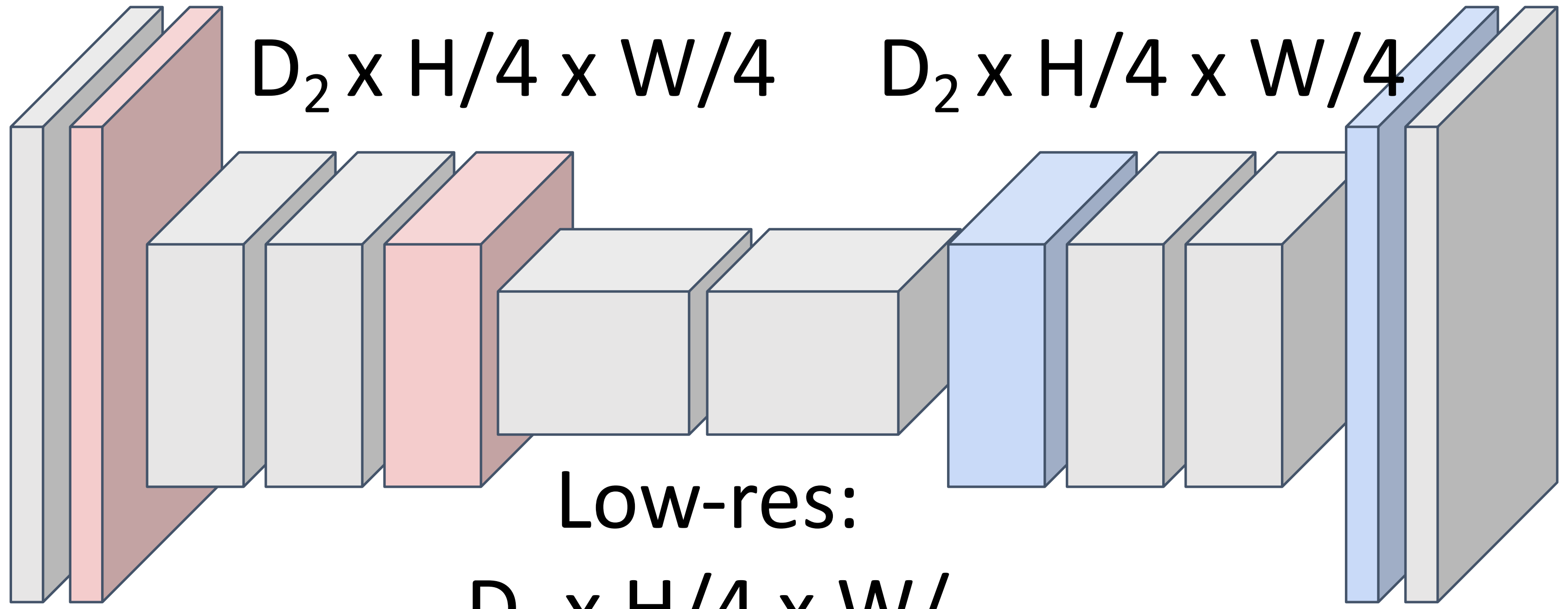
Fully Convolutional Network

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

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

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

Input: 3 x H x W



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

Predictions: H x W

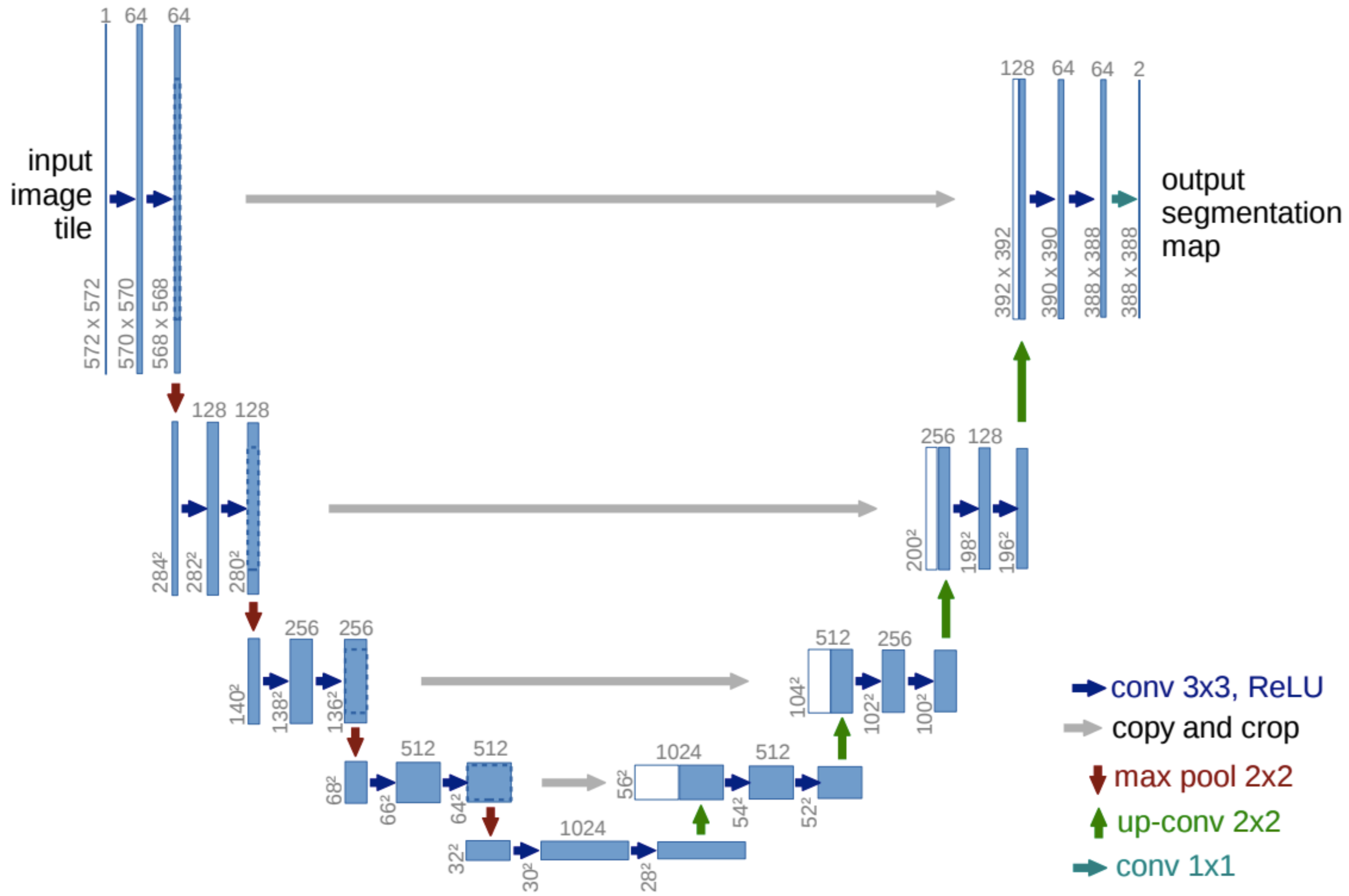
Downsampling: Pooling, strided convolution

Upsampling: ???

