

Lecture 13: Object Detection and Image Segmentation – II

Announcements

- Midterm grades released yesterday
 - Submit regrade requests within a week from yesterday
- Homework 2 due **Thursday, March 26 — Today!**
- Project proposals due **Tuesday, March 31** —
 - Carefully read project proposal guidelines on the course page
 - Submit one proposal per team

Recap: Computer Vision Tasks

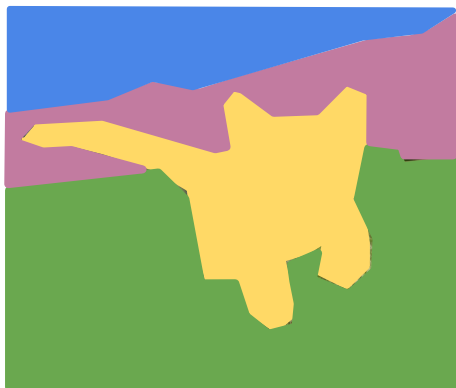
Classification



CAT

No spatial extent

Semantic Segmentation



**GRASS, CAT,
TREE, SKY**

No objects, just pixels

Object Detection



DOG, DOG, CAT

Multiple Object

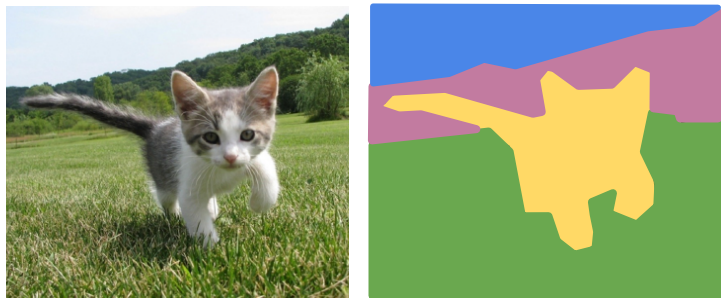
Instance Segmentation



DOG, DOG, CAT

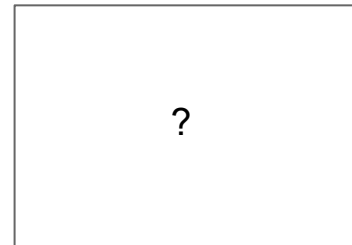
[This image is CC0 public domain](#)

Recap: Semantic Segmentation: The Problem



GRASS, **CAT**,
TREE, **SKY**, ...

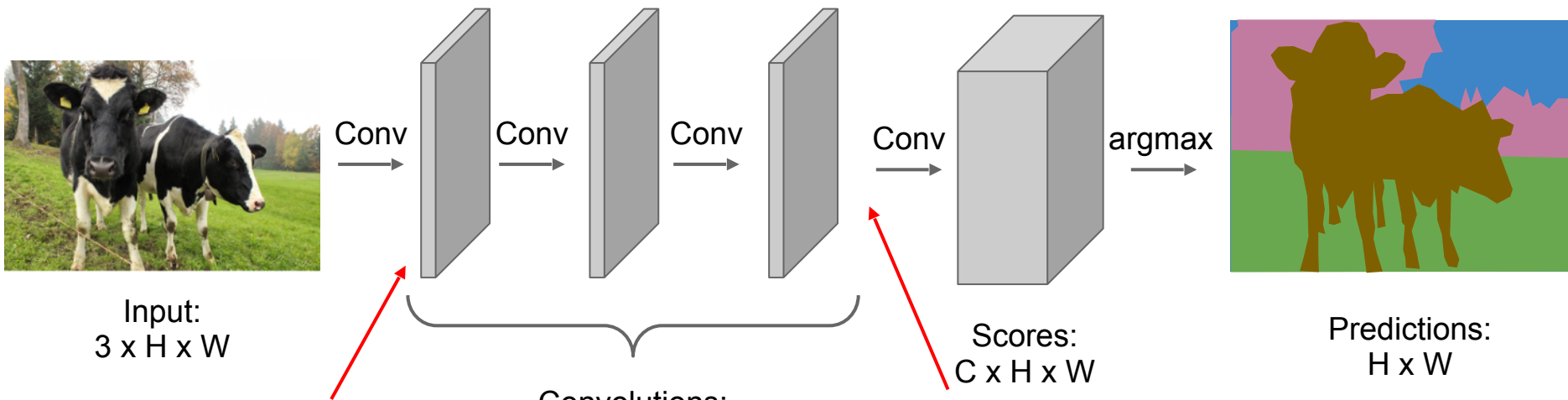
Paired training data: for each training image,
each pixel is labeled with a semantic category.



At test time, classify each pixel of a new image.

Semantic Segmentation Idea: Fully Convolutional

Design a network with only convolutional layers without downsampling operators to make predictions for pixels all at once!

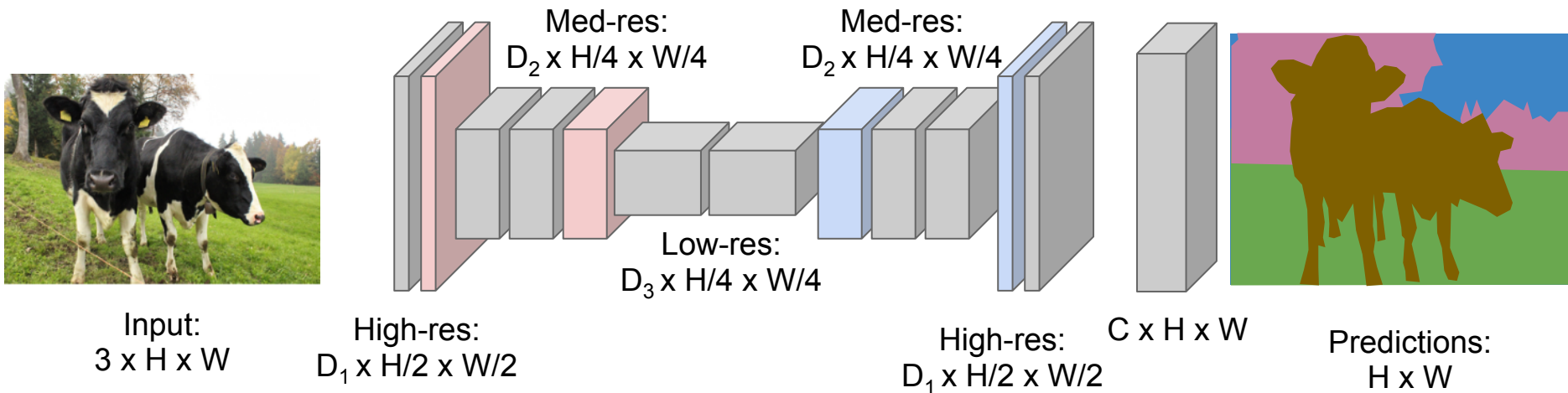


Problem #1: convolutions at original image resolution will be very expensive ...

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

Semantic Segmentation Idea: Fully Convolutional

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



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”

Nearest Neighbor

1	2
3	4



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

Input: 2 x 2

Output: 4 x 4

“Bed of Nails”

1	2
3	4

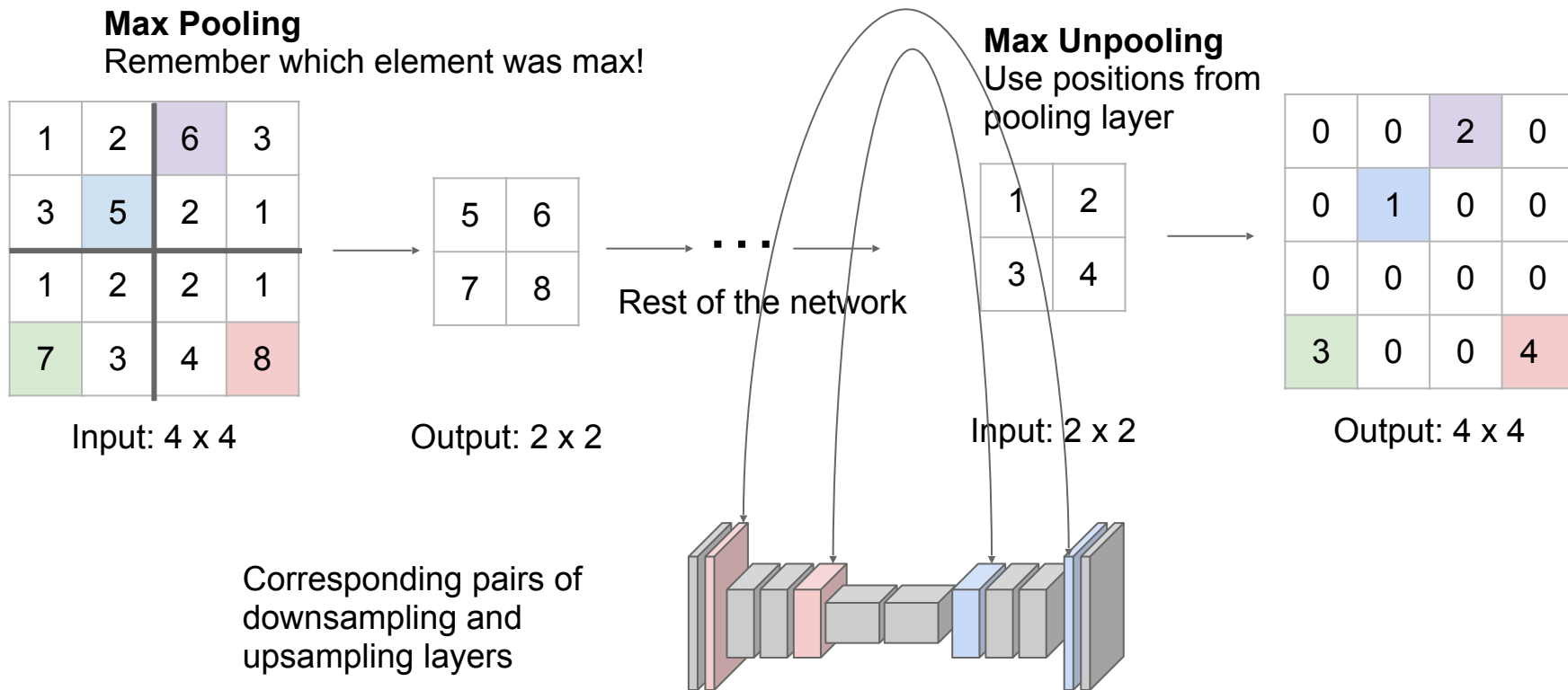


1	0	2	0
0	0	0	0
3	0	4	0
0	0	0	0

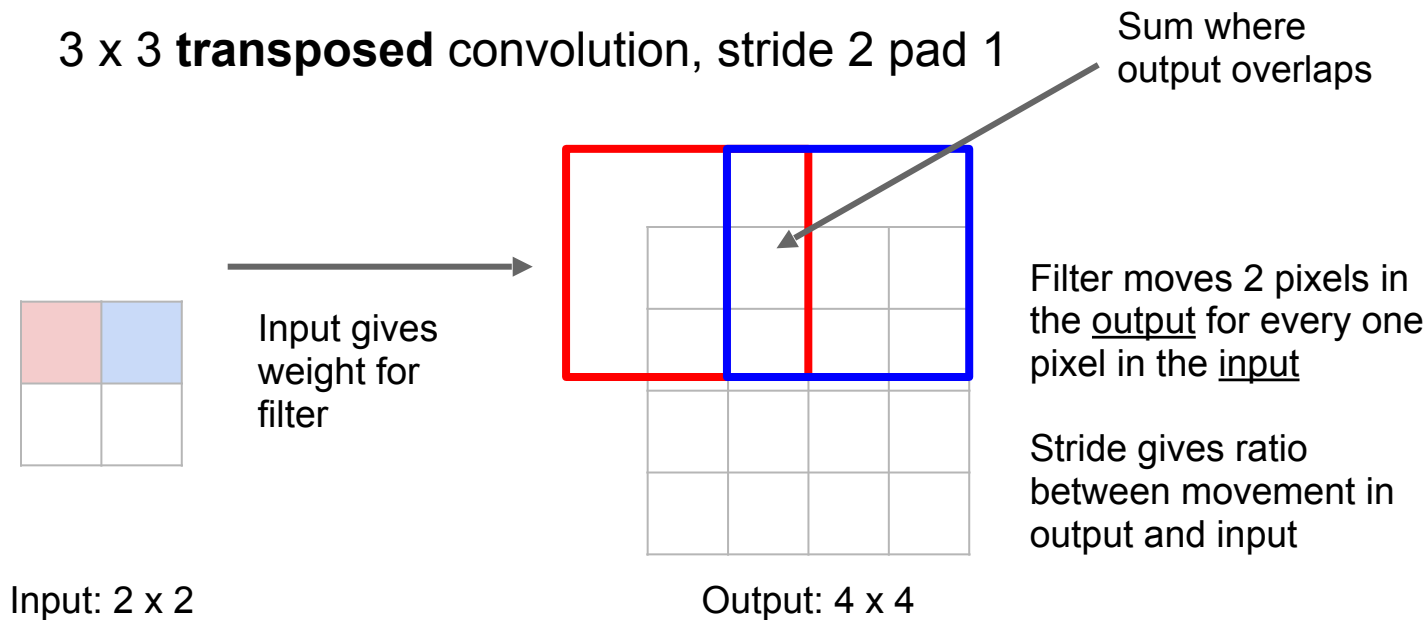
Input: 2 x 2

Output: 4 x 4

In-Network upsampling: “Max Unpooling”



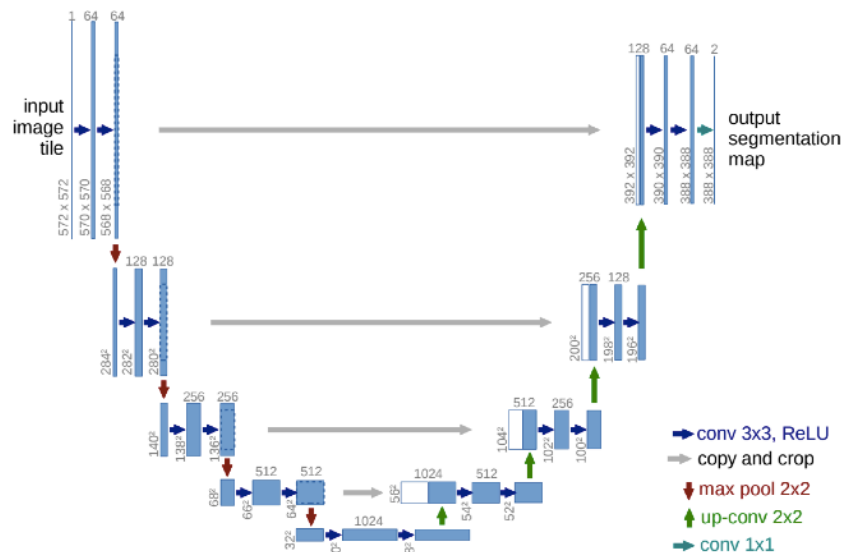
Learnable Upsampling: Transposed Convolution



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



Object Detection

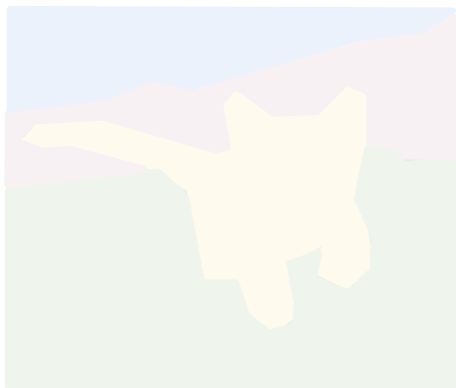
Classification



CAT

No spatial extent

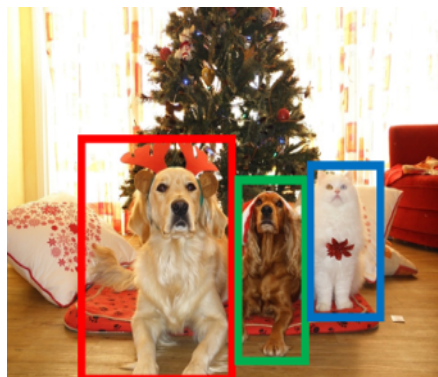
Semantic Segmentation



GRASS, CAT,
TREE, SKY

No objects, just pixels

Object Detection



DOG, DOG, CAT

Multiple Object

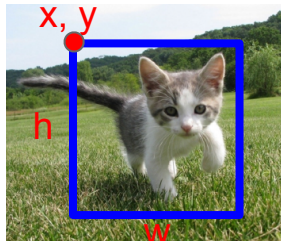
Instance Segmentation



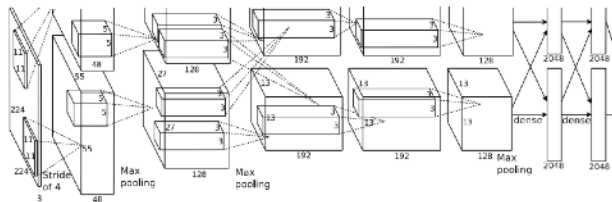
DOG, DOG, CAT

Object Detection: Single Object

(Classification + Localization)



This image is CC0 public domain



Fully Connected:
4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Correct label:
Cat

Softmax Loss

Multitask Loss

+ → **Loss**

Vector:
4096

Fully Connected:
4096 to 4

Box Coordinates
(x, y, w, h)

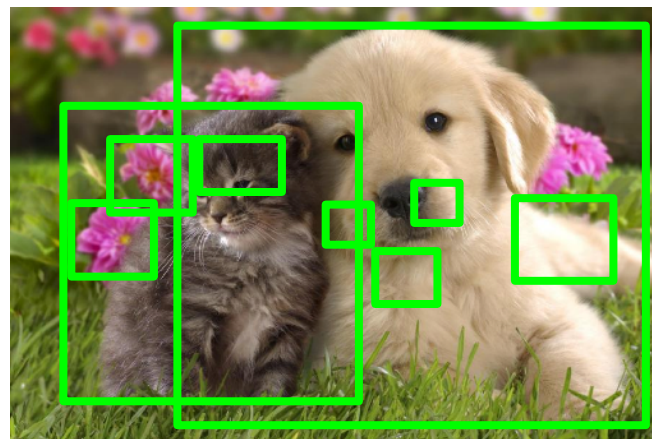
L2 Loss

Correct box:
(x', y', w', h')

Treat localization as a regression problem!

Region Proposals: Selective Search

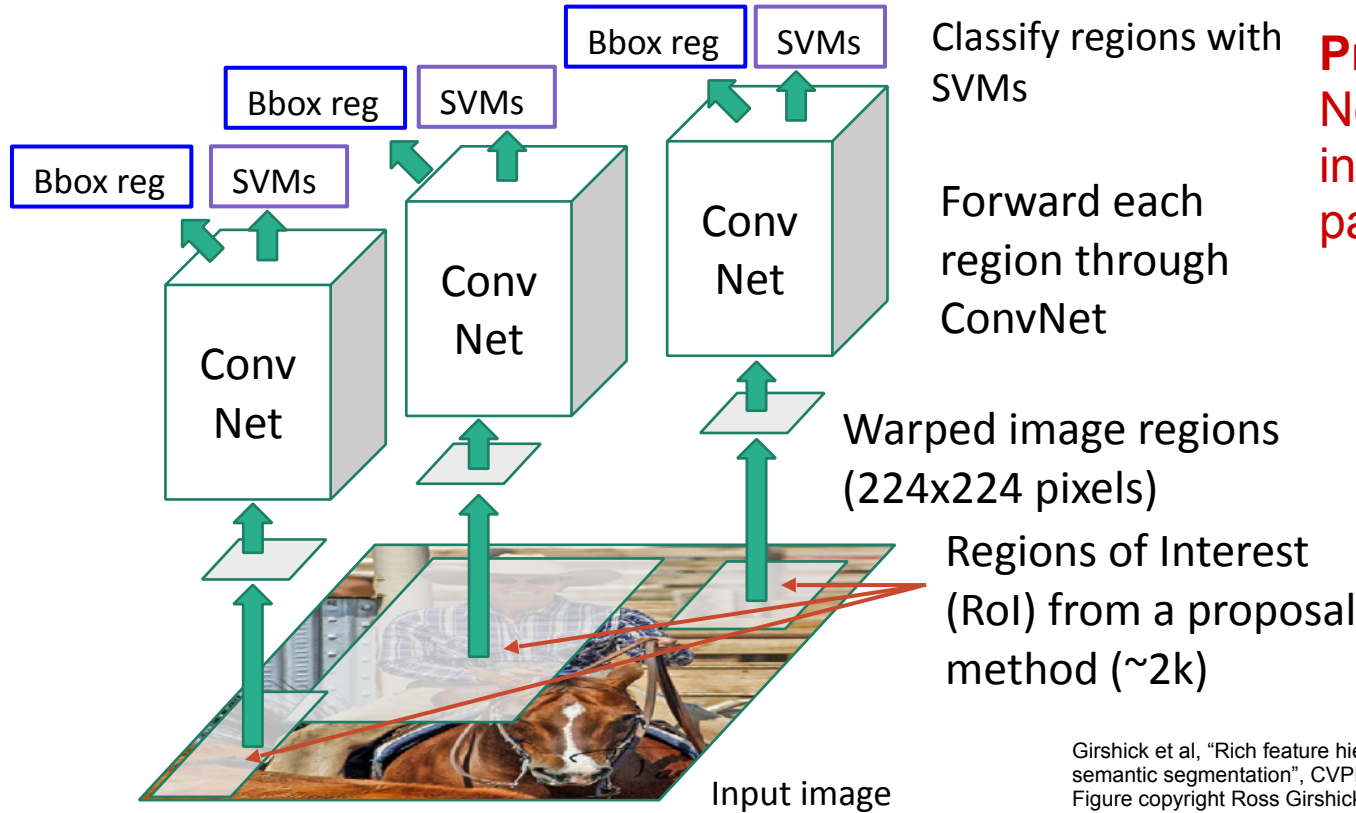
- Find “blobby” image regions that are likely to contain objects
- Relatively fast to run; e.g. Selective Search gives 2000 region proposals in a few seconds on CPU



Alexe et al, “Measuring the objectness of image windows”, TPAMI 2012
Uijlings et al, “Selective Search for Object Recognition”, IJCV 2013
Cheng et al, “BING: Binarized normed gradients for objectness estimation at 300fps”, CVPR 2014
Zitnick and Dollar, “Edge boxes: Locating object proposals from edges”, ECCV 2014

R-CNN

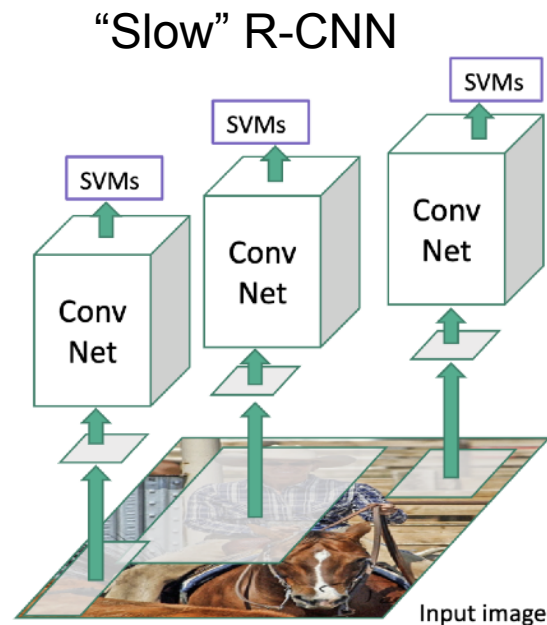
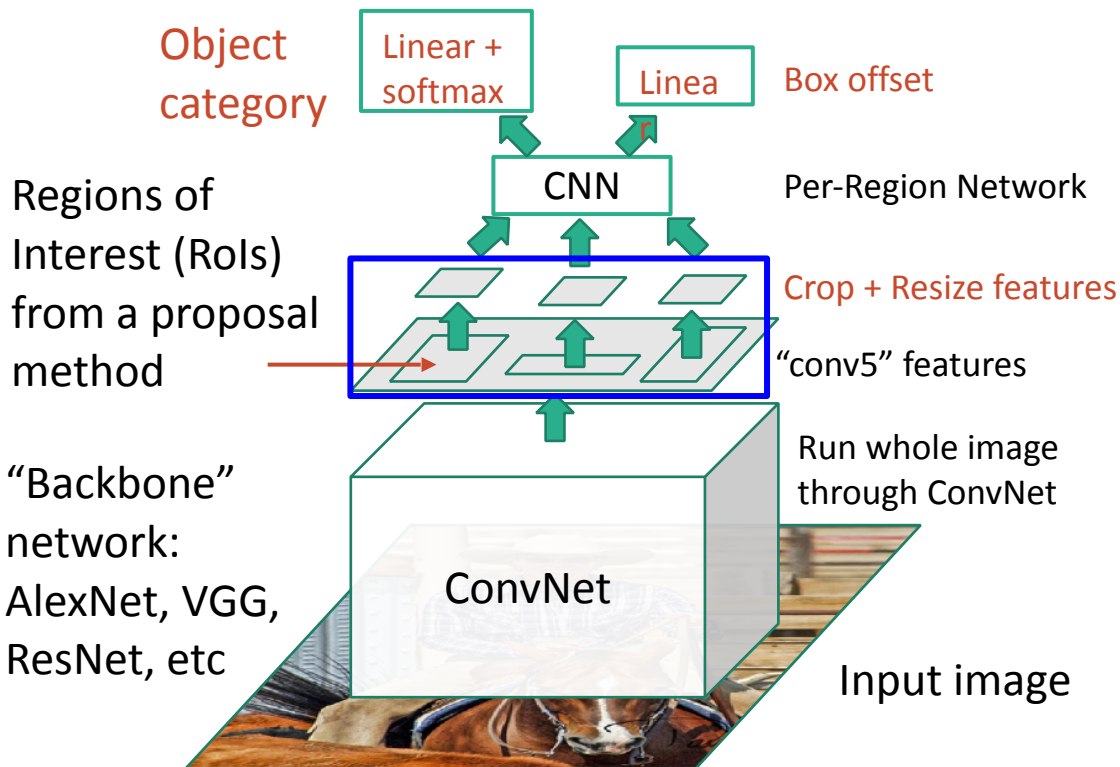
Predict “corrections” to the RoI: 4 numbers: (dx, dy, dw, dh)



Problem: Very slow!
Need to do ~2k
independent forward
passes for each image!

Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

Fast R-CNN



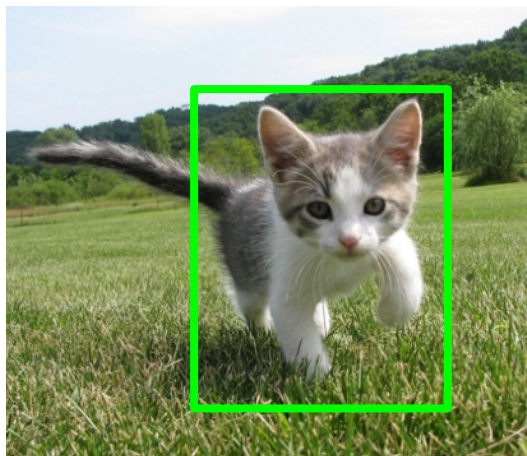
Girshick, "Fast R-CNN", ICCV 2015. Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

Cropping Features: RoI Pool

“Snap” to grid cells

Divide into 2x2 grid of (roughly) equal subregions

Project proposal onto features



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

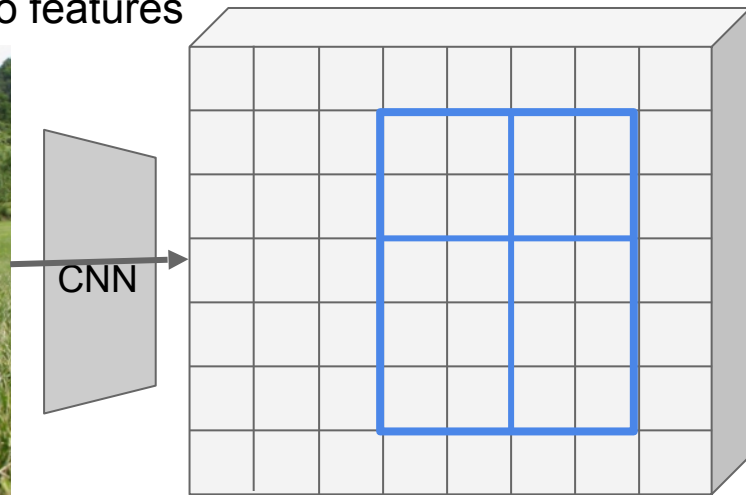
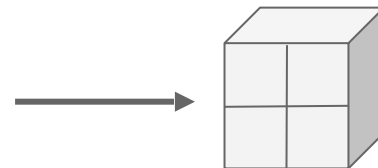


Image features: C x H x W
(e.g. 512 x 20 x 15)

Max-pool within each subregion



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!

Problem: Region features slightly misaligned

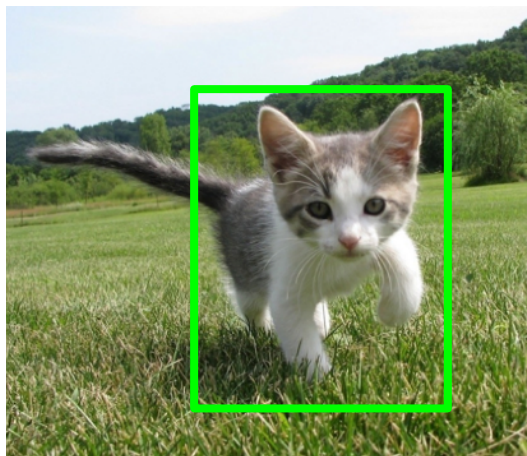
Girshick, “Fast R-CNN”, ICCV 2015.

Cropping Features: RoI Align

Sample at regular points in each subregion using bilinear interpolation

Project proposal onto features

No “snapping”!



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

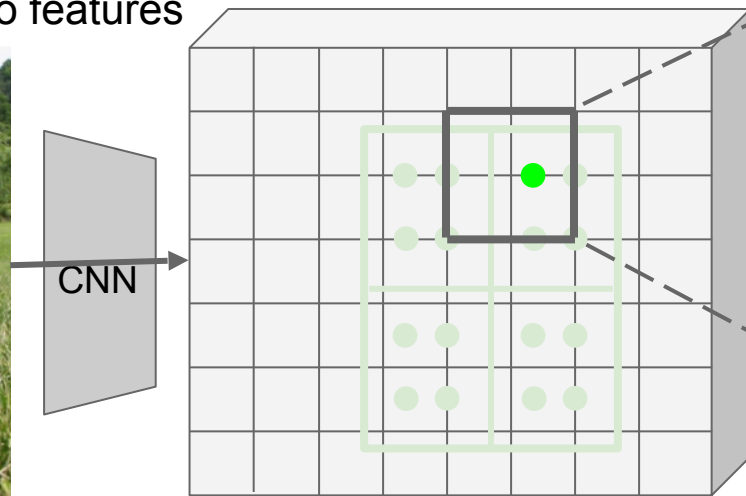
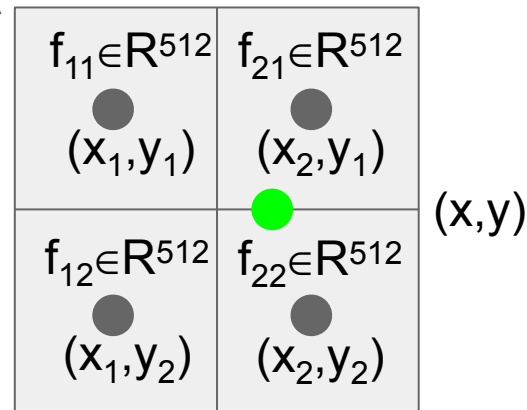


Image features: C x H x W
(e.g. 512 x 20 x 15)



Feature f_{xy} for point (x, y) is a linear combination of features at its four neighboring grid cells:

$$f_{xy} = \sum_{i,j=1}^2 f_{i,j} \max(0, 1 - |x - x_i|) \max(0, 1 - |y - y_j|)$$

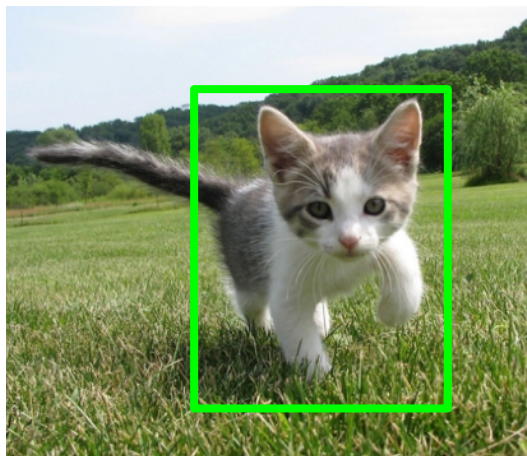
He et al, “Mask R-CNN”, ICCV 2017

Cropping Features: RoI Align

Sample at regular points in each subregion using bilinear interpolation

No “snapping”!

Project proposal onto features



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

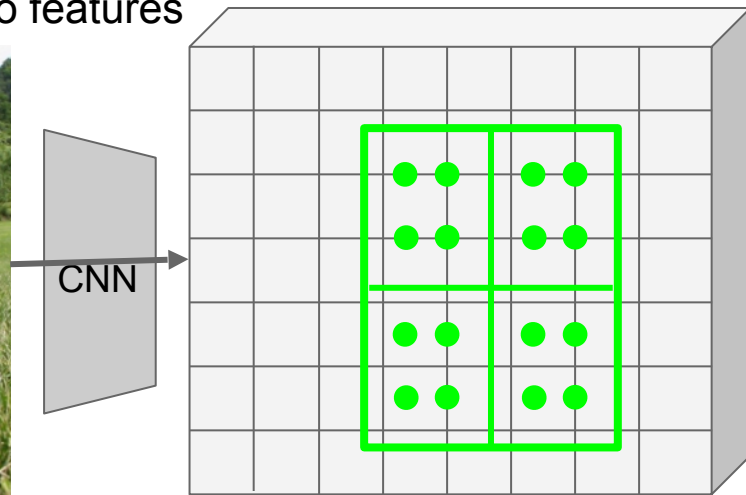
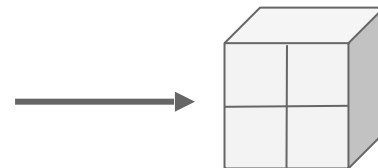


Image features: C x H x W
(e.g. 512 x 20 x 15)

Max-pool within each subregion

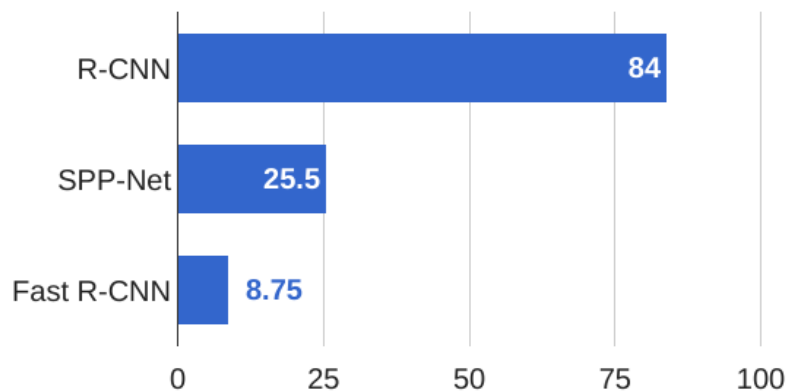


Region features
(here 512 x 2 x 2;
In practice e.g 512 x 7 x 7)

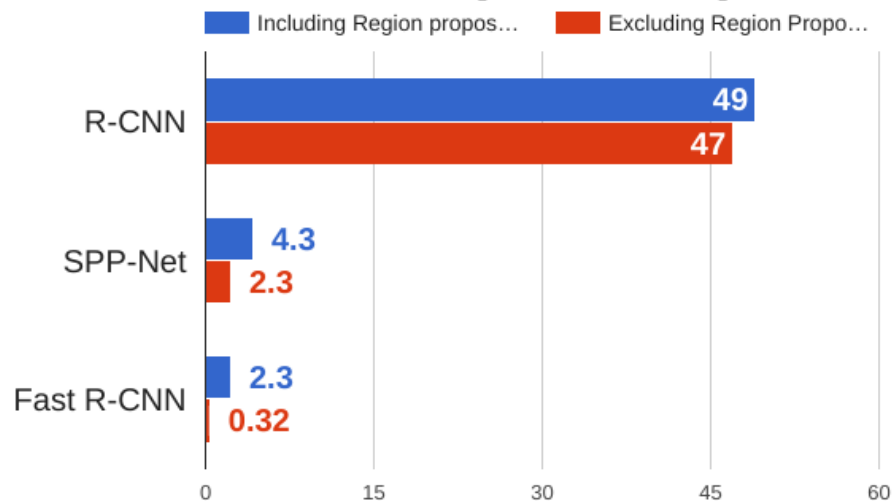
He et al, “Mask R-CNN”, ICCV 2017

R-CNN vs Fast R-CNN

Training time (Hours)



Test time (seconds)



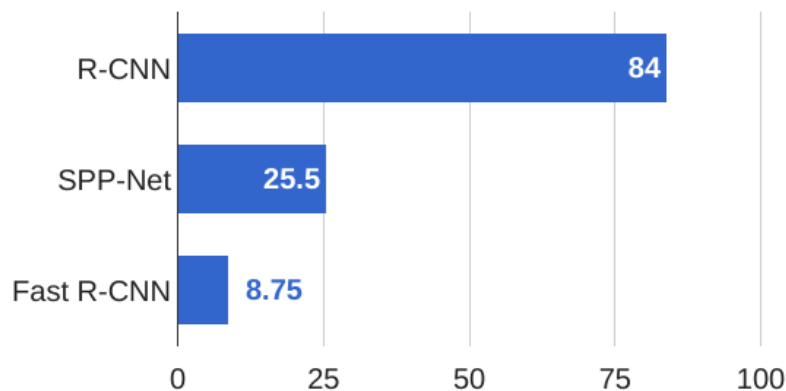
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

He et al, "Spatial pyramid pooling in deep convolutional networks for visual recognition", ECCV 2014