U-Net

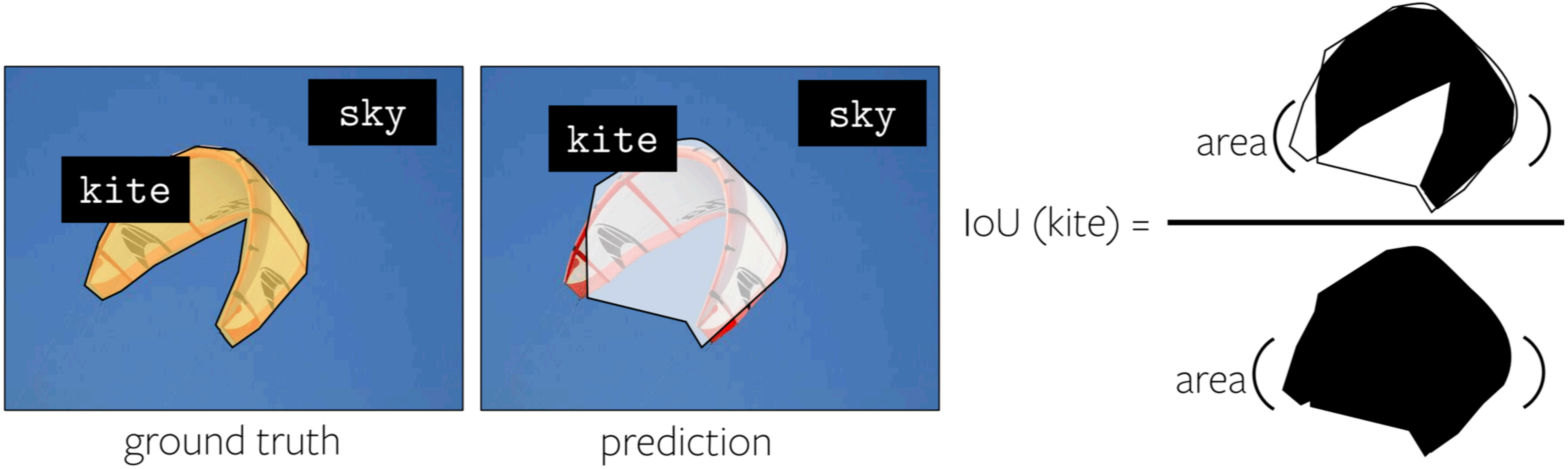
O. Ronneberger, P. Fischer, T. Brox, [U-Net: Convolutional Networks for Biomedical Image Segmentation](), MICCAI 2015

- Like FCN, fuse upsampled higher-level feature maps with higher-res, lower-level feature maps
- Unlike FCN, fuse by concatenation, predict at the end



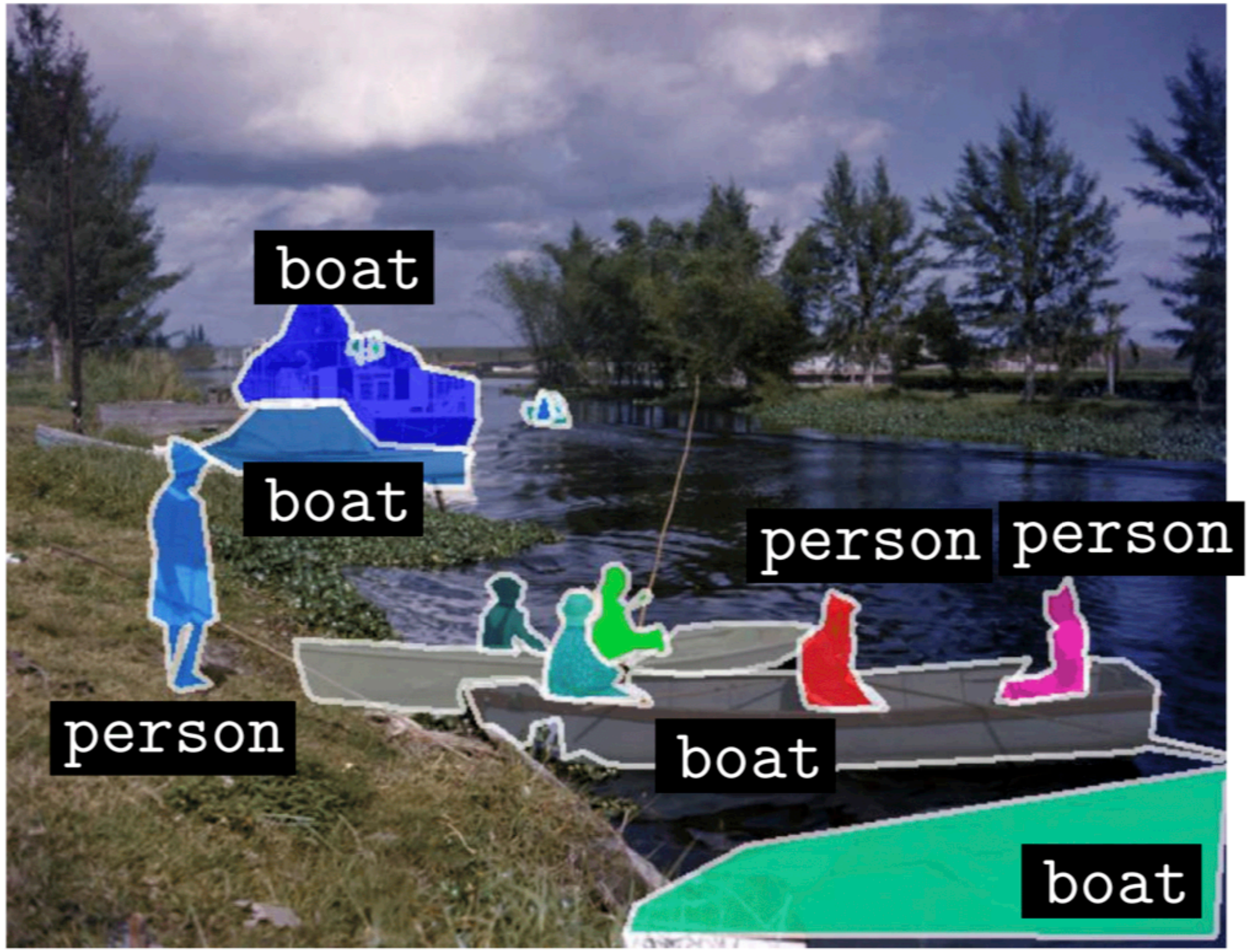

road
sidewalk
building
wall
fence
pole
traffic light
traffic sign
vegetation
terrain
sky
person
rider
car
truck
bus
train
motorcycle
bicycle