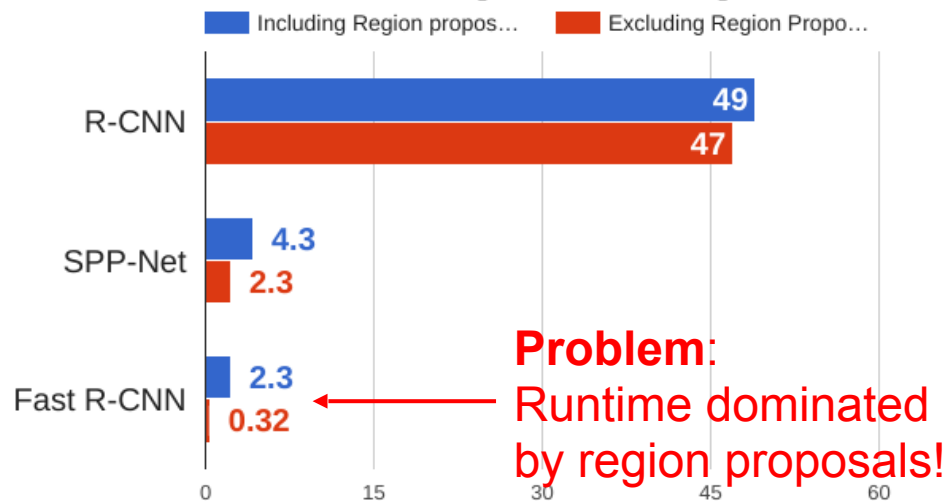
Girshick, "Fast R-CNN", ICCV 2015

R-CNN vs Fast R-CNN

Training time (Hours)



Test time (seconds)



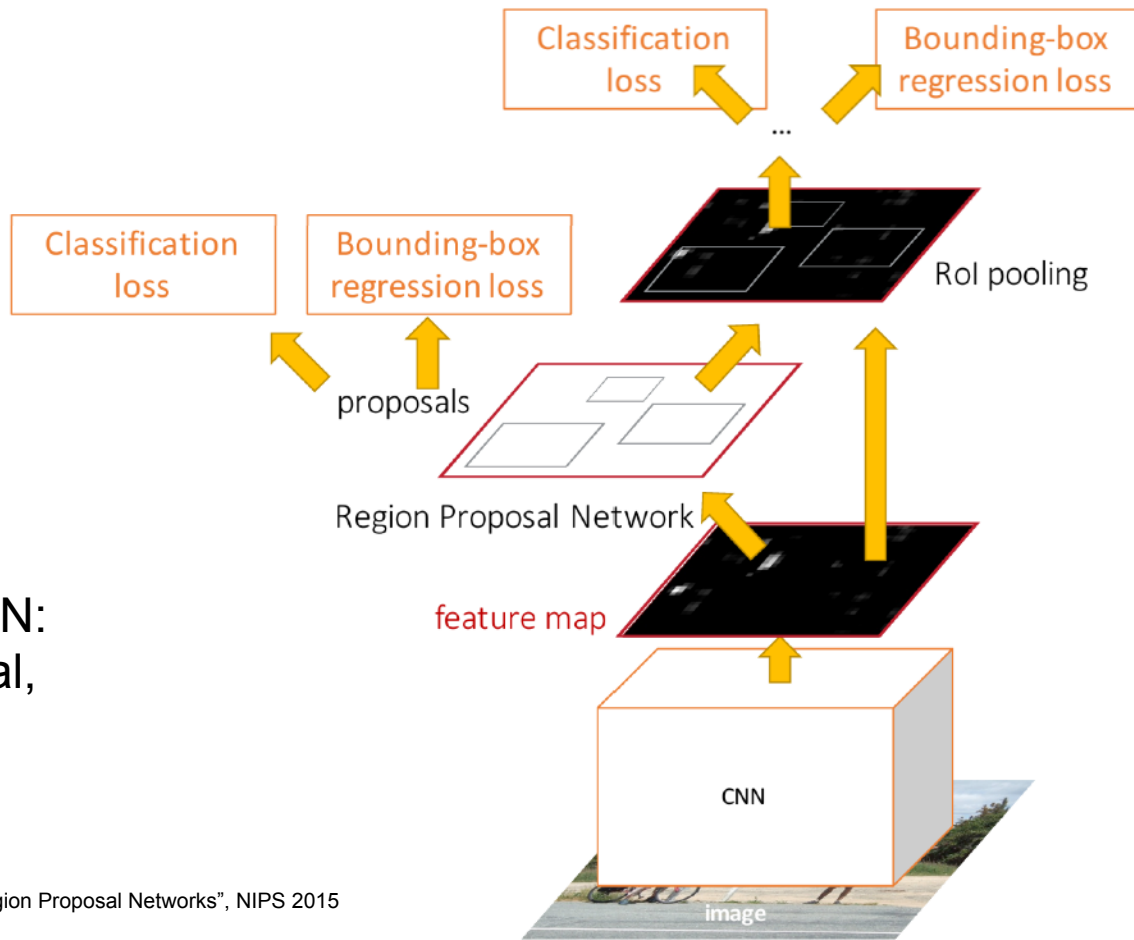
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
He et al, "Spatial pyramid pooling in deep convolutional networks for visual recognition", ECCV 2014
Girshick, "Fast R-CNN", ICCV 2015

Faster R-CNN:

Make CNN do proposals!

Insert **Region Proposal Network (RPN)** to predict proposals from features

Otherwise same as Fast R-CNN:
Crop features for each proposal,
classify each one



Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

Region Proposal Network



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

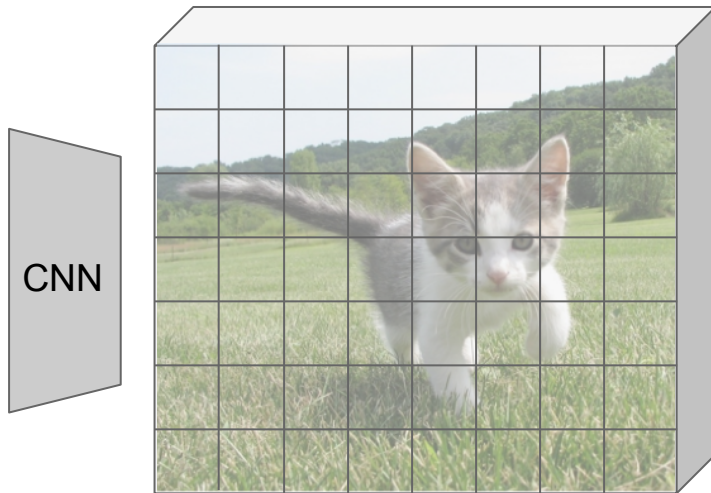


Image features
(e.g. 512 x 20 x 15)

Region Proposal Network

Imagine an **anchor box** of fixed size at each point in the feature map



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

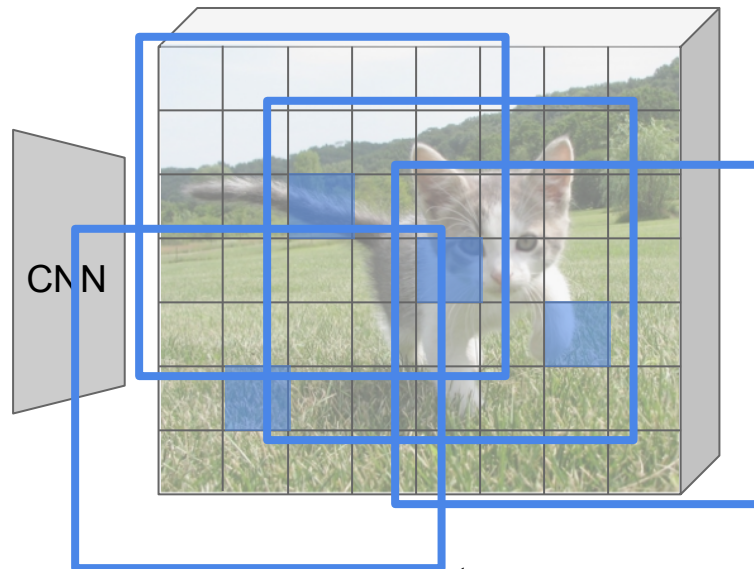


Image features
(e.g. 512 x 20 x 15)

Region Proposal Network

Imagine an **anchor box** of fixed size at each point in the feature map



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

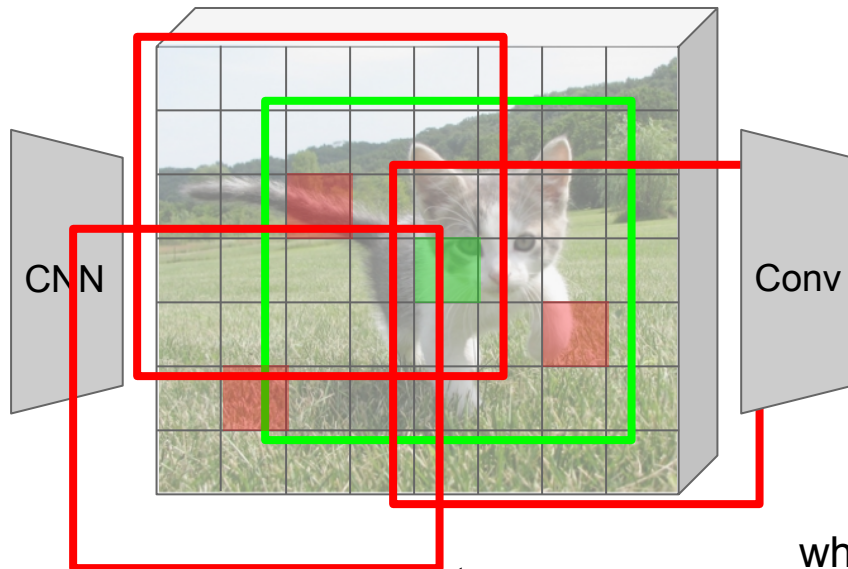


Image features
(e.g. 512 x 20 x 15)

Anchor is an object?
1 x 20 x 15

At each point, predict whether the corresponding anchor contains an object (binary classification)

Region Proposal Network

Imagine an **anchor box** of fixed size at each point in the feature map



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

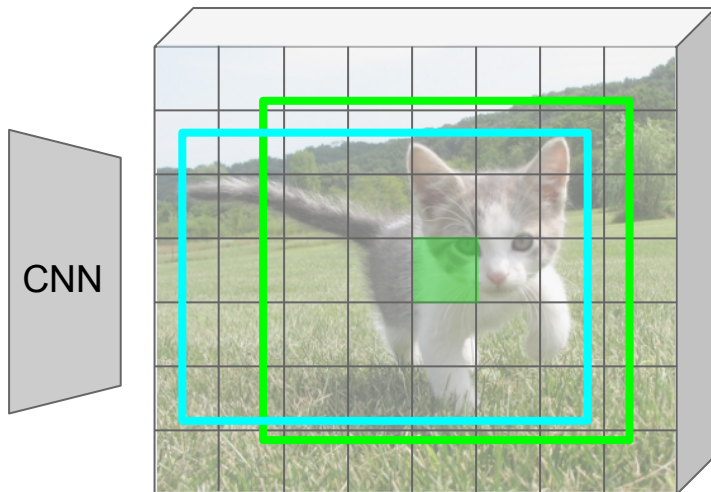


Image features
(e.g. 512 x 20 x 15)



Anchor is an object?
1 x 20 x 15

Box corrections
4 x 20 x 15

For positive boxes, also predict a corrections from the anchor to the ground-truth box (regress 4 numbers per pixel)

Region Proposal Network

In practice use K different anchor boxes of different size / scale at each point



Input Image
(e.g. $3 \times 640 \times 480$)

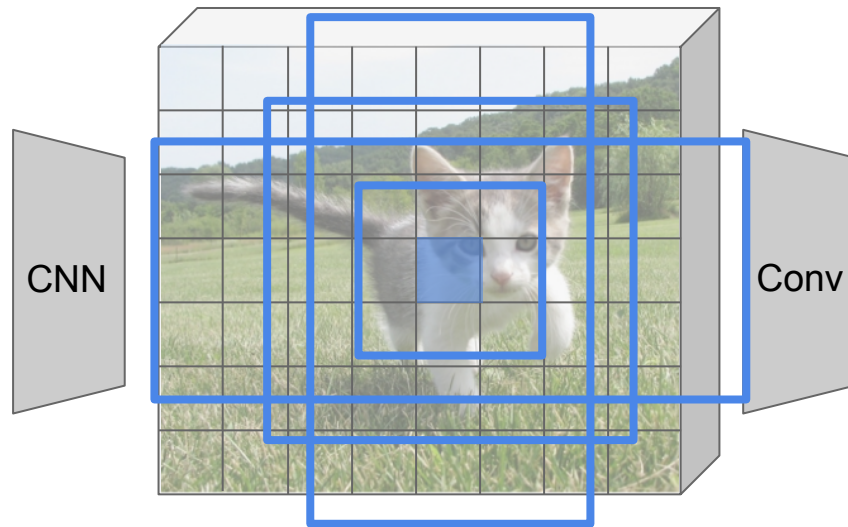


Image features
(e.g. $512 \times 20 \times 15$)

Anchor is an object?
 $\mathbf{K} \times 20 \times 15$

Box transforms
 $\mathbf{4K} \times 20 \times 15$

Region Proposal Network

In practice use K different anchor boxes of different size / scale at each point



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

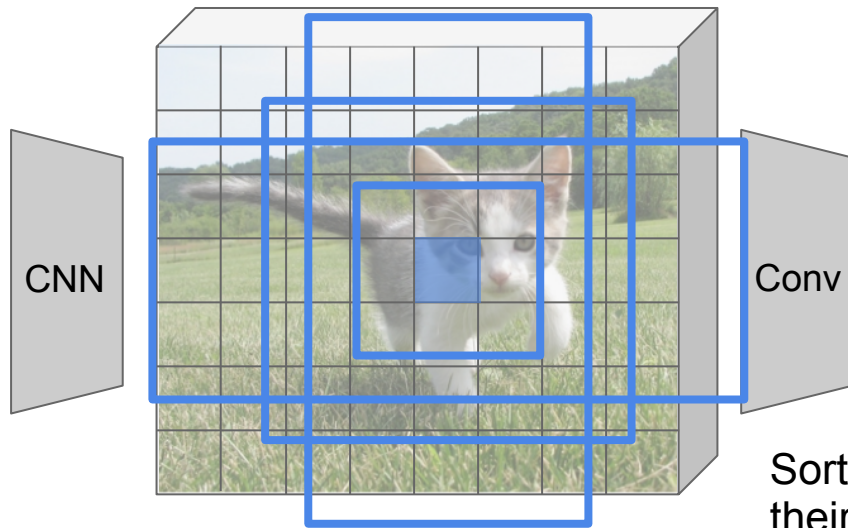


Image features
(e.g. 512 x 20 x 15)

Anchor is an object?
 $K \times 20 \times 15$

Box transforms
 $4K \times 20 \times 15$

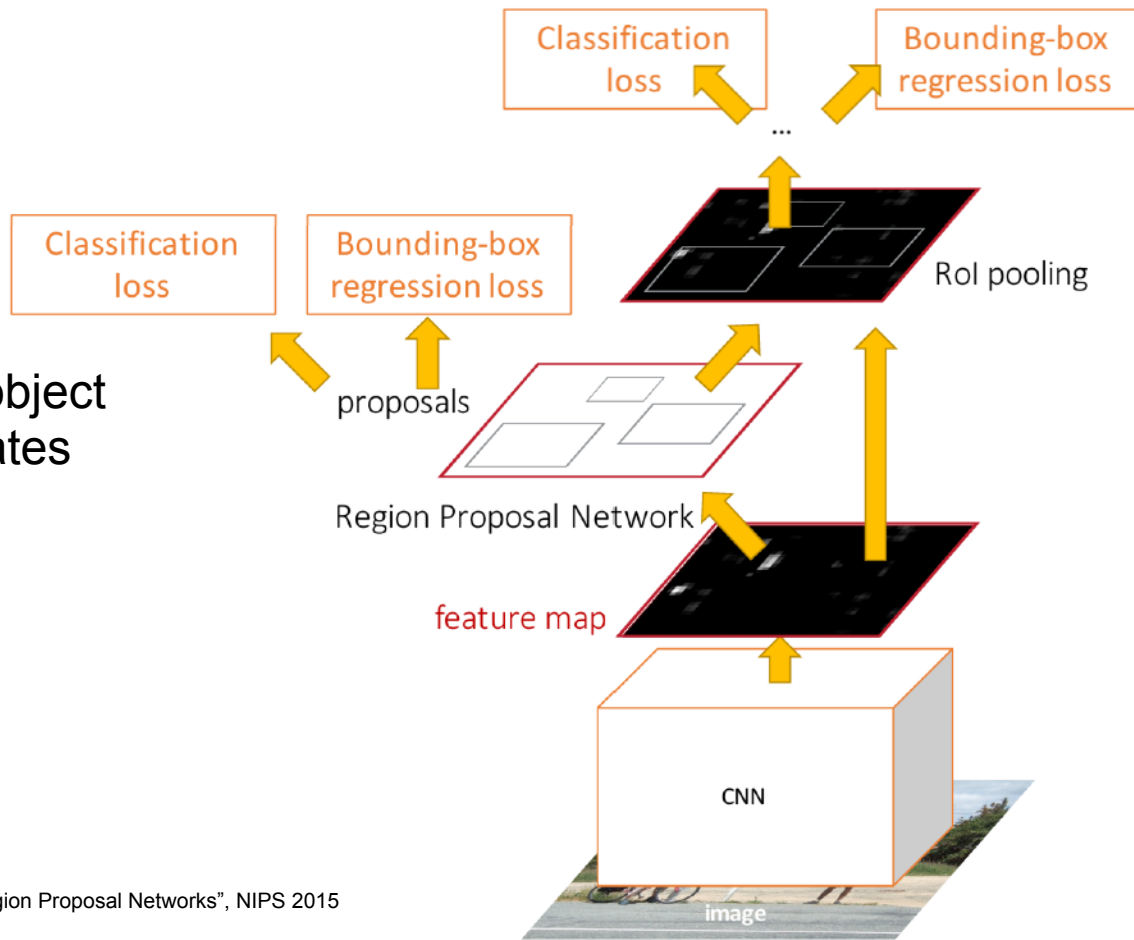
Sort the $K \times 20 \times 15$ boxes by their “objectness” score, take top ~ 300 as our proposals

Faster R-CNN:

Make CNN do proposals!