Evaluation of Semantic Segmentation

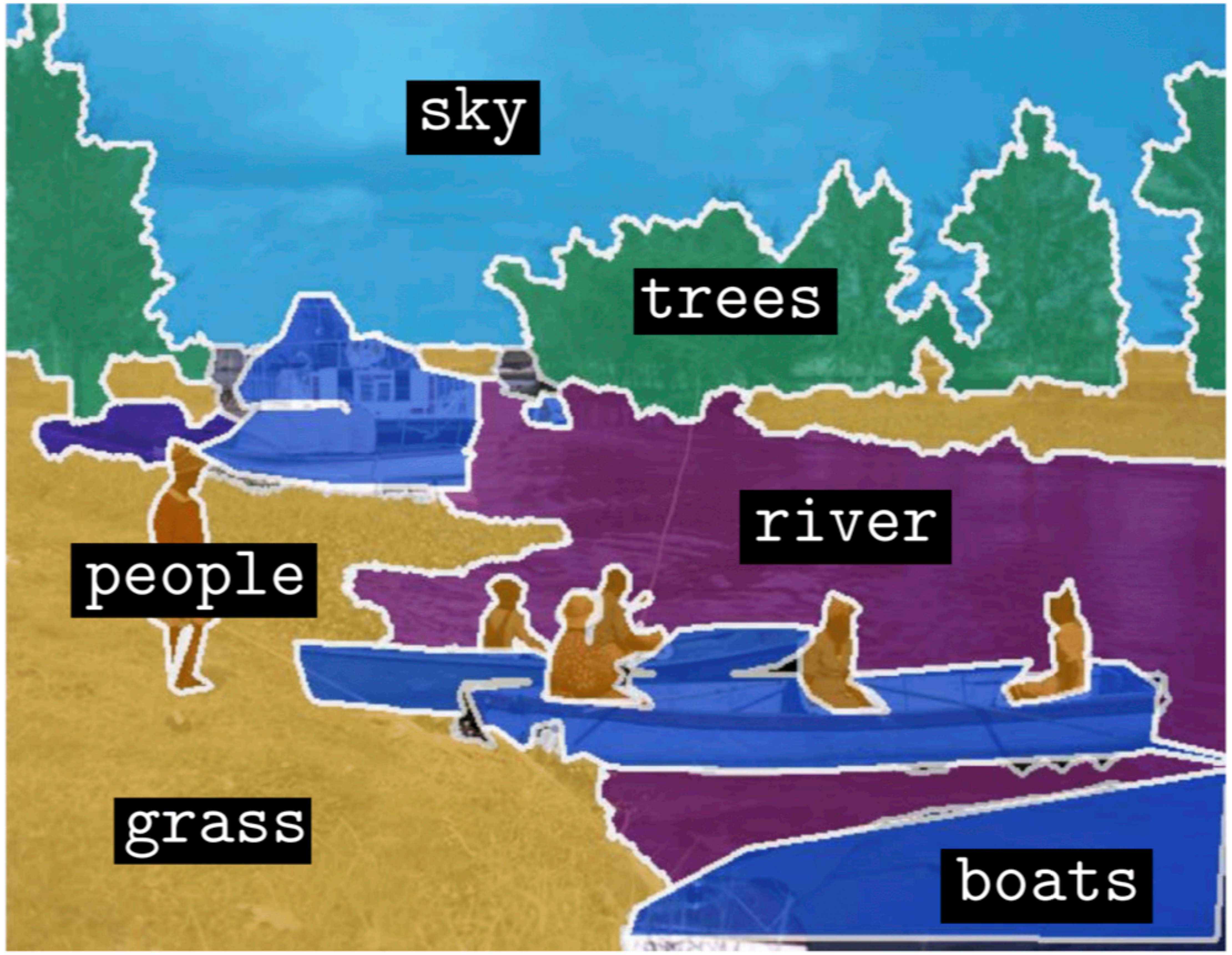


mIoU (mean IoU) per class

Instance and Semantic Segmentation



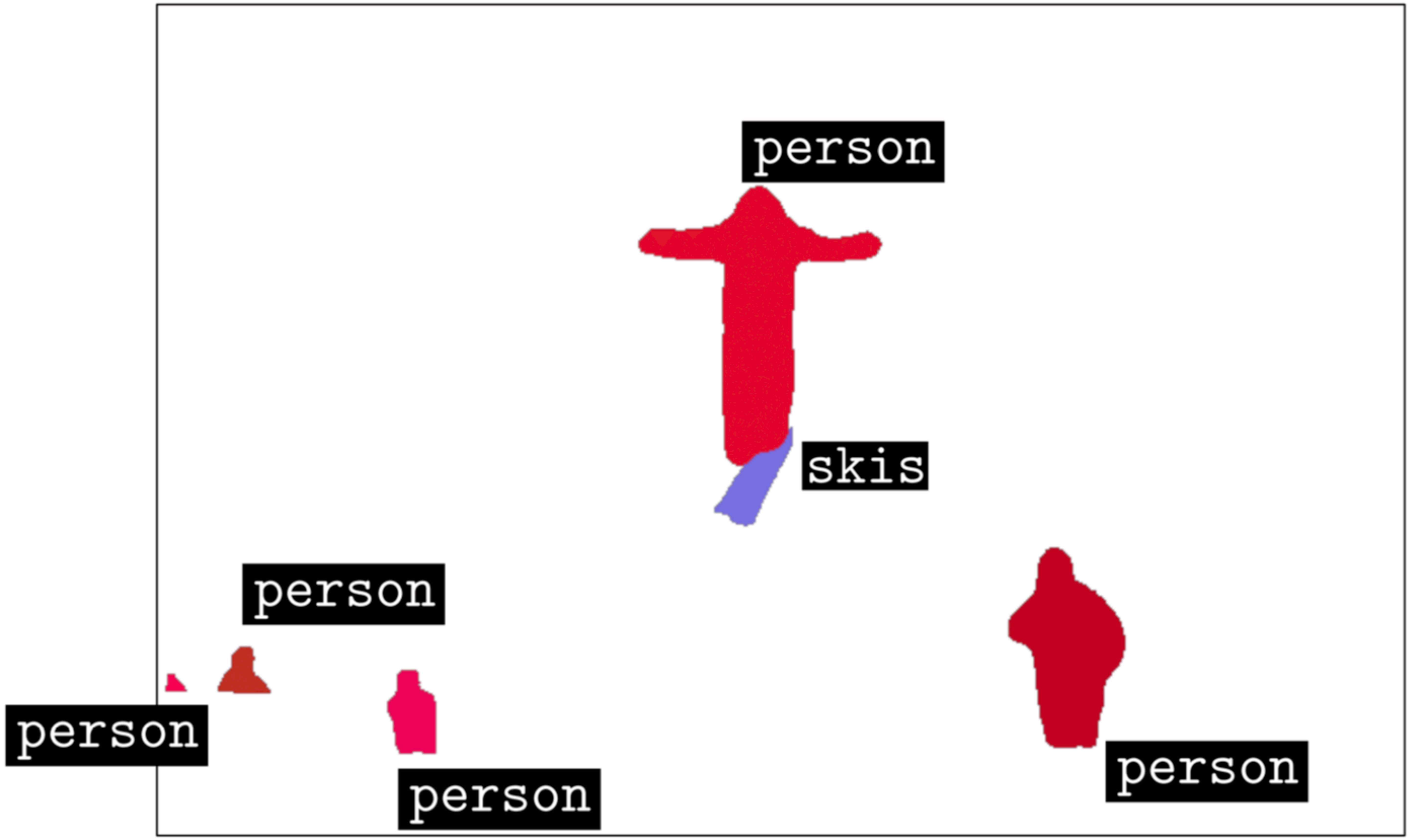
instance segmentation



semantic segmentation

real-world application likely requires both modalities

What do instance segmentation models see?

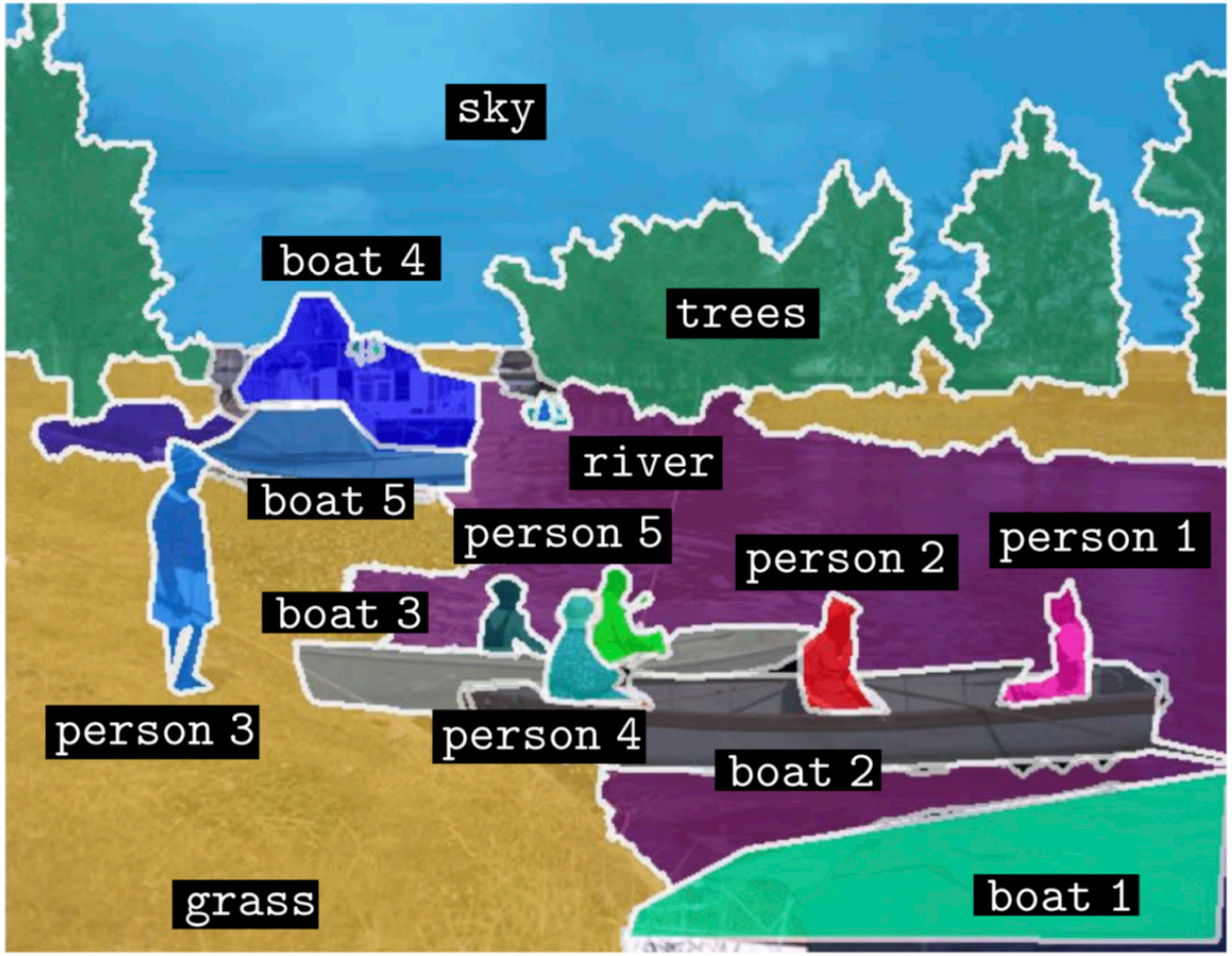


no understanding of the general scene layout

What do semantic segmentation models see?