Jointly train with 4 losses:

1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates

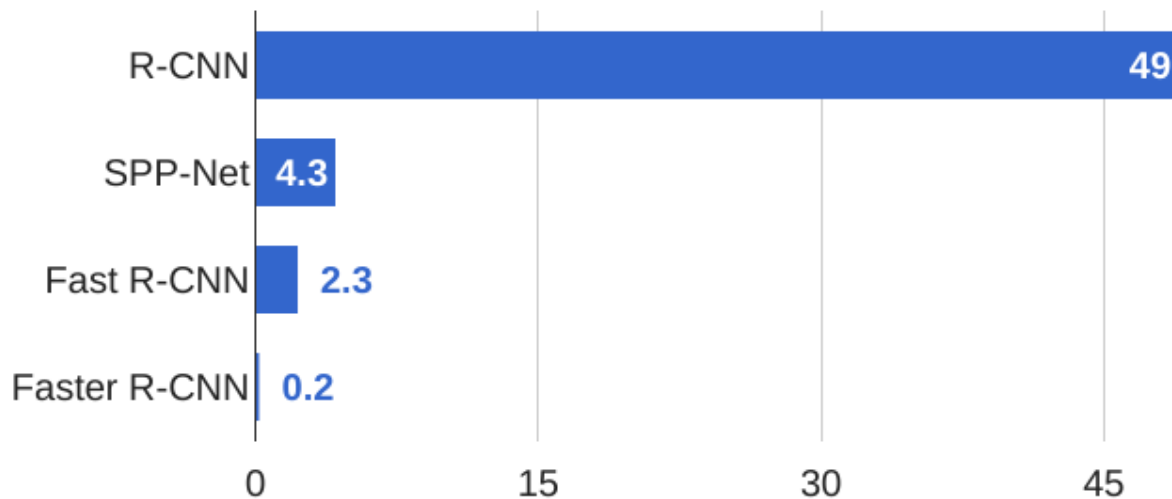


Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

Faster R-CNN:

Make CNN do proposals!

R-CNN Test-Time Speed

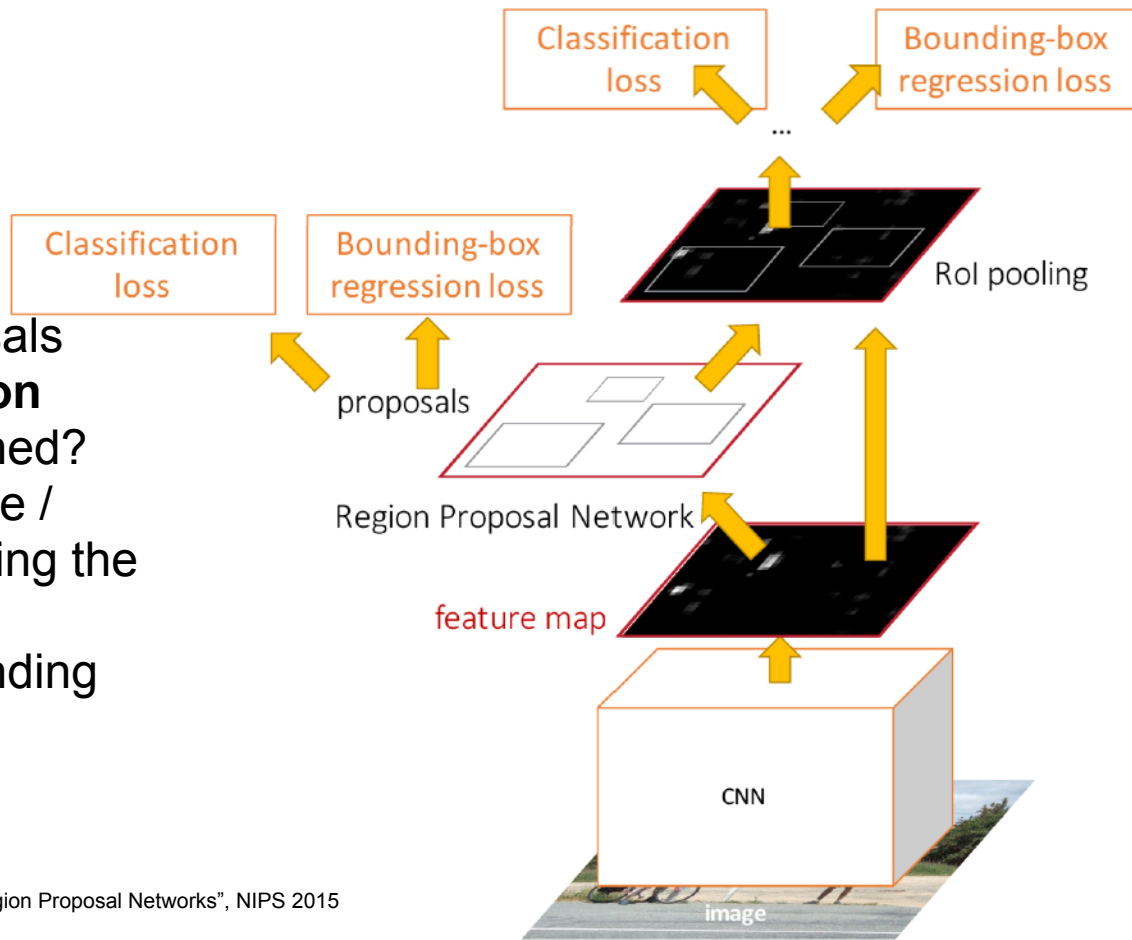


Faster R-CNN:

Make CNN do proposals!

Glossing over many details:

- Ignore overlapping proposals with **non-max suppression**
- How are anchors determined?
- How do we sample positive / negative samples for training the RPN?
- How to parameterize bounding box regression?



Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

Faster R-CNN:

Make CNN do proposals!

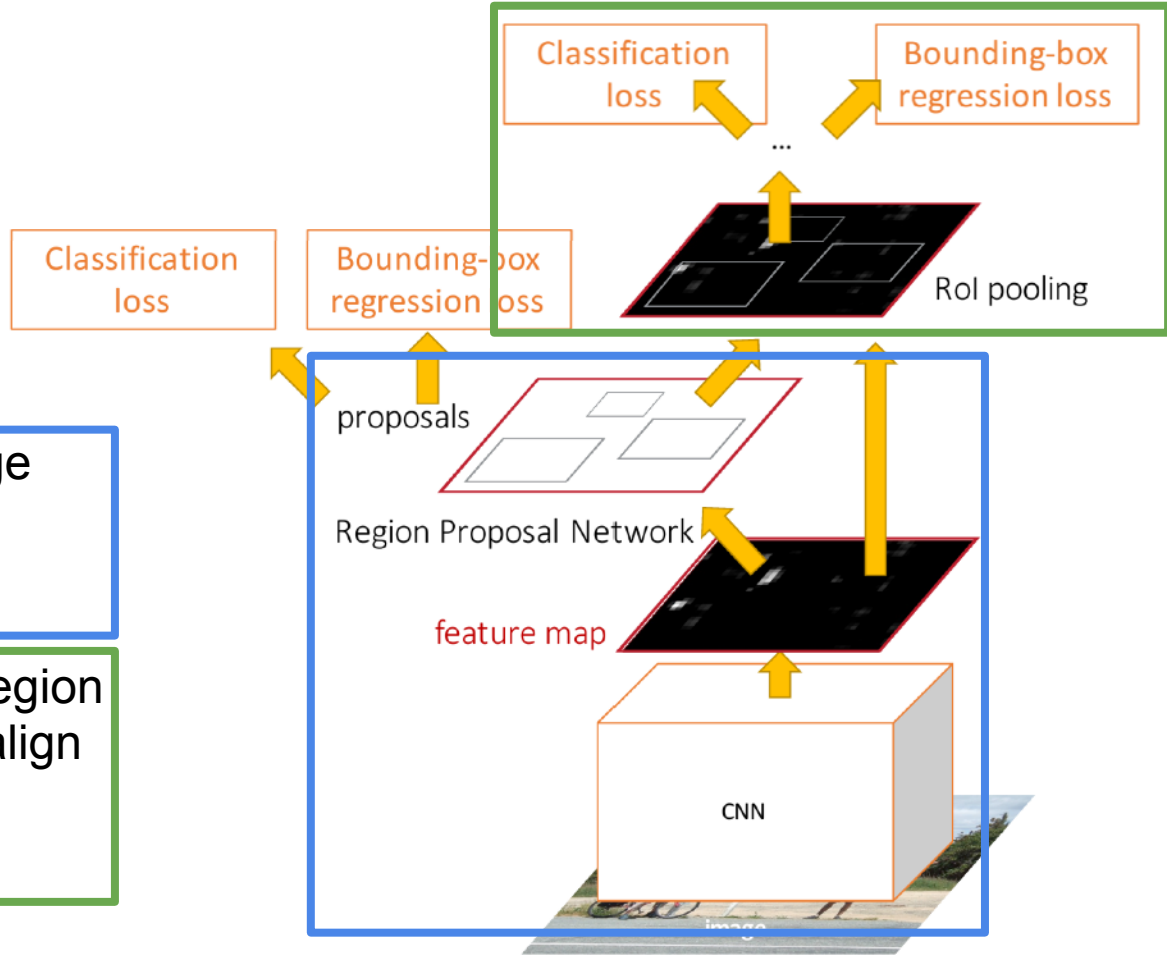
Faster R-CNN is a
Two-stage object detector

First stage: Run once per image

- Backbone network
- Region proposal network

Second stage: Run once per region

- Crop features: RoI pool / align
- Predict object class
- Prediction bbox offset



Faster R-CNN:

Make CNN do proposals!

Faster R-CNN is a **Two-stage object detector**

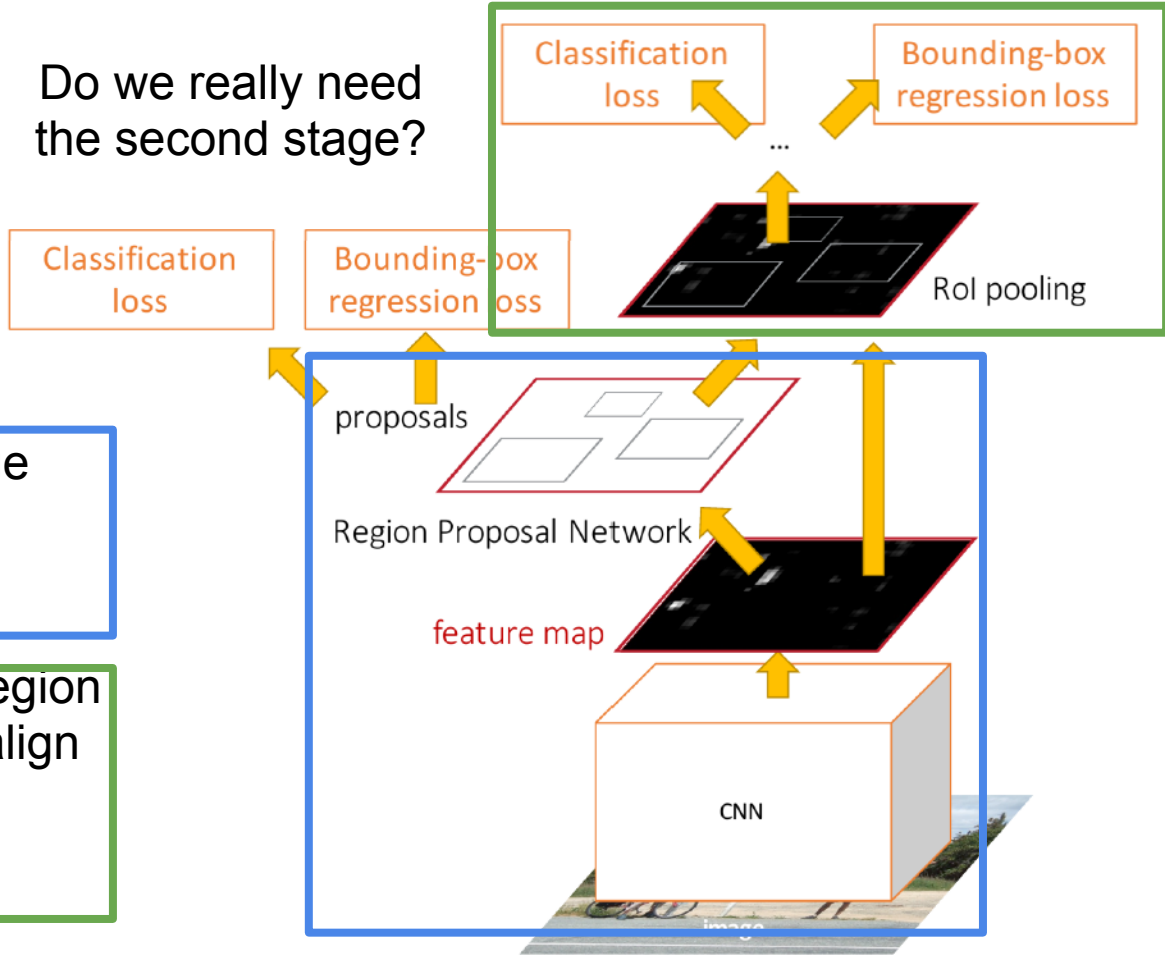
First stage: Run once per image

- Backbone network
- Region proposal network

Second stage: Run once per region

- Crop features: RoI pool / align
- Predict object class
- Prediction bbox offset

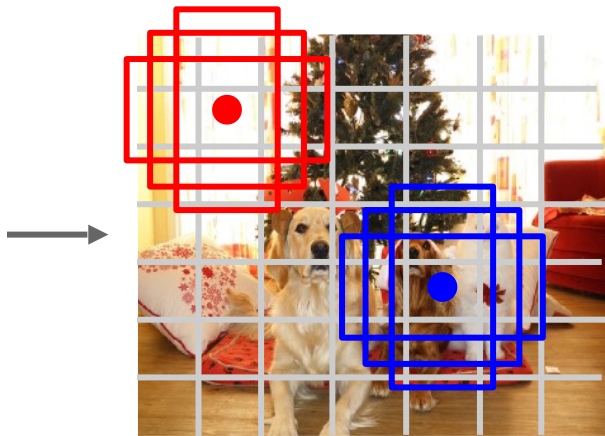
Do we really need the second stage?



Single-Stage Object Detectors: YOLO / SSD / RetinaNet



Input image
 $3 \times H \times W$



Divide image into grid
 7×7
Image a set of **base boxes**
centered at each grid cell
Here $B = 3$

- Within each grid cell:
- Regress from each of the B base boxes to a final box with 5 numbers: $(dx, dy, dh, dw, confidence)$
 - Predict scores for each of C classes (including background as a class)
 - Looks a lot like RPN, but category-specific!

Output:
 $7 \times 7 \times (5 * B + C)$

Redmon et al, "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016
Liu et al, "SSD: Single-Shot MultiBox Detector", ECCV 2016
Lin et al, "Focal Loss for Dense Object Detection", ICCV 2017

Object Detection: Lots of variables ...

Backbone Network

VGG16

ResNet-101

Inception V2

Inception V3

Inception

ResNet

MobileNet

“Meta-Architecture”

Two-stage: Faster R-CNN

Single-stage: YOLO / SSD

Hybrid: R-FCN

Image Size

Region Proposals

...

Takeaways

Faster R-CNN is slower
but more accurate

SSD is much faster but
not as accurate

Bigger / Deeper
backbones work better

Huang et al, “Speed/accuracy trade-offs for modern convolutional object detectors”, CVPR 2017

R-FCN: Dai et al, “R-FCN: Object Detection via Region-based Fully Convolutional Networks”, NIPS 2016

Inception-V2: Ioffe and Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”, ICML 2015

Inception V3: Szegedy et al, “Rethinking the Inception Architecture for Computer Vision”, arXiv 2016

Inception ResNet: Szegedy et al, “Inception-V4, Inception-ResNet and the Impact of Residual Connections on Learning”, arXiv 2016

MobileNet: Howard et al, “Efficient Convolutional Neural Networks for Mobile Vision Applications”, arXiv 2017

Object Detection: Lots of variables ...

Backbone Network

VGG16

ResNet-101

Inception V2

Inception V3

Inception

ResNet

MobileNet

“Meta-Architecture”

Two-stage: Faster R-CNN

Single-stage: YOLO / SSD

Hybrid: R-FCN

Image Size

Region Proposals

...

Takeaways

Faster R-CNN is slower
but more accurate

SSD is much faster but
not as accurate

Bigger / Deeper
backbones work better

Huang et al, “Speed/accuracy trade-offs for modern convolutional object detectors”, CVPR 2017

Zou et al, “Object Detection in 20 Years: A Survey”, arXiv 2019

R-FCN: Dai et al, “R-FCN: Object Detection via Region-based Fully Convolutional Networks”, NIPS 2016

Inception-V2: Ioffe and Szegedy, “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”, ICML 2015

Inception V3: Szegedy et al, “Rethinking the Inception Architecture for Computer Vision”, arXiv 2016

Inception ResNet: Szegedy et al, “Inception-V4, Inception-ResNet and the Impact of Residual Connections on Learning”, arXiv 2016

MobileNet: Howard et al, “Efficient Convolutional Neural Networks for Mobile Vision Applications”, arXiv 2017

Instance Segmentation

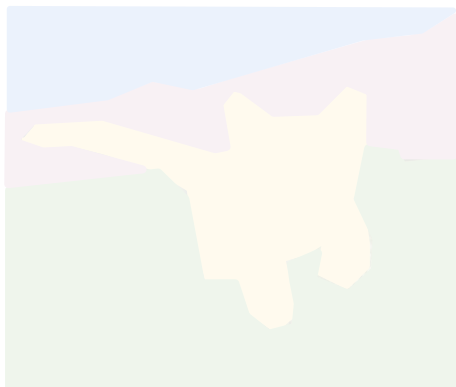
Classification



CAT

No spatial extent

Semantic Segmentation



GRASS, CAT,
TREE, SKY

No objects, just pixels

Object Detection



DOG, DOG, CAT

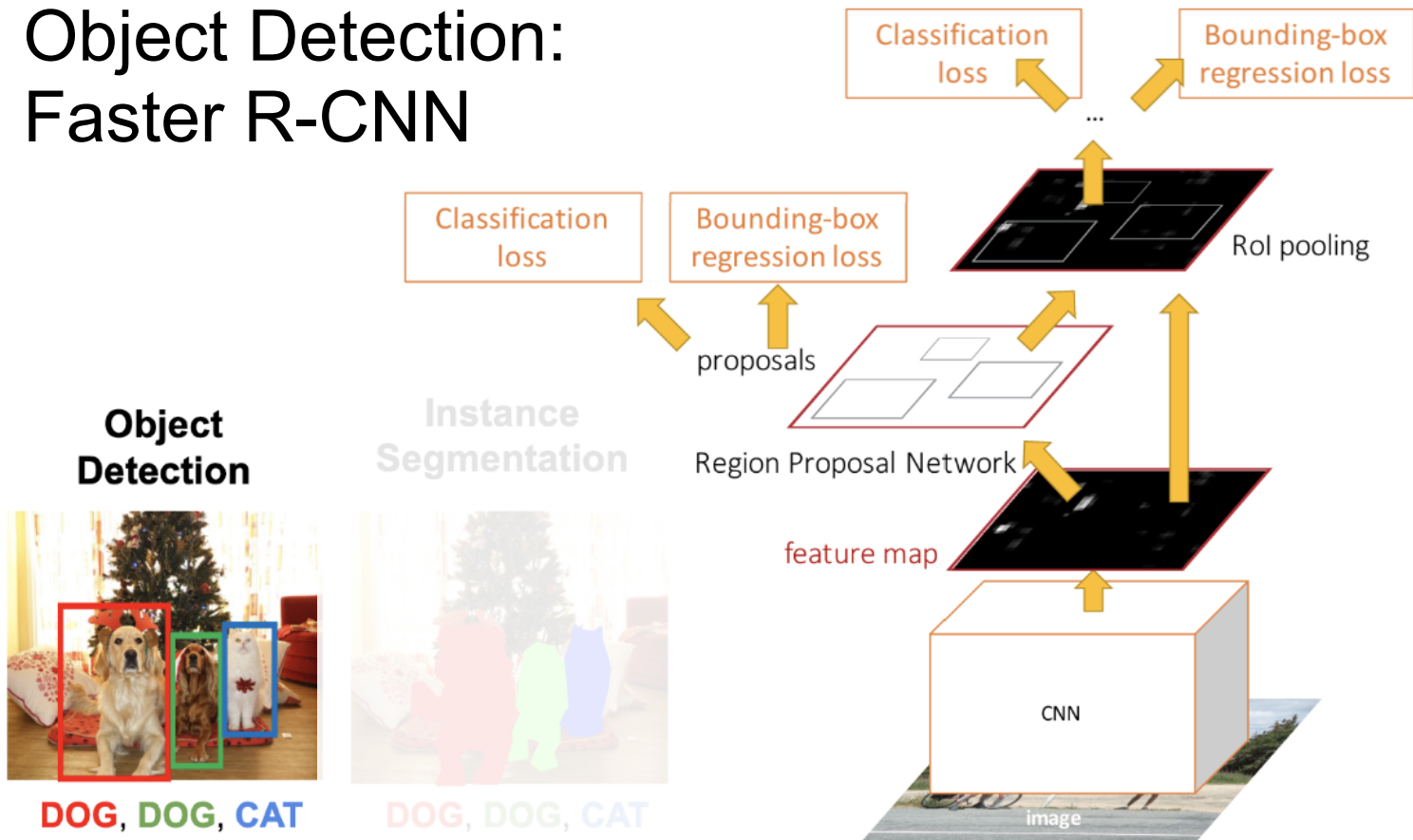
Multiple Object

Instance Segmentation

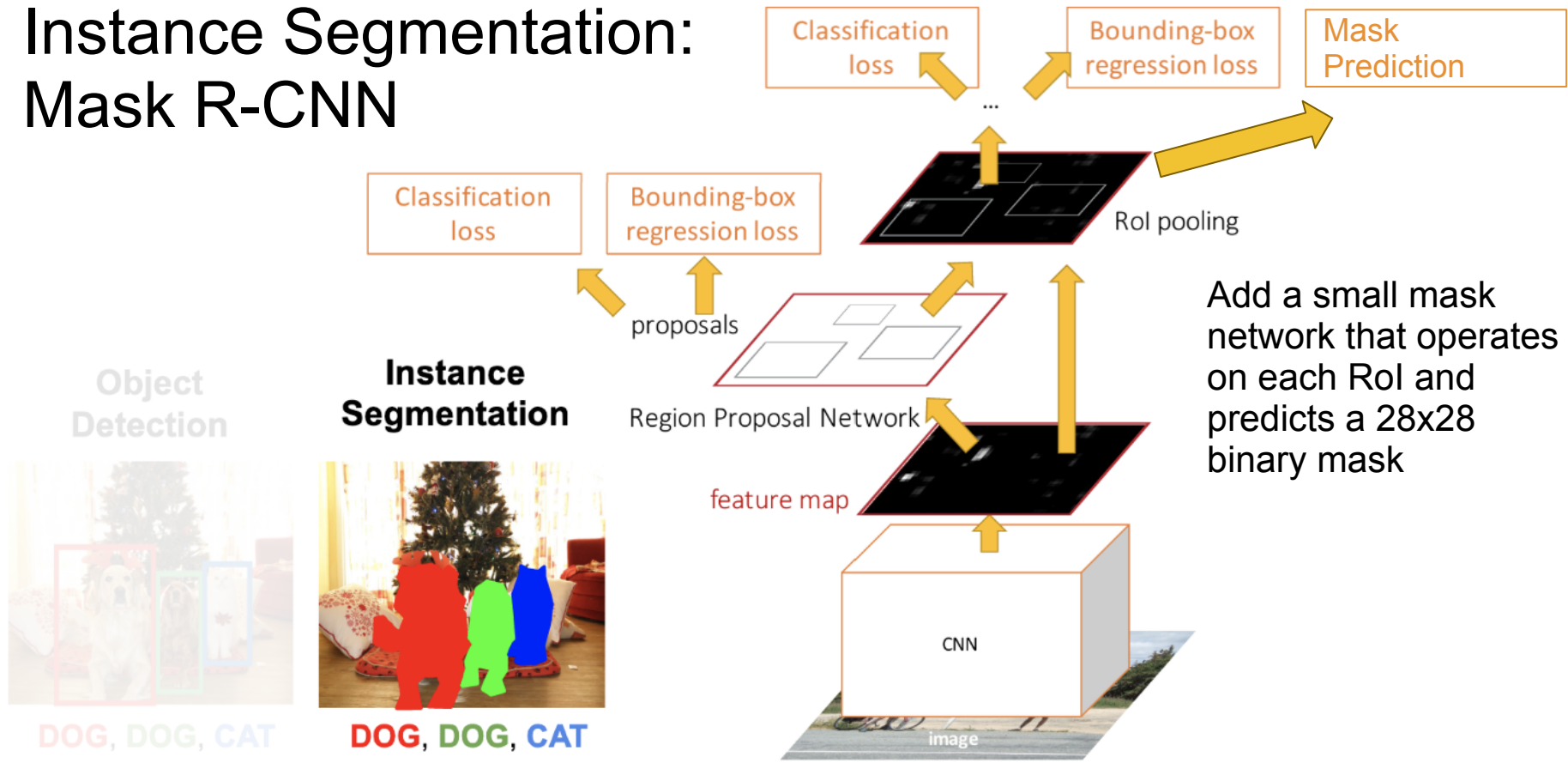


DOG, DOG, CAT

Object Detection: Faster R-CNN

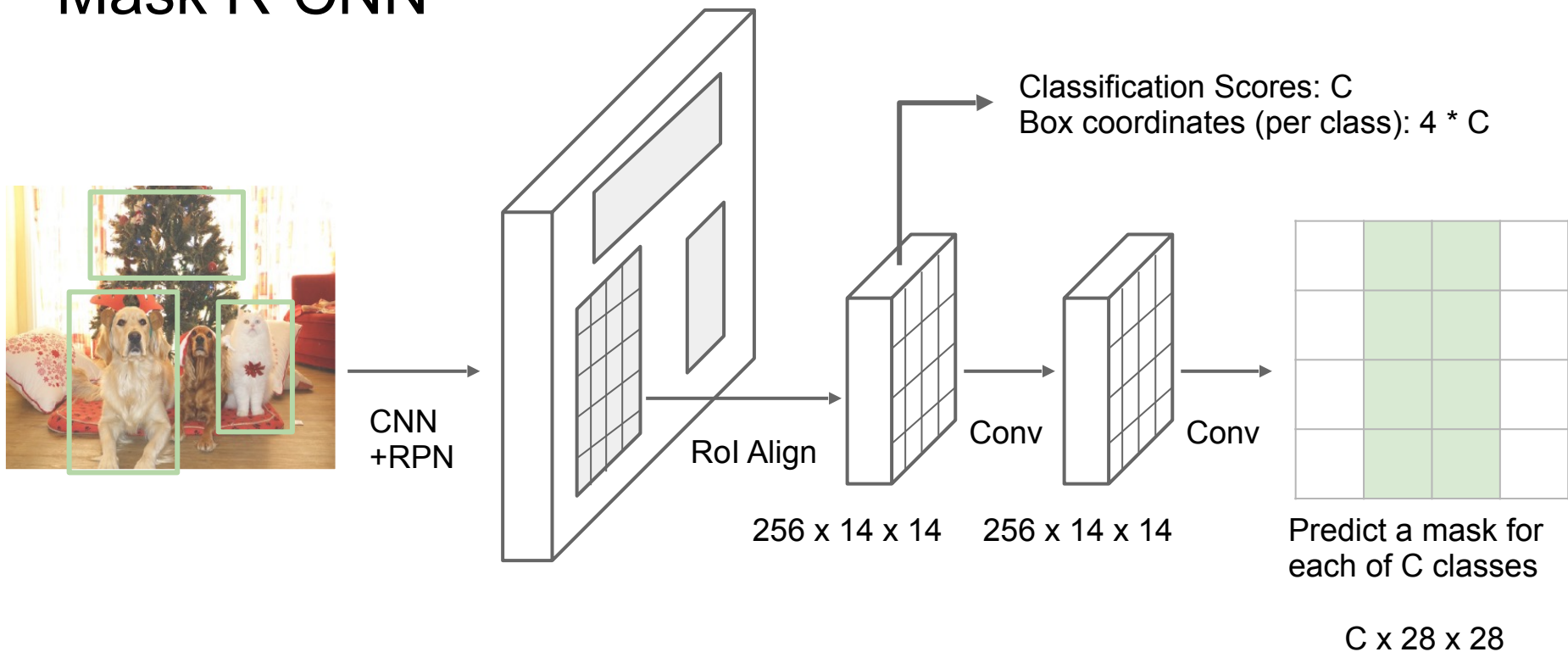


Instance Segmentation: Mask R-CNN



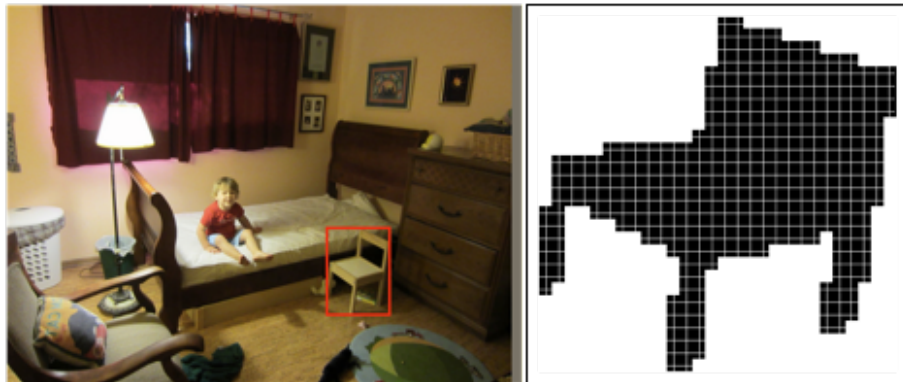
He et al, "Mask R-CNN", ICCV 2017

Mask R-CNN

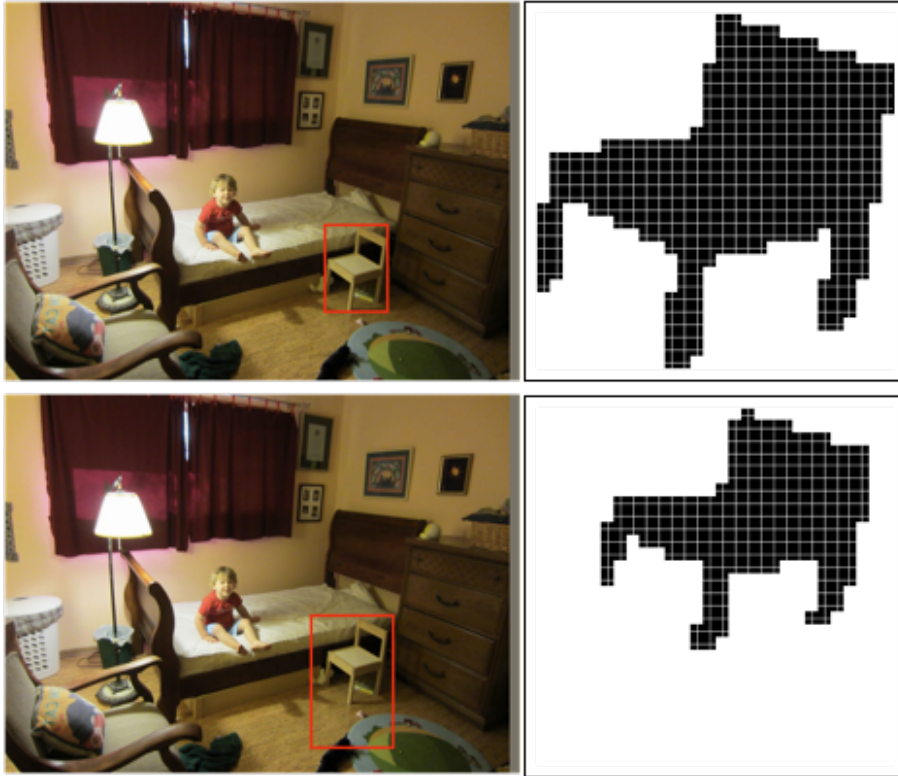


He et al, "Mask R-CNN", arXiv 2017

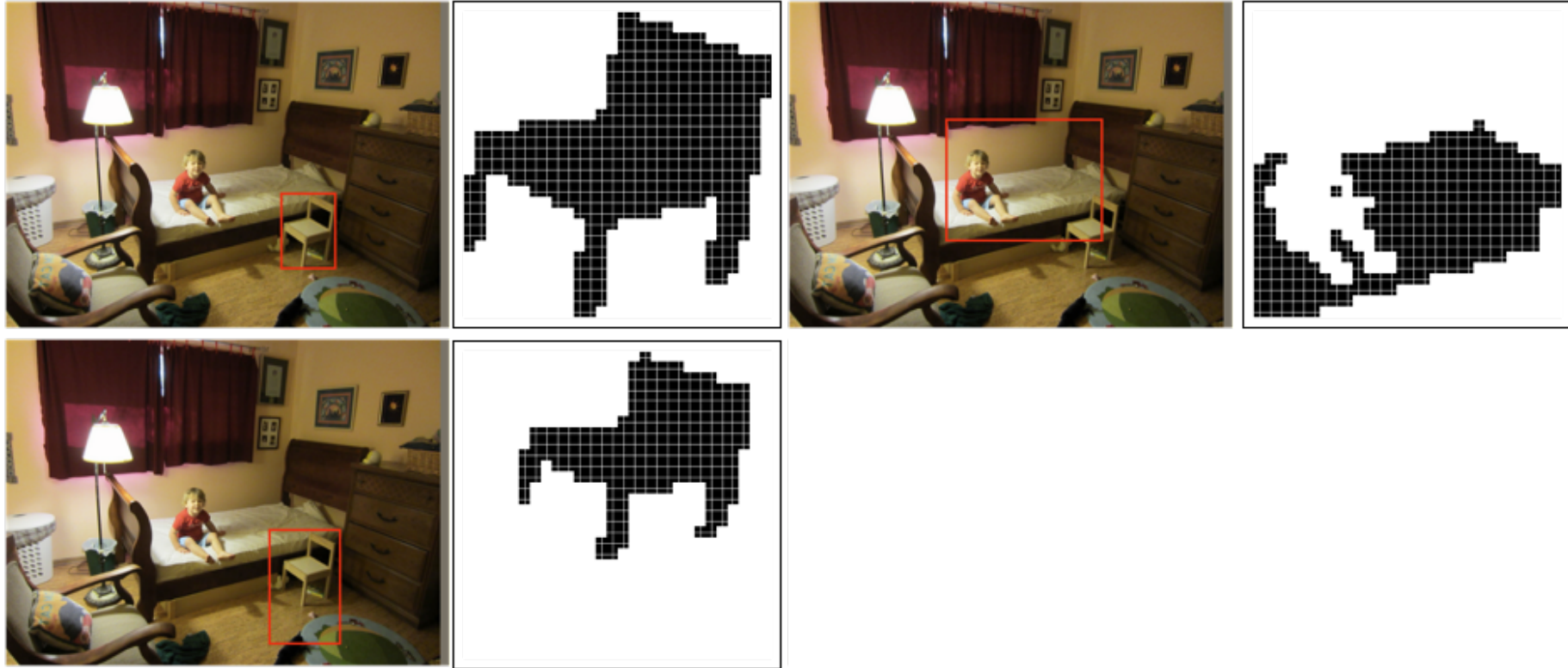
Mask R-CNN: Example Mask Training Targets



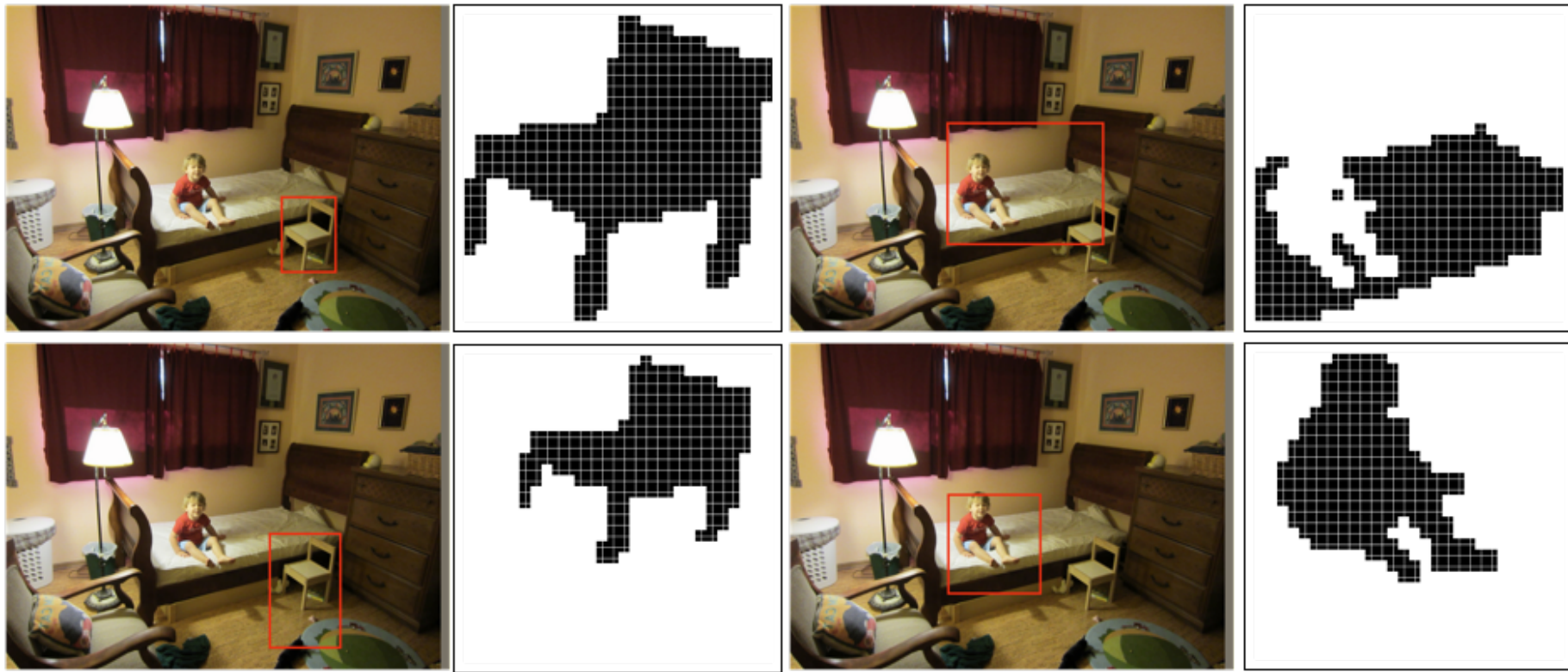
Mask R-CNN: Example Mask Training Targets