Does not differentiate different instances

Panoptic Segmentation: Unified Segmentation



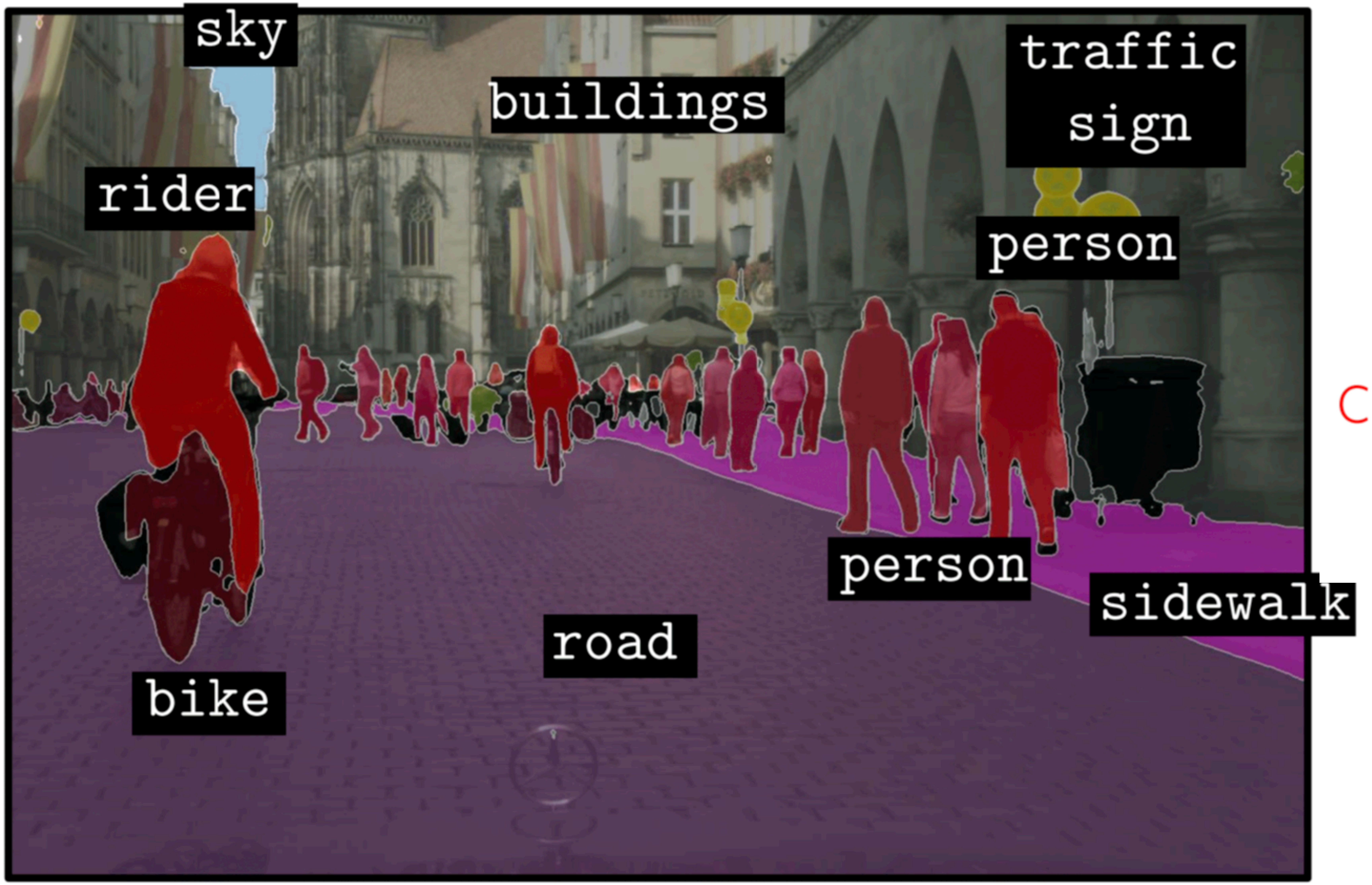
single task that combines semantic and instance segmentation

things: categories with instance-level annotation (person, boat)
stuff: categories without the notion of instances (sky, road)

Panoptic: see everything at once

Figure Credit: Alexander Kirillov

Panoptic Segmentation



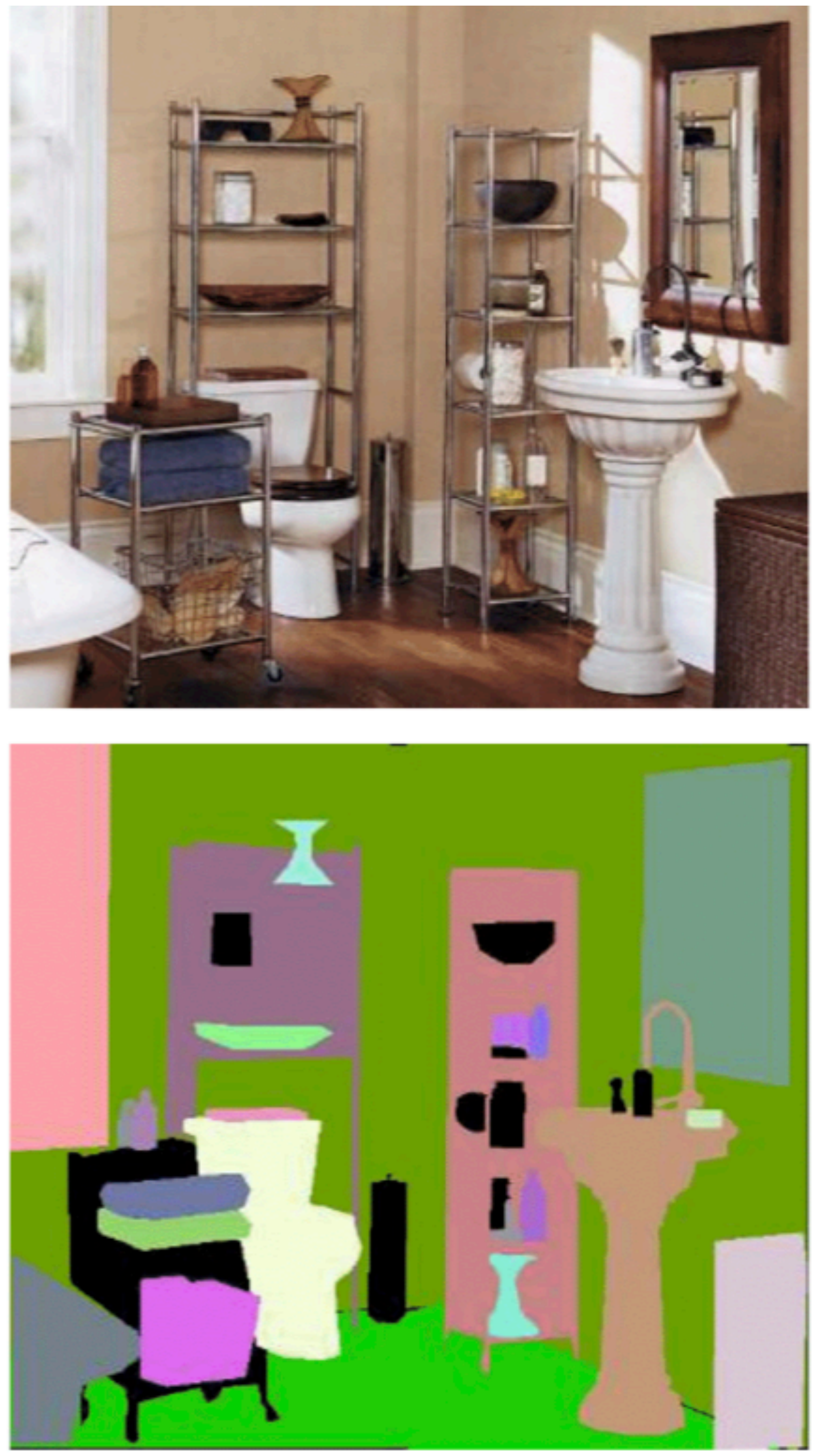
C

Available Panoptic Segmentation Datasets

CO (2014) + COCO-stuff (2017)
COCO-panoptic challenges:
ECCV`18, **ICCV`19**

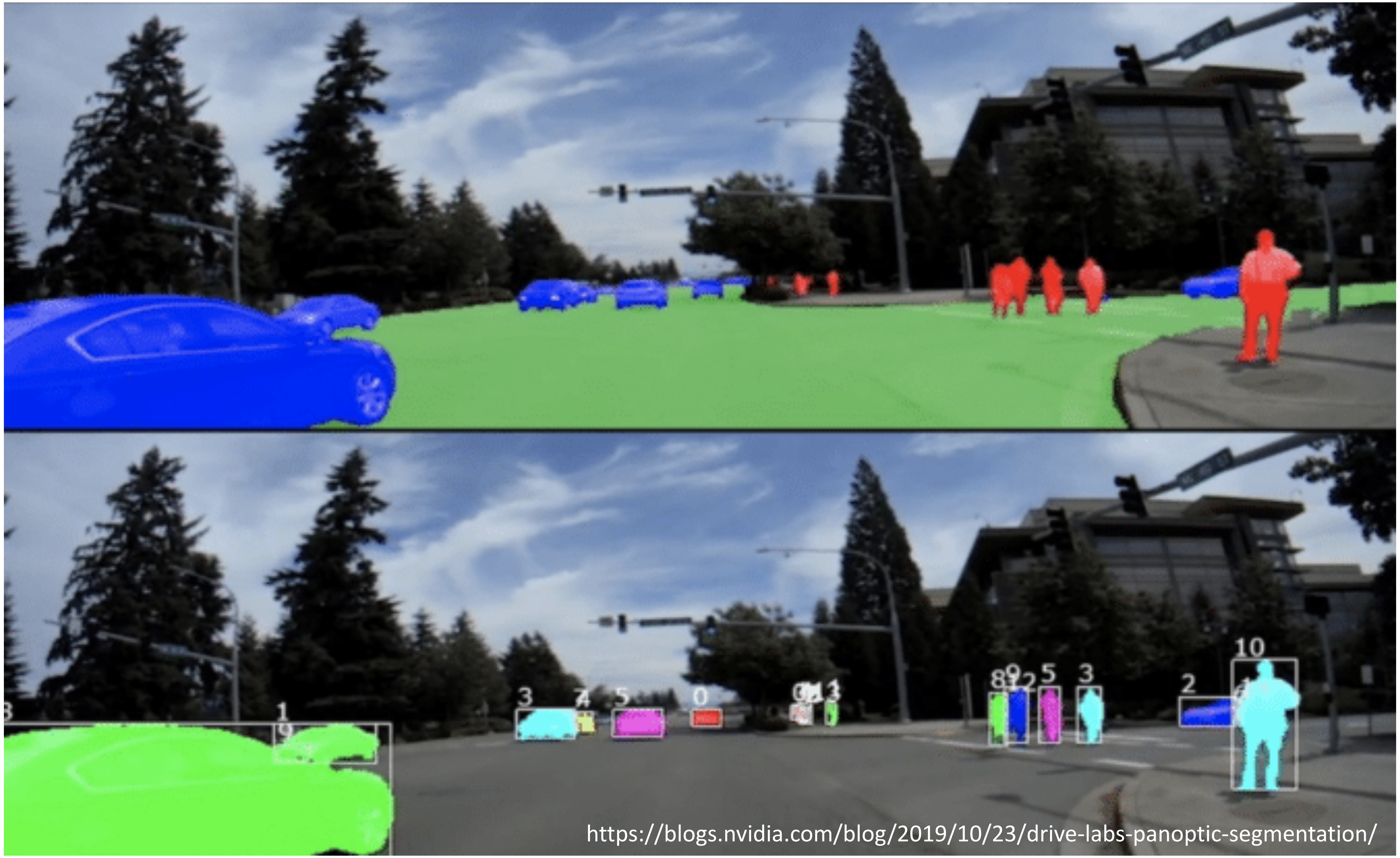
Mapillary Vistas (2017)
Vistas-panoptic challenges:
ECCV`18, **ICCV`19**

Cityscapes (2015)
panoptic test set
leaderboard (2019)