Mask R-CNN: Example Mask Training Targets

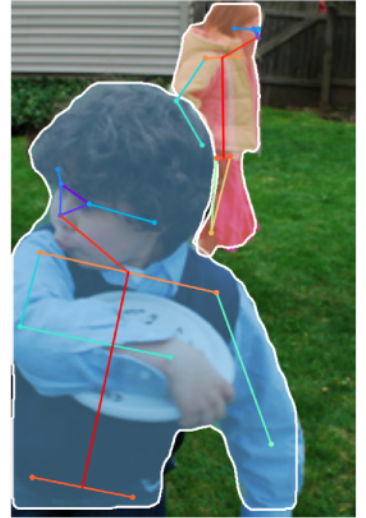
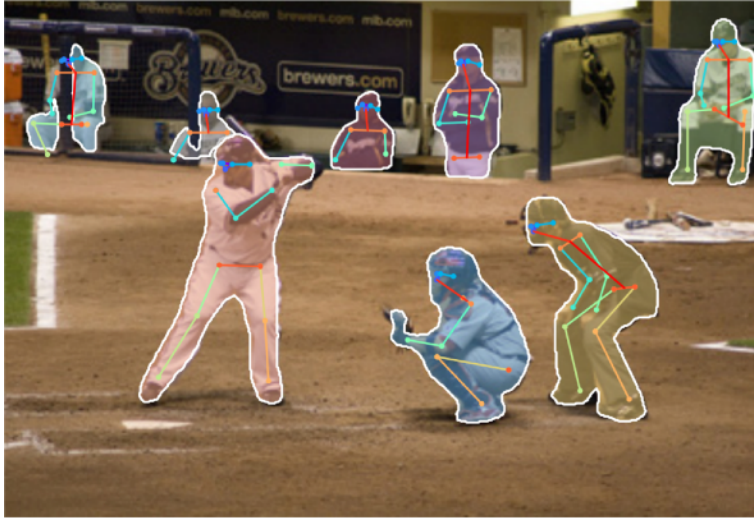


Mask R-CNN: Example Mask Training Targets



Mask R-CNN

Also does pose



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

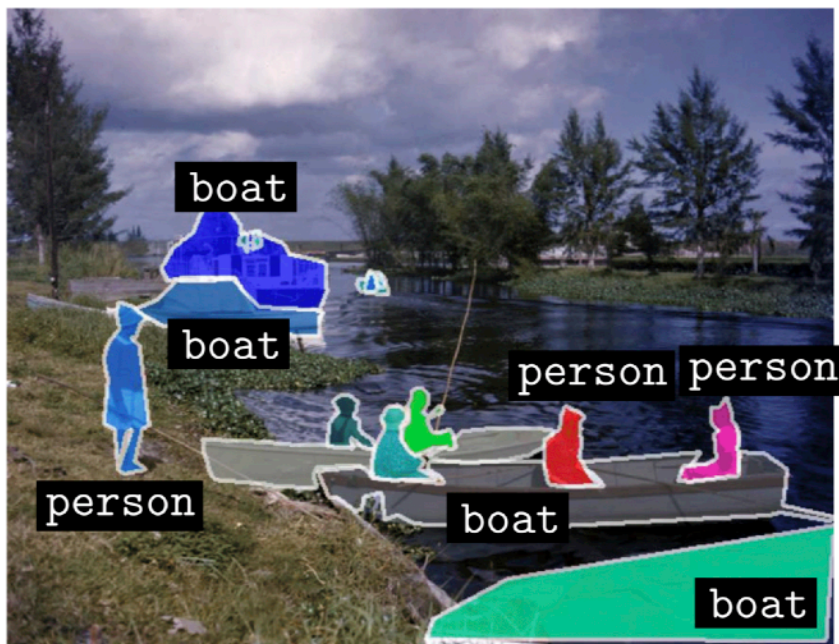
Subhransu Maji

Slide credits: Erik Learned-Miller, Fei-Fei Li, Jiajun Wu, Ruohan Gao

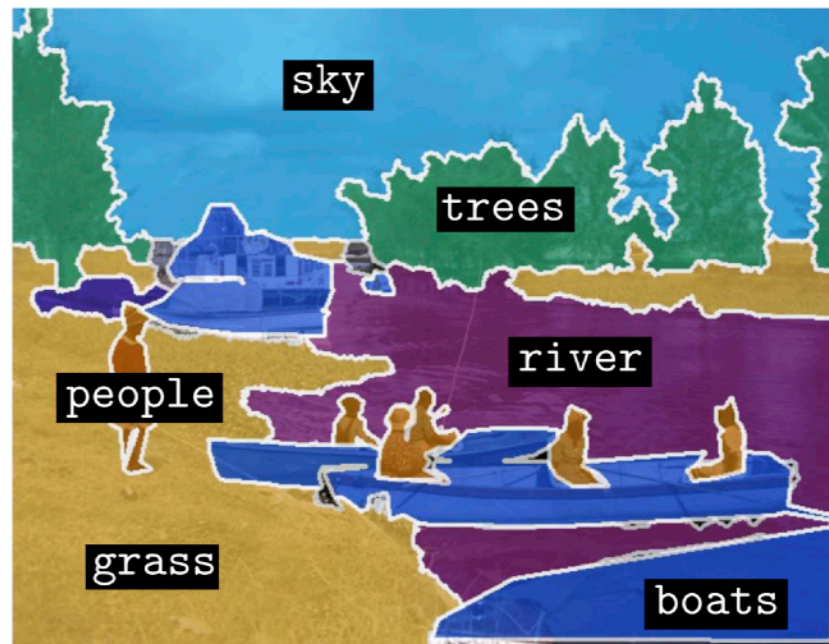
Lecture 13 - 45

March 26, 2026

Instance and Semantic Segmentation



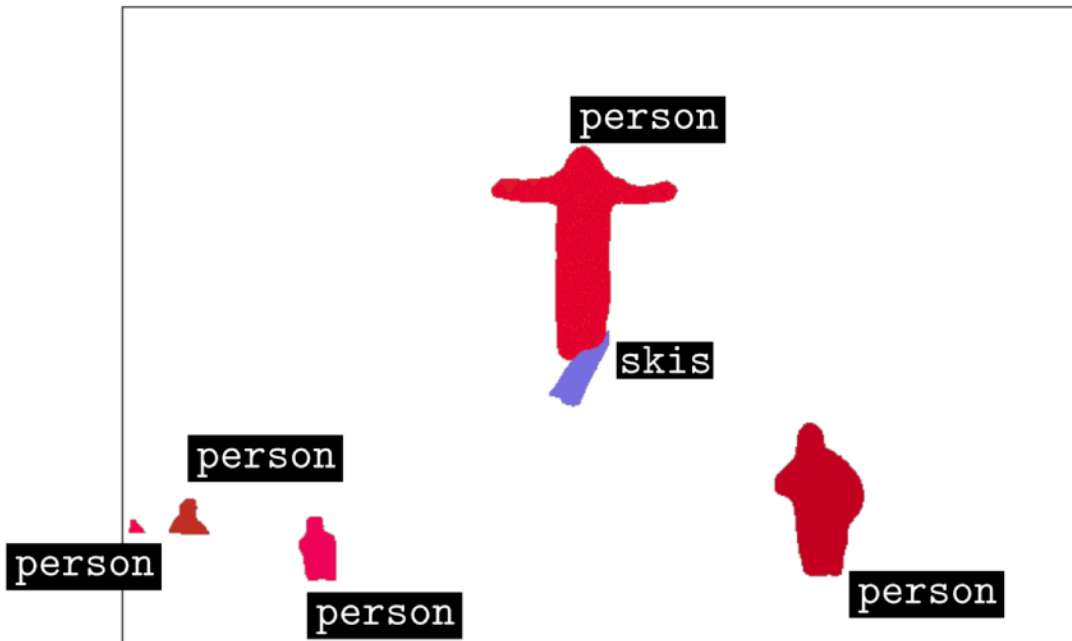
instance segmentation



semantic segmentation

real-world application likely requires both modalities

What do instance segmentation models see?



no understanding of the
general scene layout

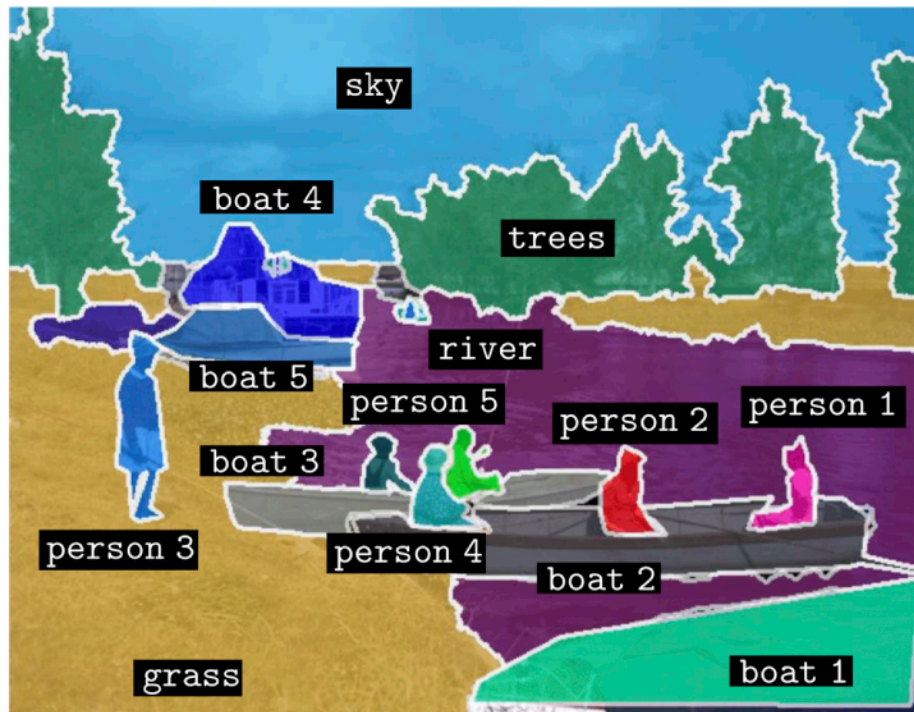
Figure Credit: Alexander Kirillov

What do semantic segmentation models see?



Does not differentiate
different instances

Panoptic Segmentation: Unified Segmentation



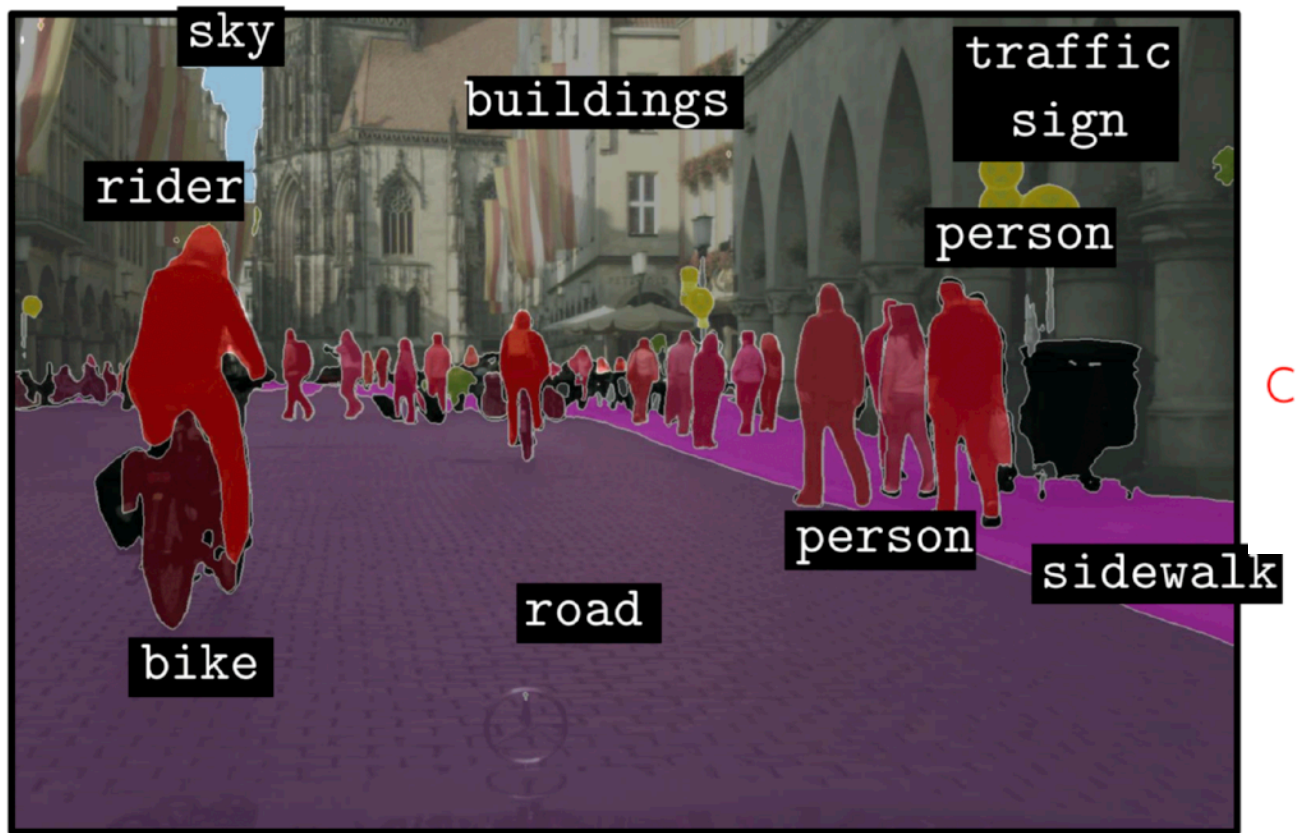
single task that combines semantic
and instance segmentation

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

Figure Credit: Alexander Kirillov

Panoptic: see everything at once

Panoptic Segmentation



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

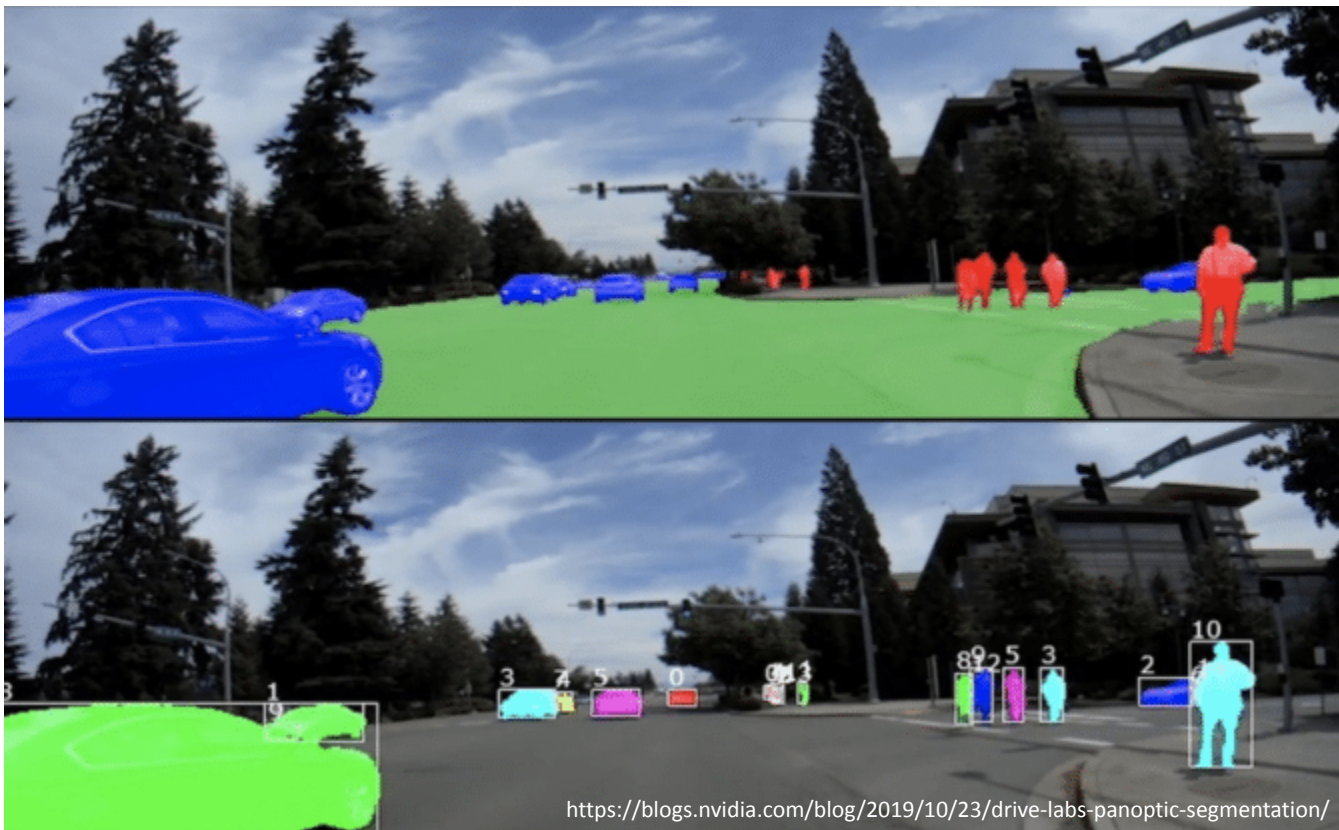


Cityscapes (2015)
panoptic test set
leaderboard (2019)