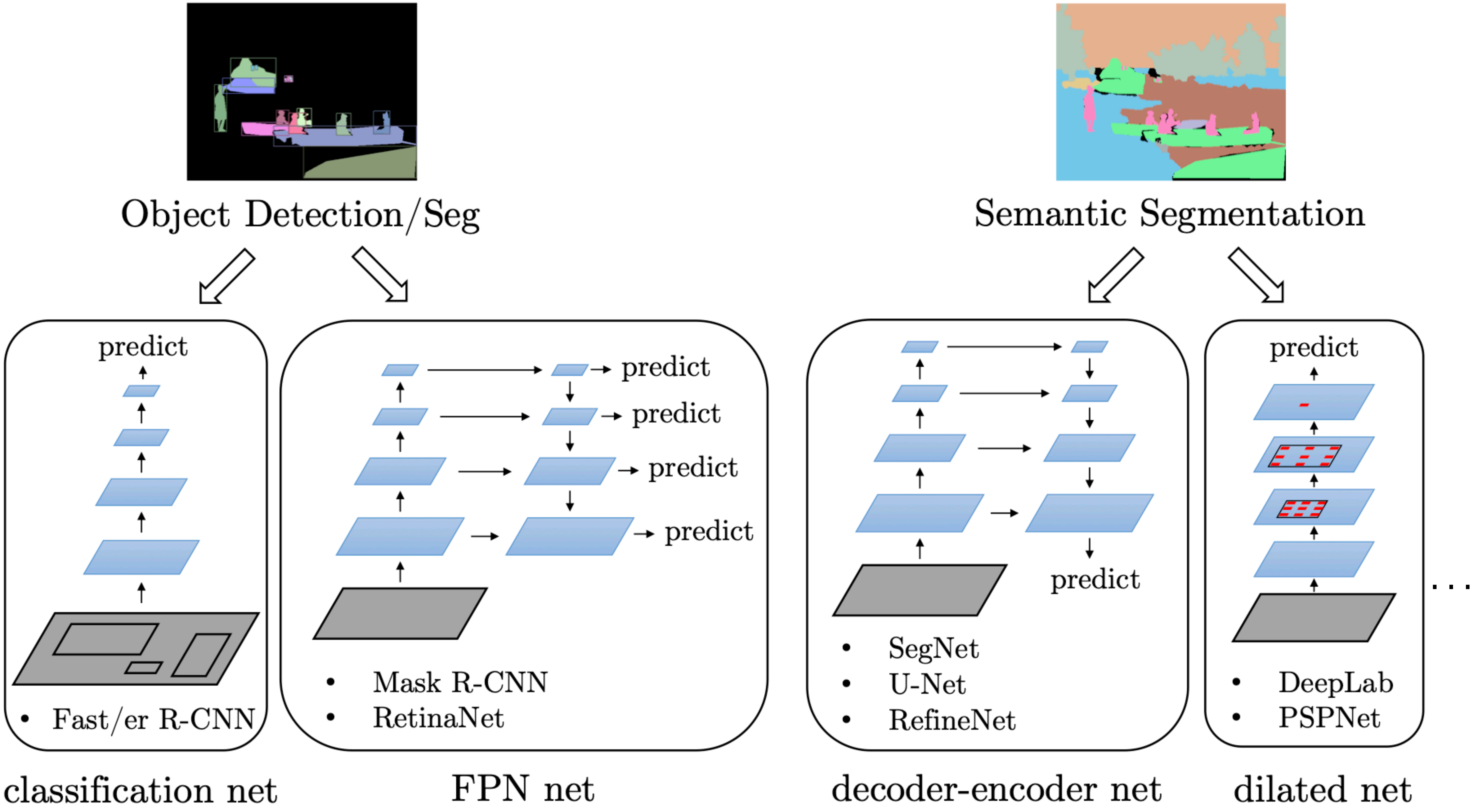


ADE20k (2016)
>22k images, 150 categories

Panoptic Segmentation for Autonomous Driving



Deep Networks for Segmentation Tasks