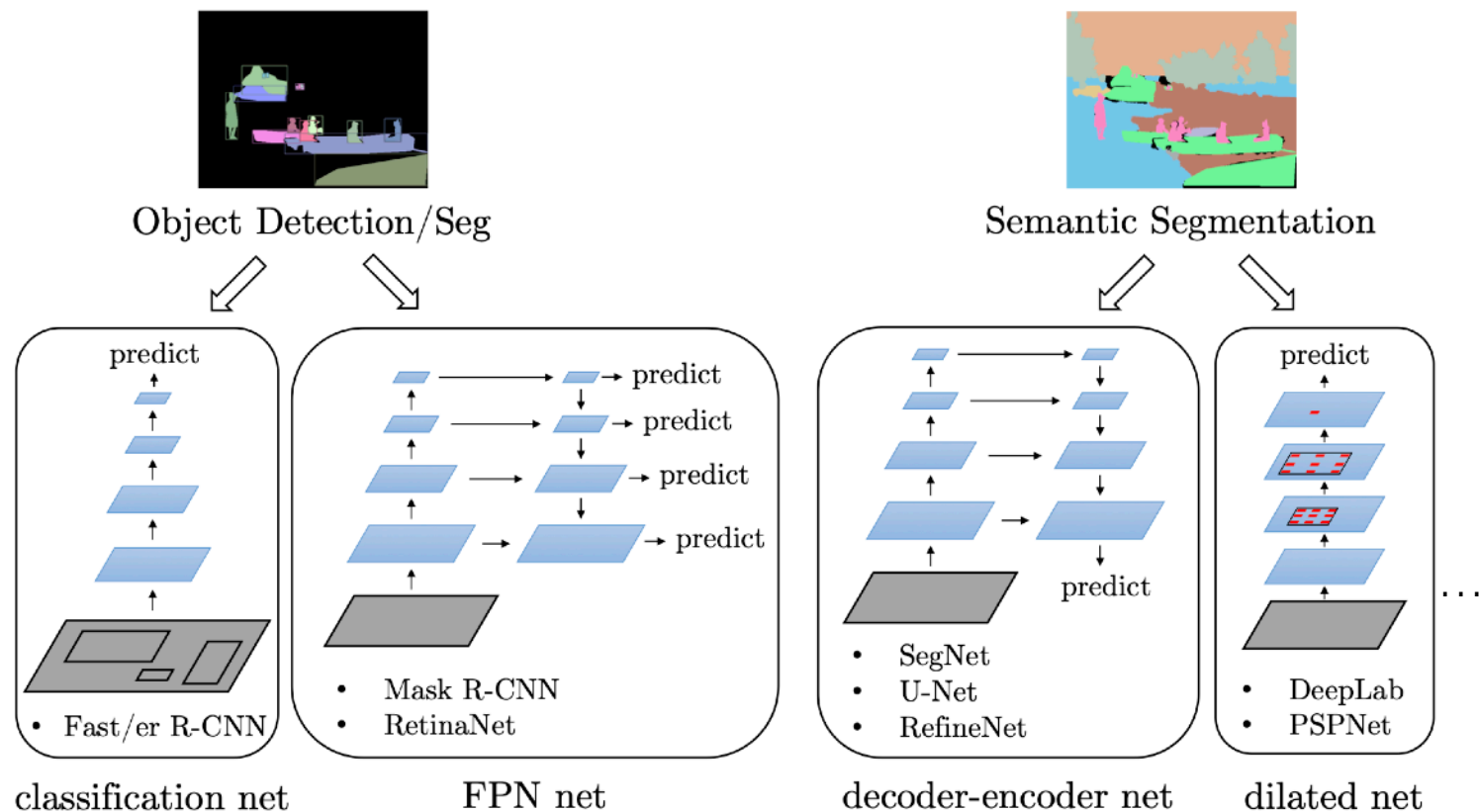


ADE20k (2016)
>22k images, 150 categories

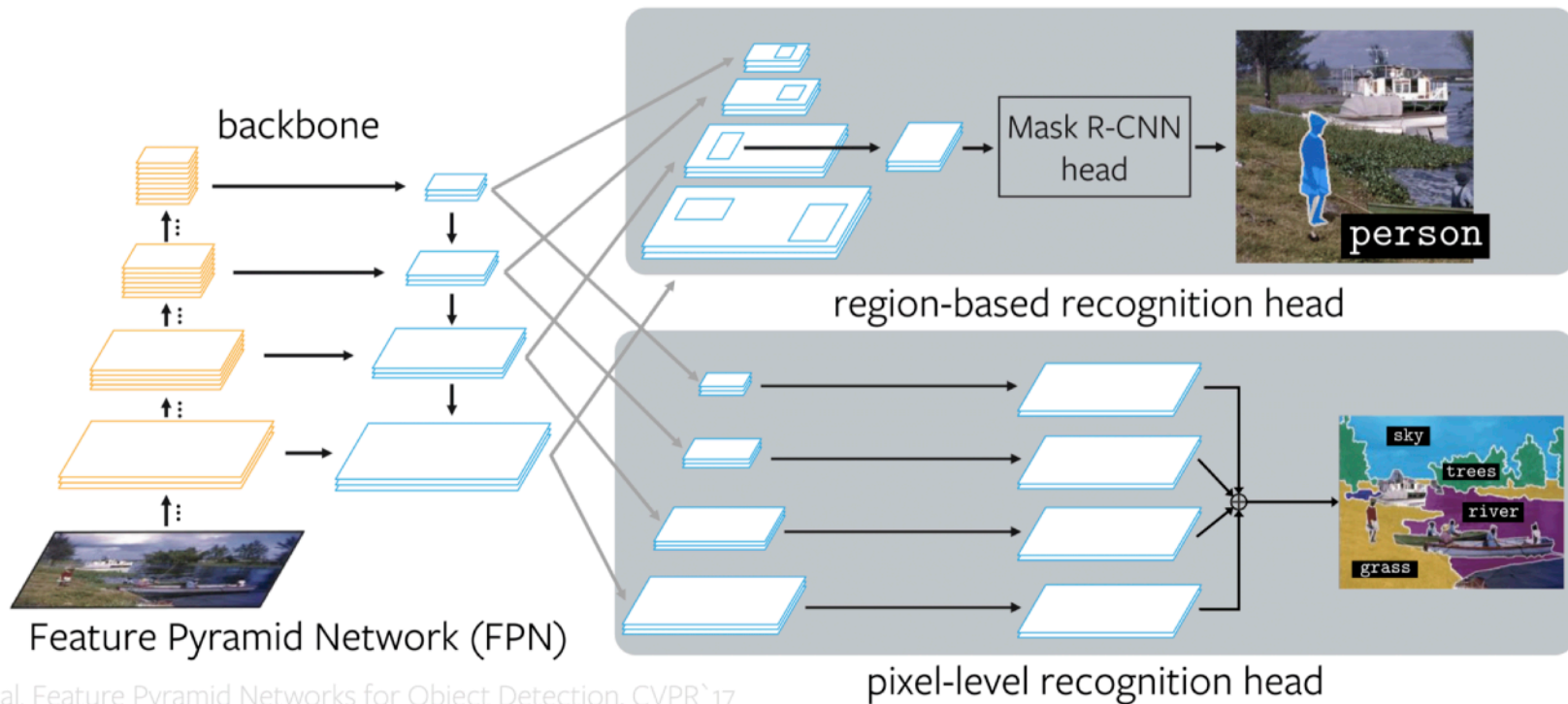
Panoptic Segmentation for Autonomous Driving



Deep Networks for Segmentation Tasks



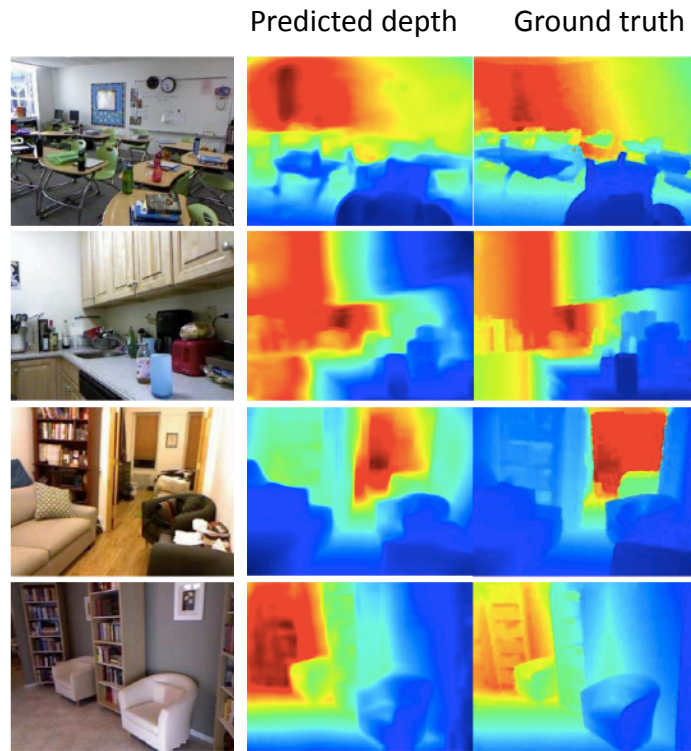
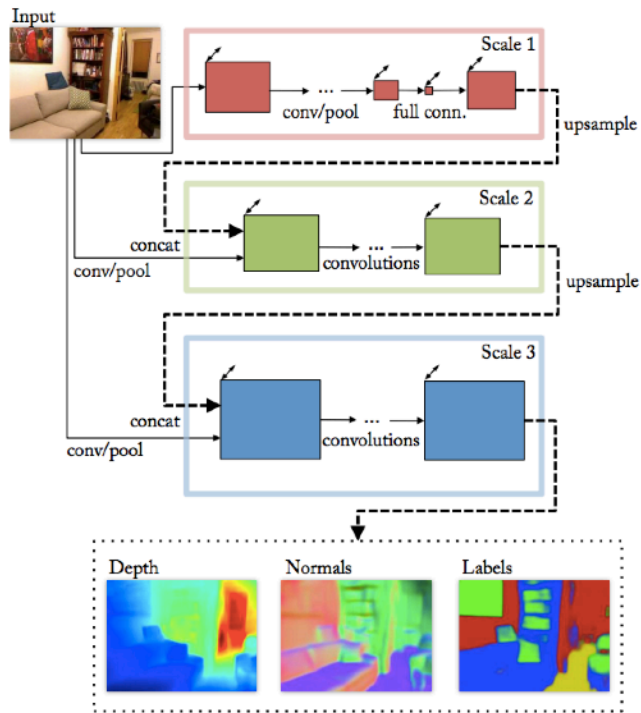
Panoptic FPN



et al. Feature Pyramid Networks for Object Detection, CVPR'17

Figure Credit: Alexander Kirillov

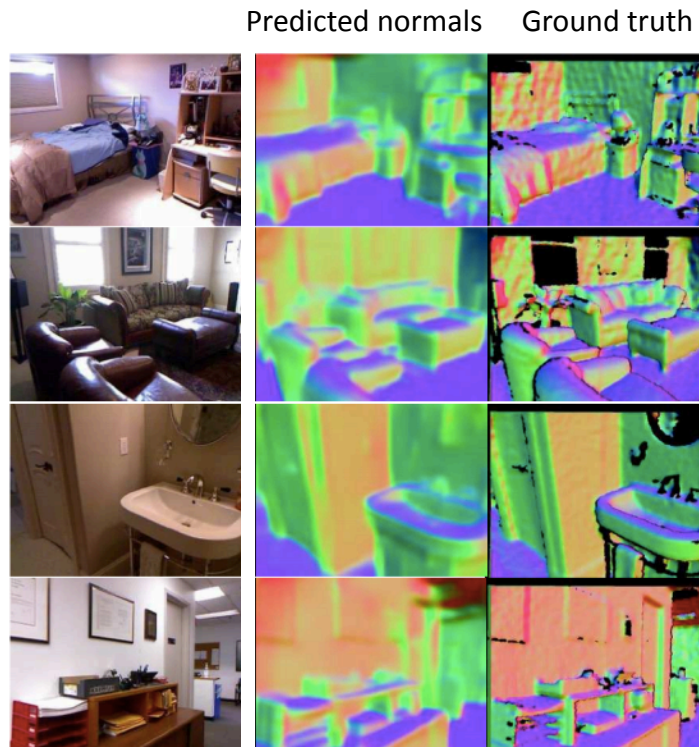
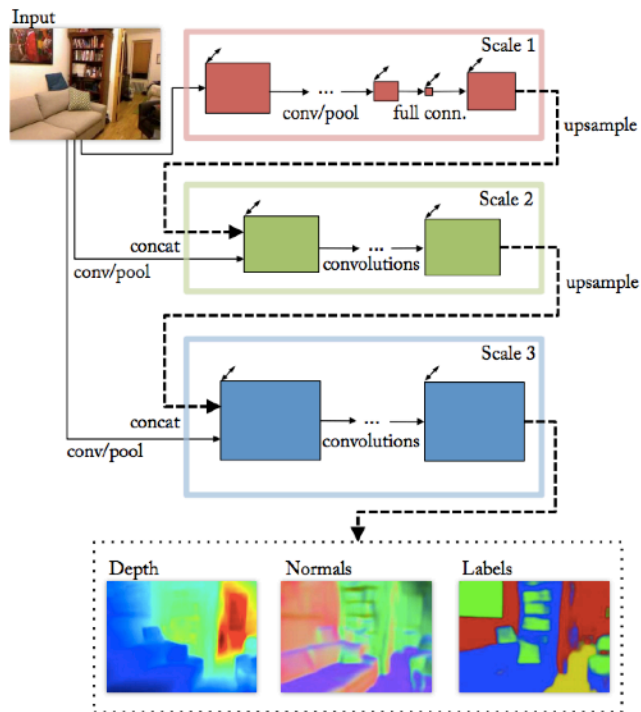
Dense Prediction: Depth and normal estimation



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

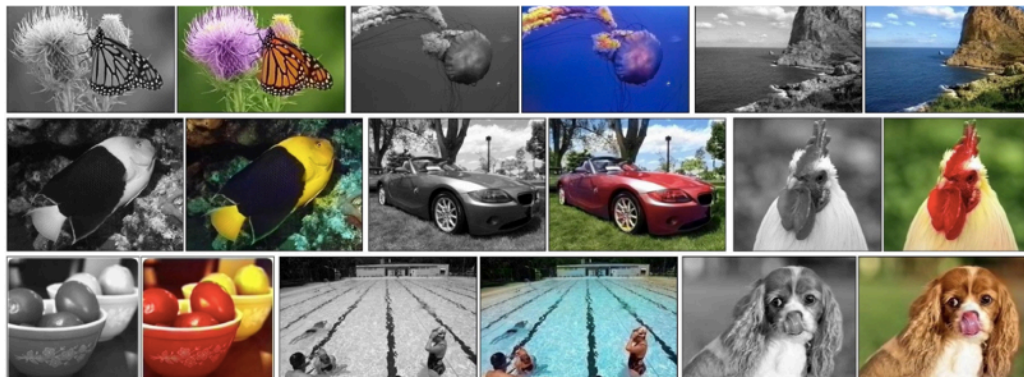
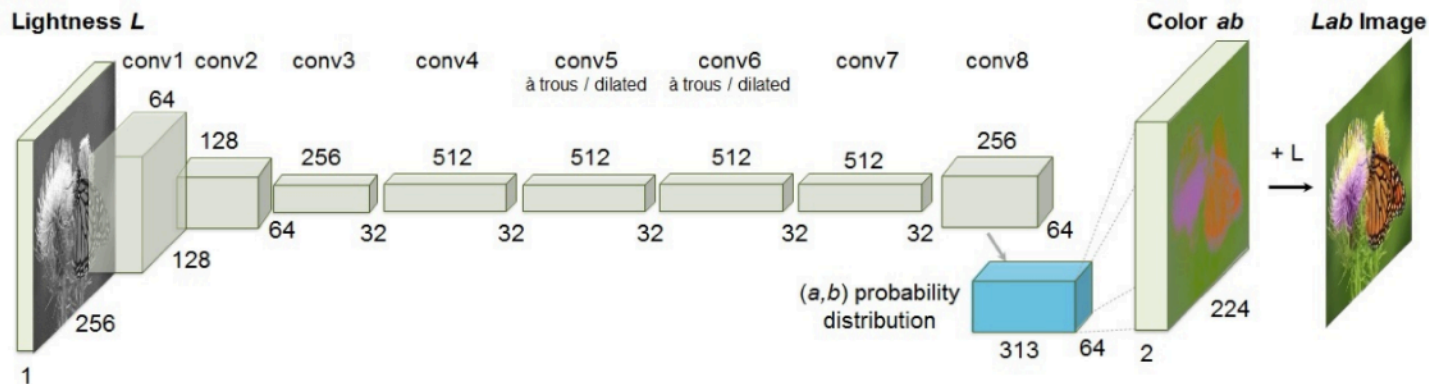
Dense Prediction: Depth and normal estimation



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



Slide credit: S. Lazebnik

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

Open Source Frameworks

Lots of good implementations on GitHub!

TensorFlow Detection API: https://github.com/tensorflow/models/tree/master/research/object_detection

Faster RCNN, SSD, RFCN, Mask R-CNN, ...

Detectron2 (PyTorch)

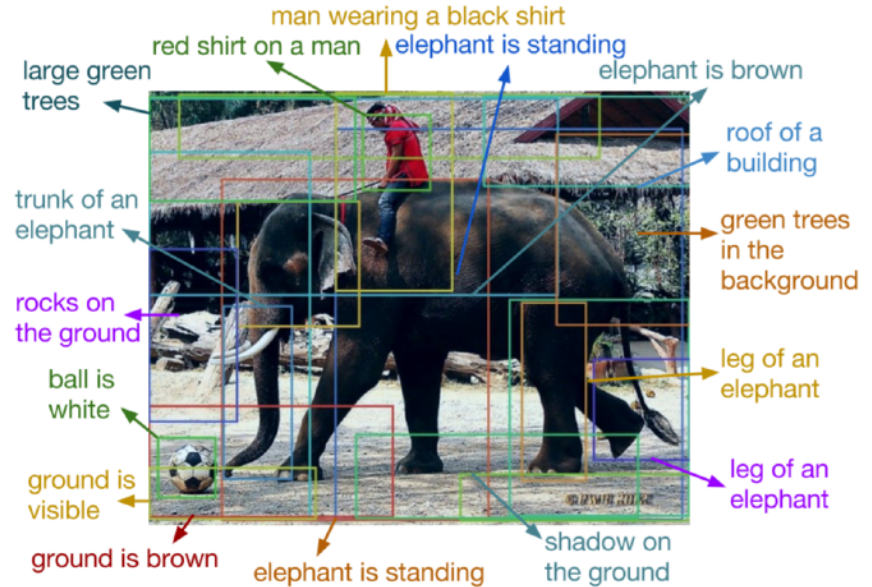
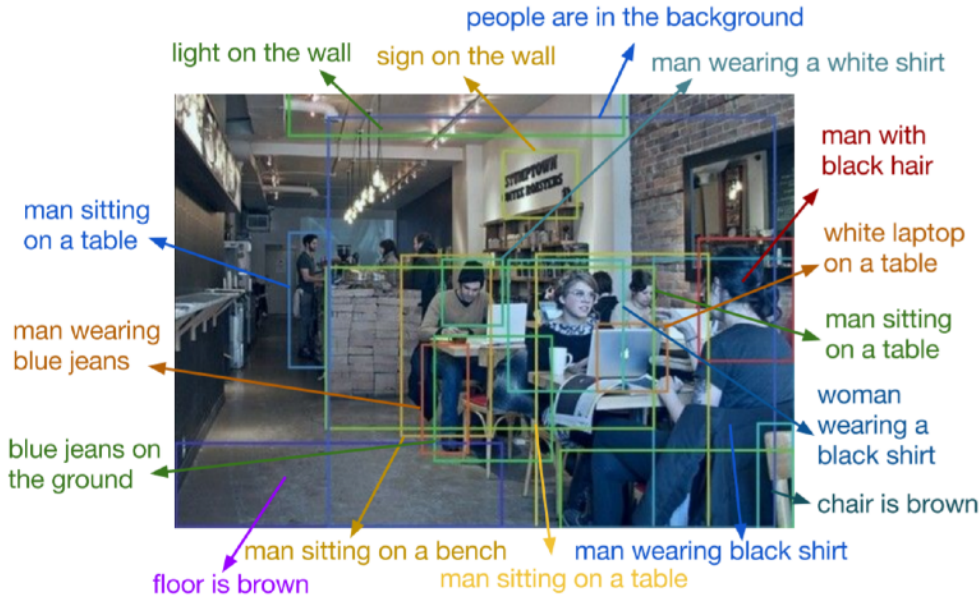
<https://github.com/facebookresearch/detectron2>

Mask R-CNN, RetinaNet, Faster R-CNN, RPN, Fast R-CNN, R-FCN, ...