Panoptic FPN

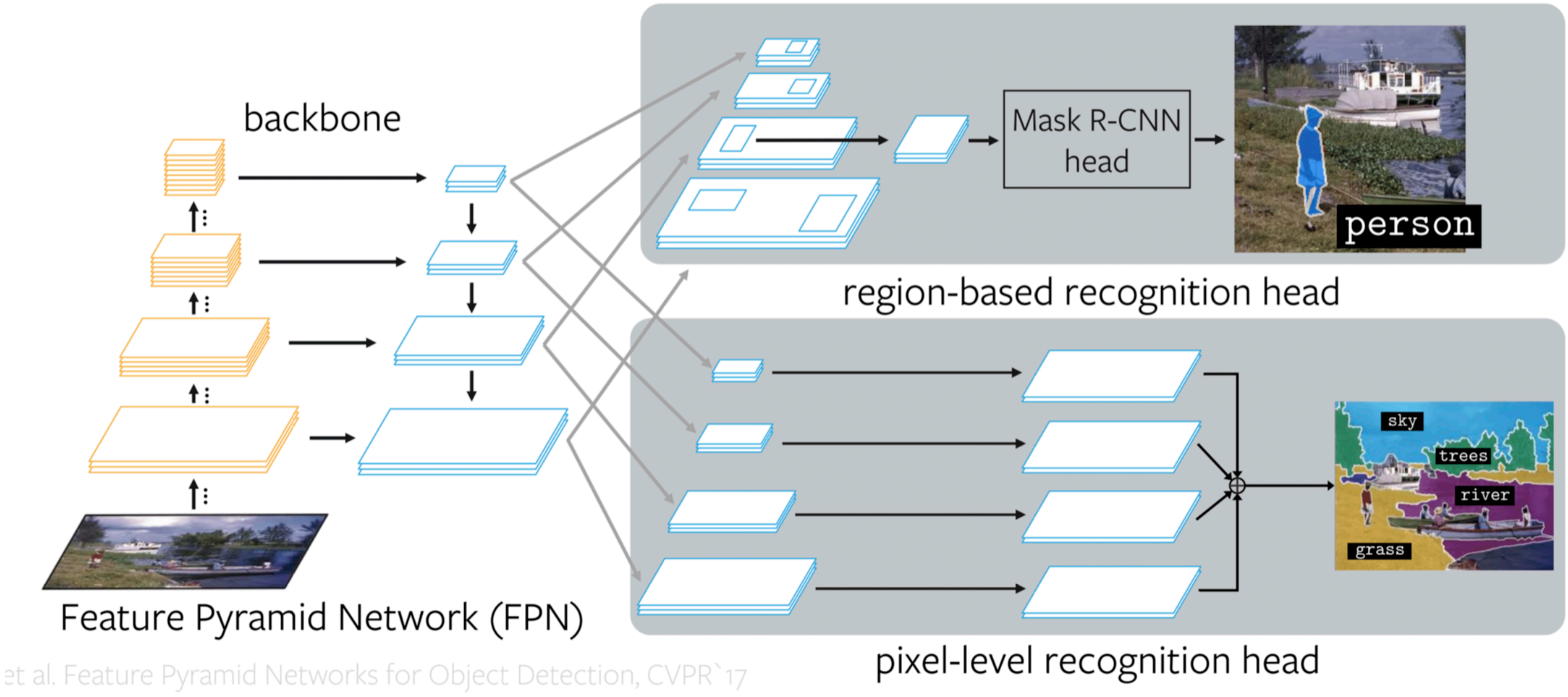
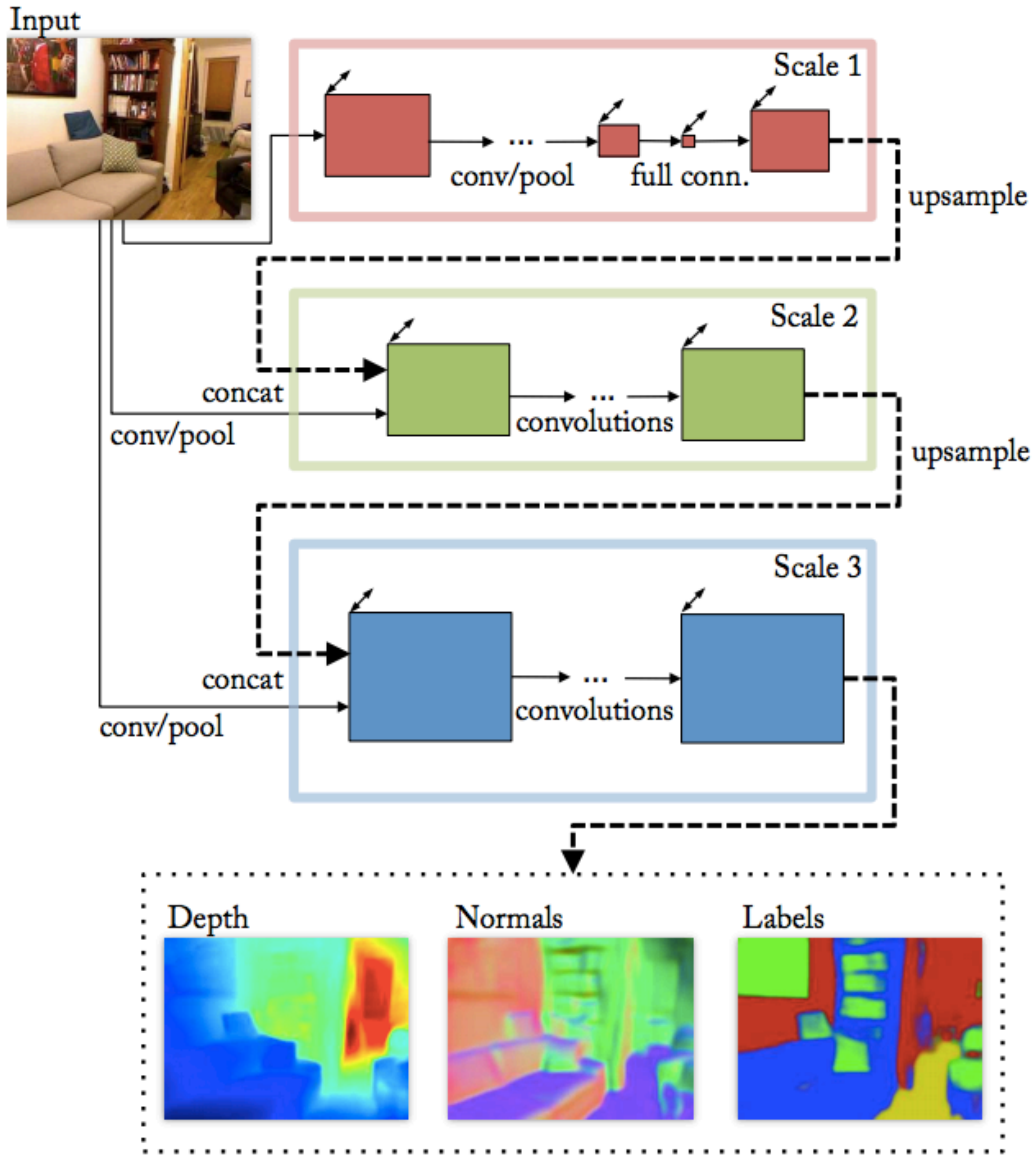
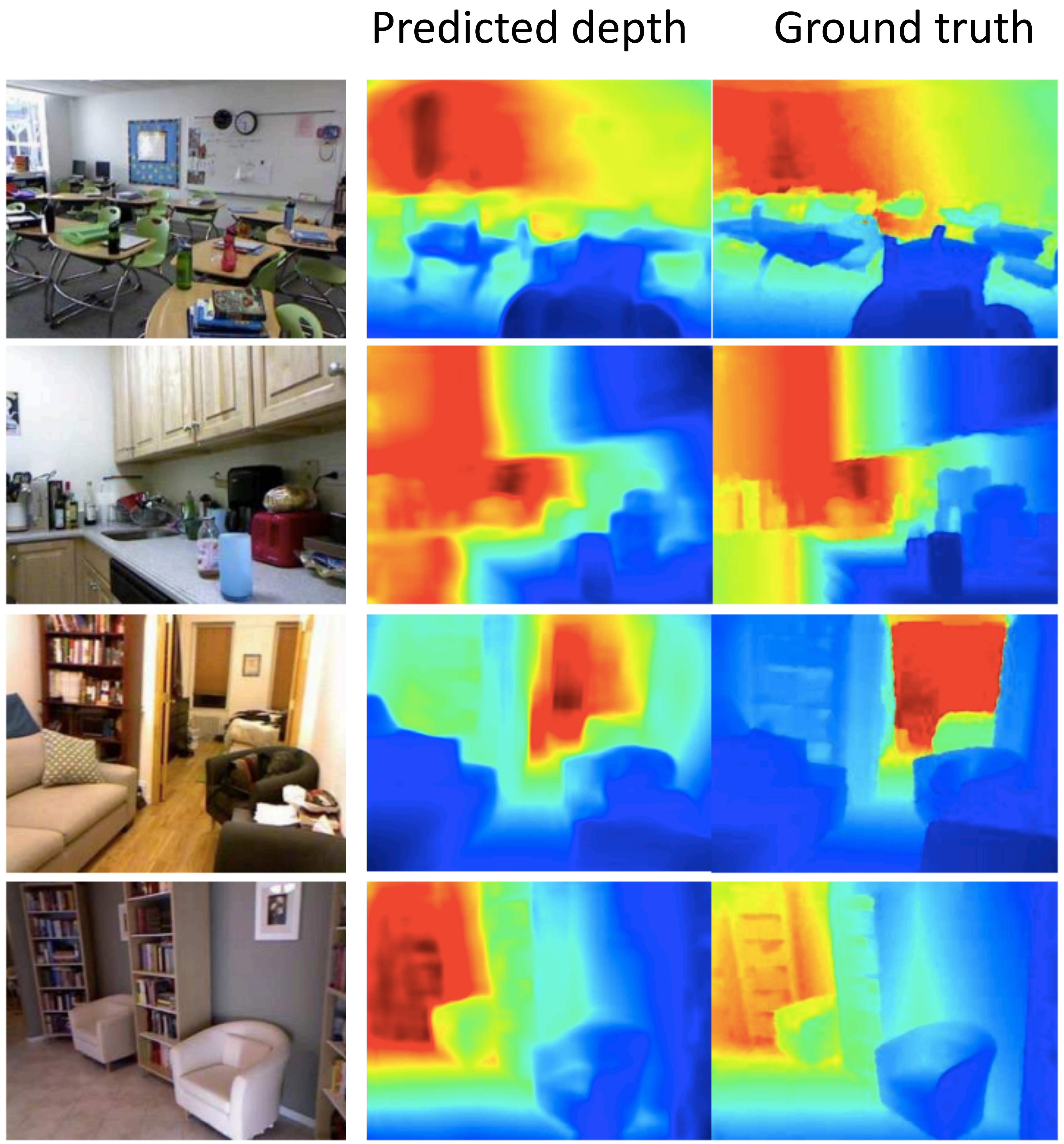


Figure Credit: Alexander Kirillov

Dense Prediction: Depth and normal estimation

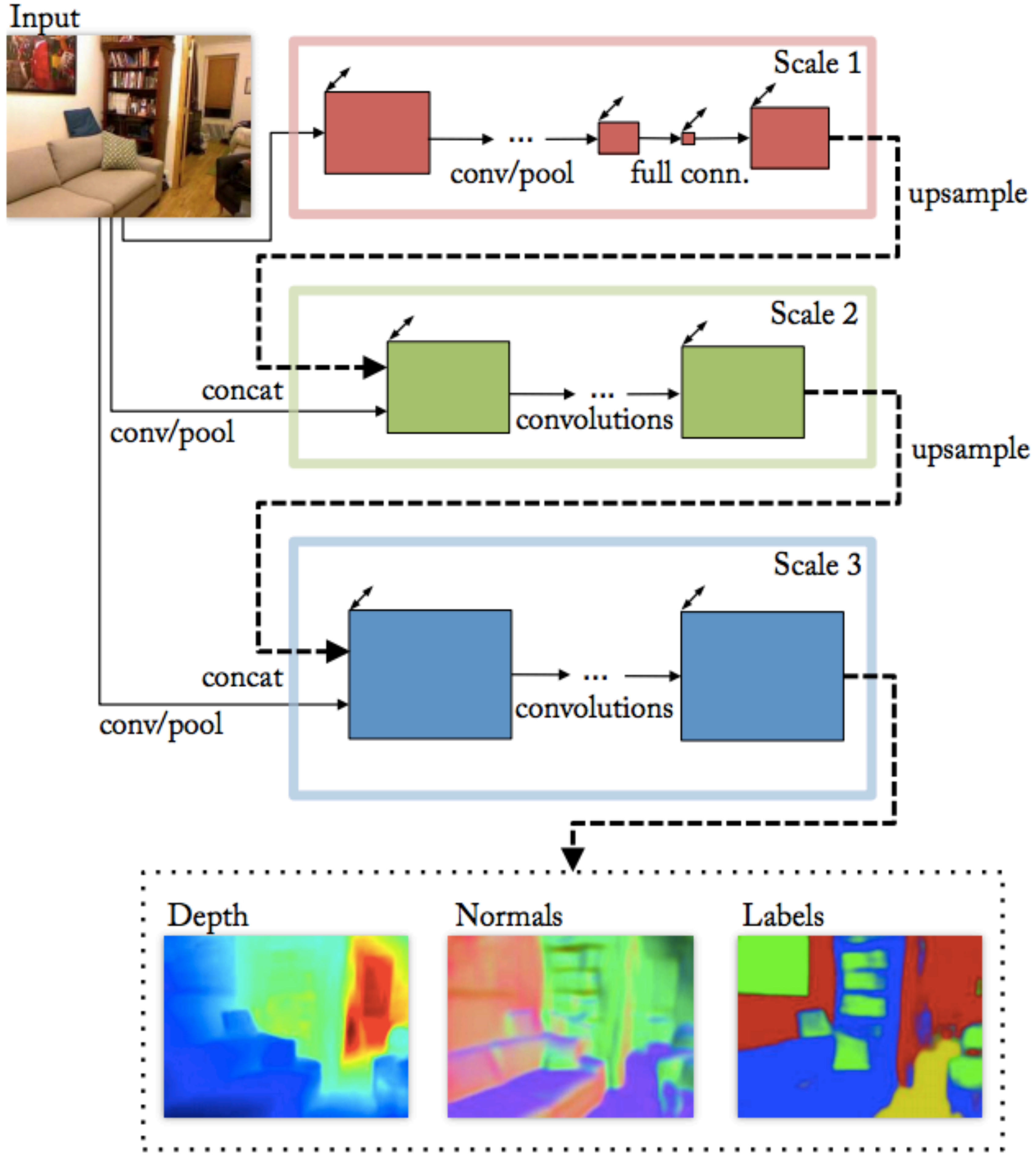




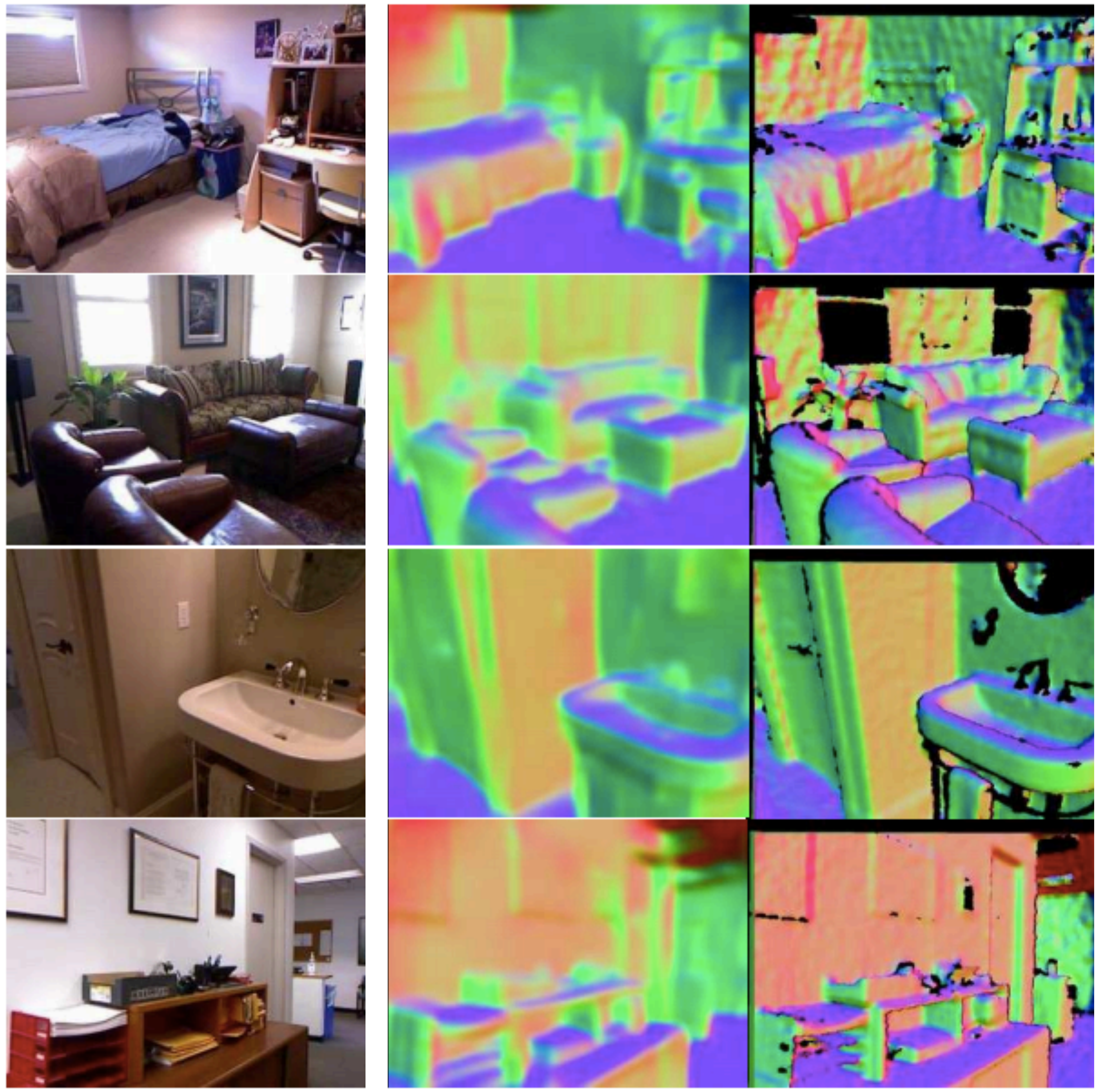
D. Eigen and R. Fergus, [Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture](), ICCV 2015

Slide credit: S. Lazebnik

Dense Prediction: Depth and normal estimation



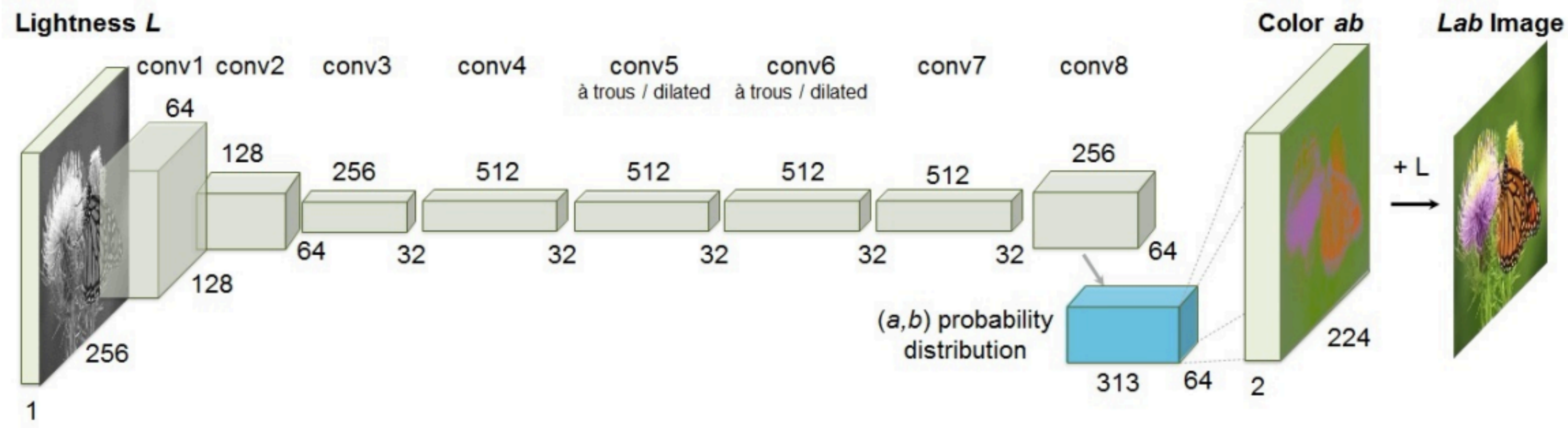
Predicted normals Ground truth

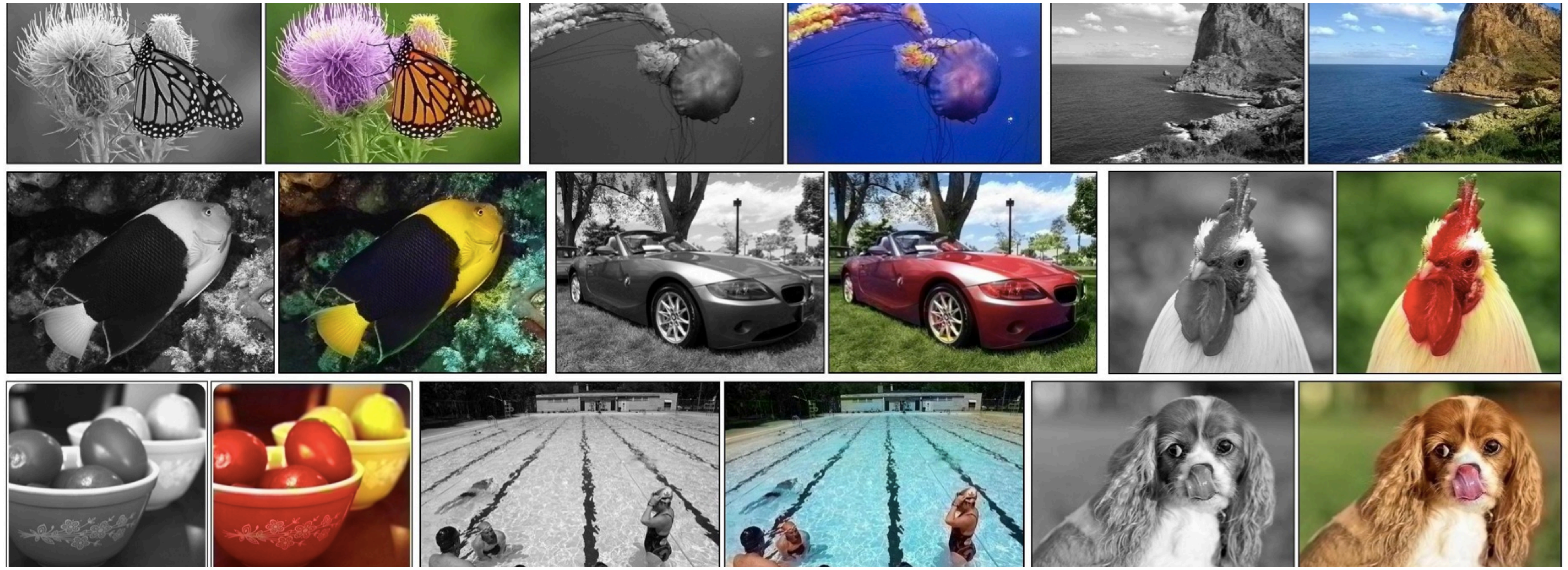


D. Eigen and R. Fergus, [Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-Scale Convolutional Architecture](), ICCV 2015

Slide credit: S. Lazebnik

Dense Prediction: Colorization





R. Zhang, P. Isola, and A. Efros, [Colorful Image Colorization](), ECCV 2016

Slide credit: S. Lazebnik