Finetune on your own dataset with pre-trained models

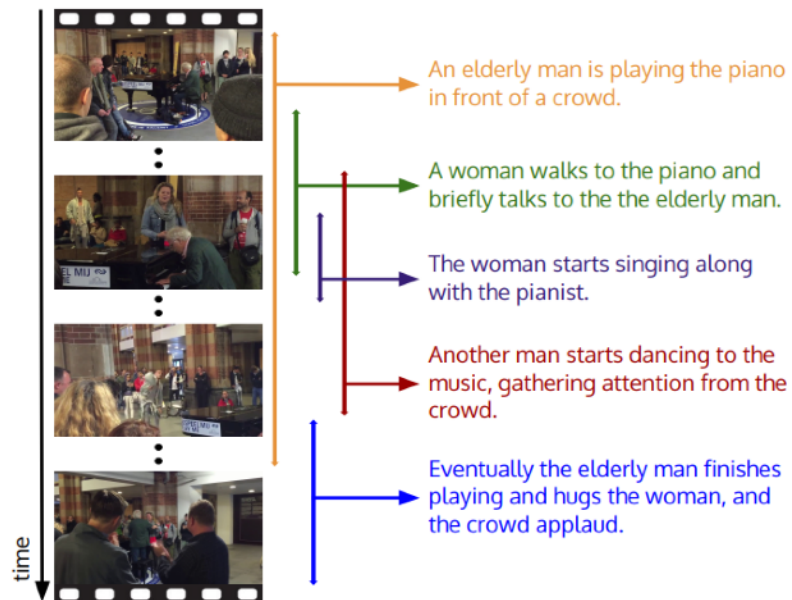
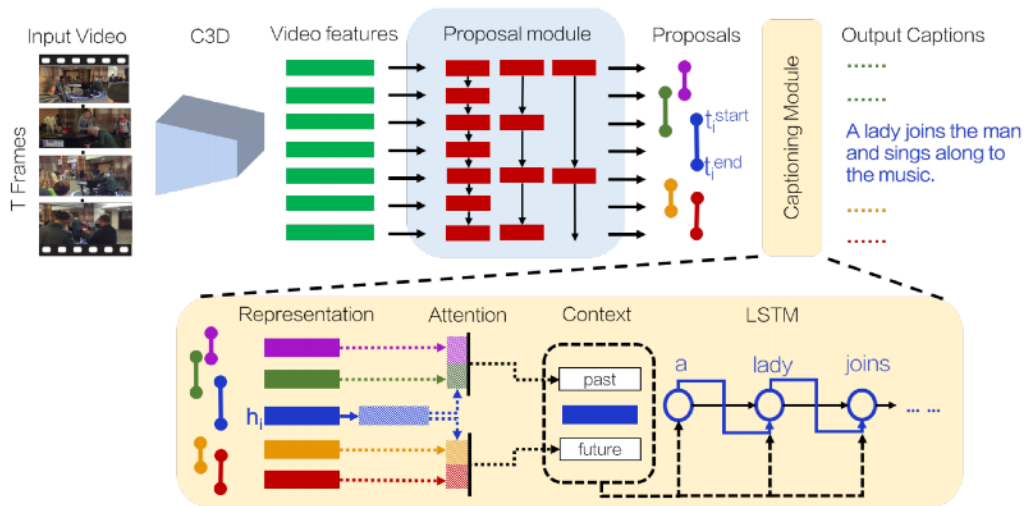
Beyond 2D Object Detection...

Object Detection + Captioning = Dense Captioning



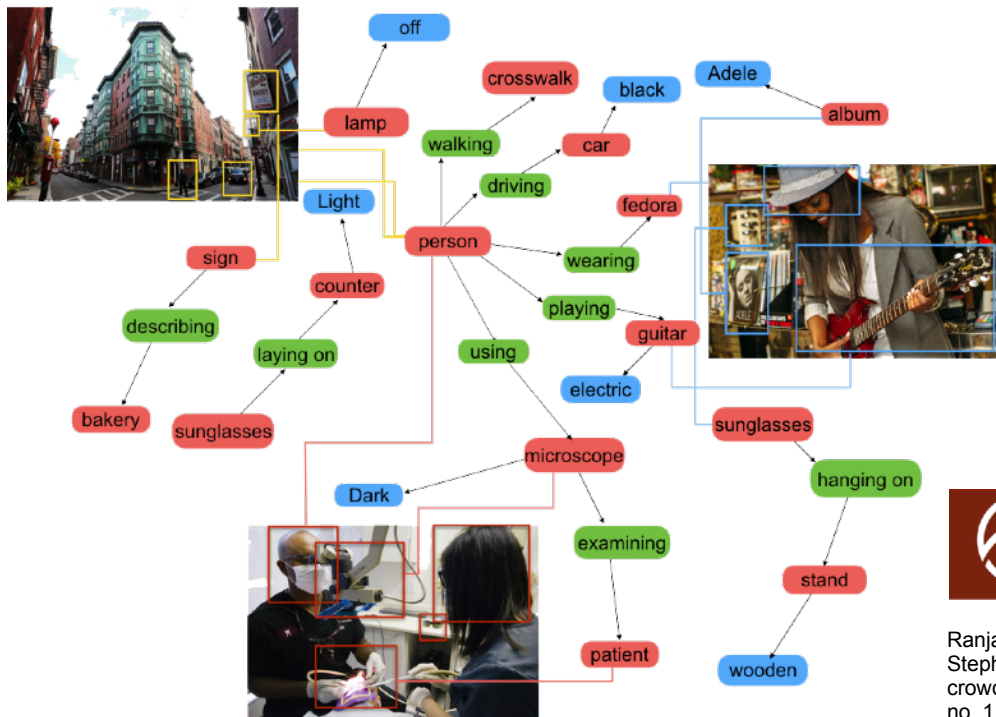
Johnson, Karpathy, and Fei-Fei, "DenseCap: Fully Convolutional Localization Networks for Dense Captioning", CVPR 2016
Figure copyright IEEE, 2016. Reproduced for educational purposes.

Dense Video Captioning



Ranjay Krishna et al., "Dense-Captioning Events in Videos", ICCV 2017
 Figure copyright IEEE, 2017. Reproduced with permission.

Objects + Relationships = Scene Graphs

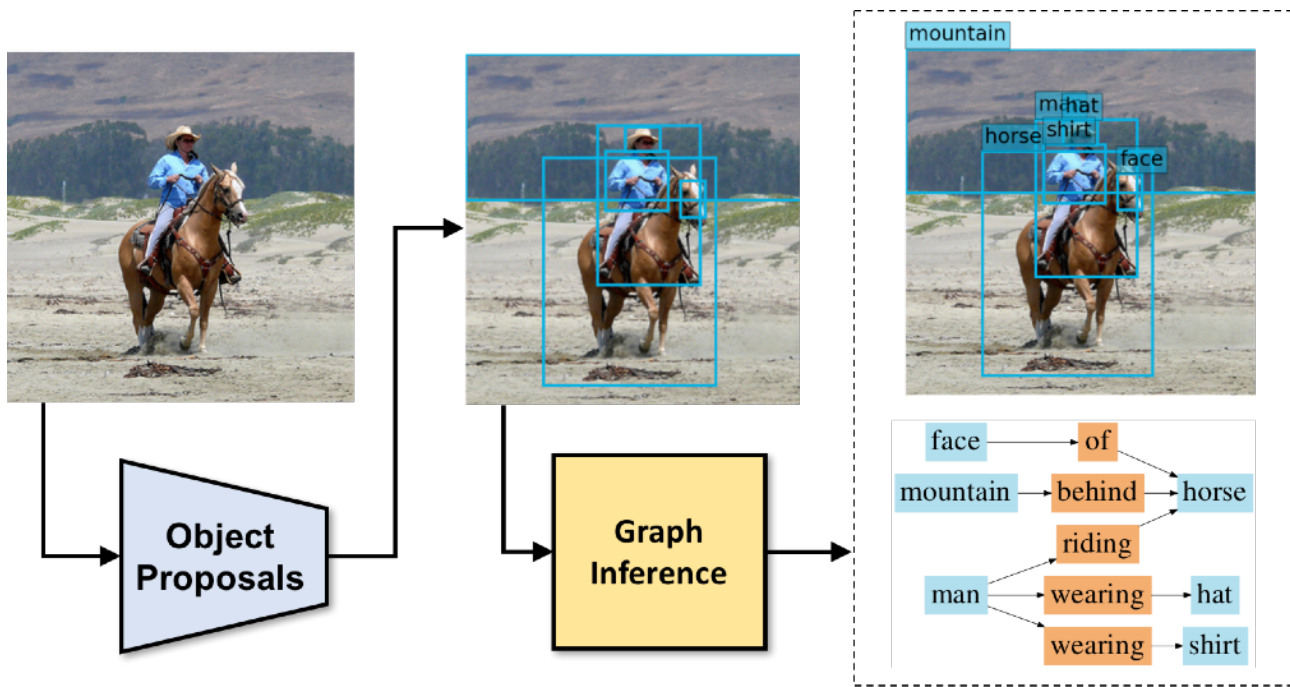


108,077 Images
5.4 Million Region Descriptions
1.7 Million Visual Question Answers
3.8 Million Object Instances
2.8 Million Attributes
2.3 Million Relationships
Everything Mapped to Wordnet Synsets



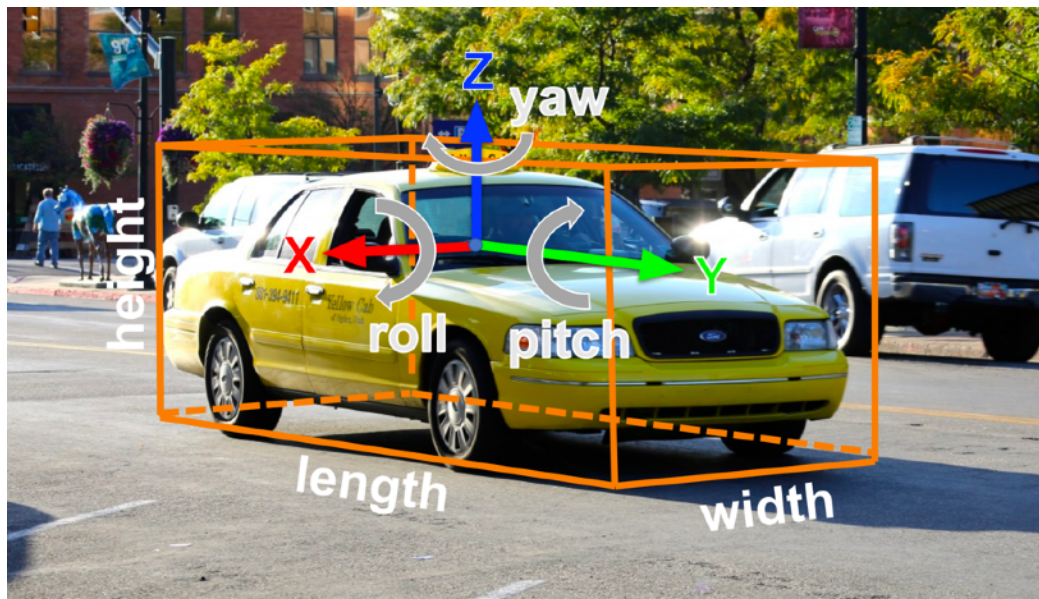
Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen et al. "Visual genome: Connecting language and vision using crowdsourced dense image annotations." International Journal of Computer Vision 123, no. 1 (2017): 32-73.

Scene Graph Prediction



Xu, Zhu, Choy, and Fei-Fei, "Scene Graph Generation by Iterative Message Passing", CVPR 2017
Figure copyright IEEE, 2018. Reproduced for educational purposes.

3D Object Detection



2D Object Detection:

2D bounding box

(x, y, w, h)

3D Object Detection:

3D oriented bounding box

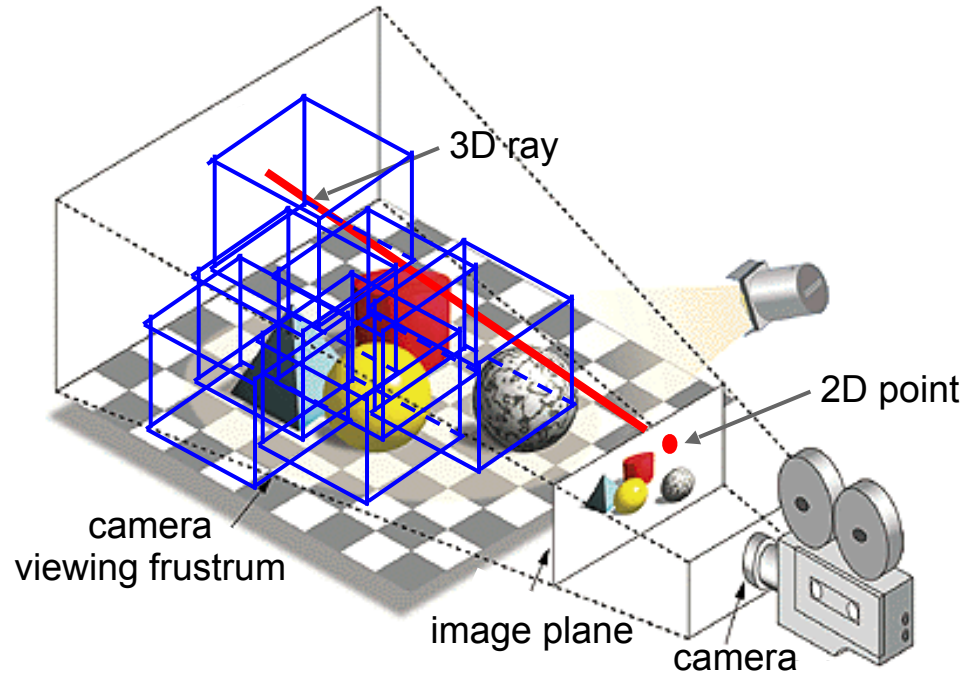
$(x, y, z, w, h, l, r, p, y)$

Simplified bbox: no roll & pitch

Much harder problem than 2D
object detection!

[This image](#) is [CC0 public domain](#)

3D Object Detection: Simple Camera Model



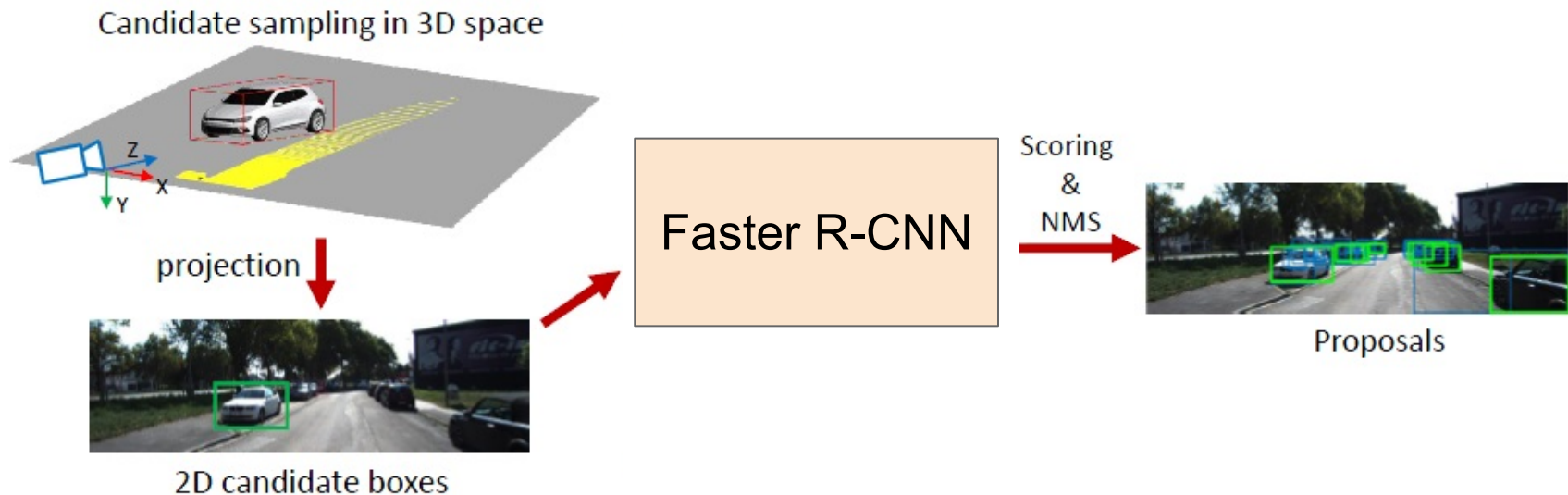
A point on the image plane corresponds to a **ray** in the 3D space

A 2D bounding box on an image is a **frustum** in the 3D space

Localize an object in 3D:
The object can be anywhere in the **camera viewing frustum!**

Image source: https://www.pcmag.com/encyclopedia_images/FRUSTUM.GIF

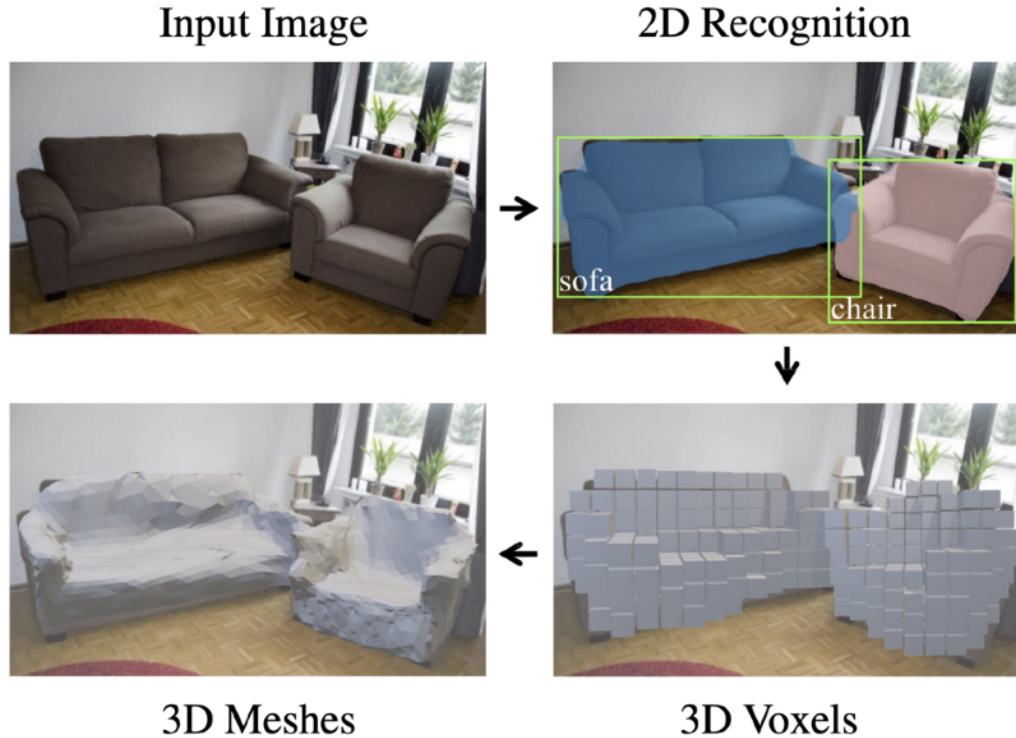
3D Object Detection: Monocular Camera



- Same idea as Faster RCNN, but proposals are in 3D
- 3D bounding box proposal, regress 3D box parameters + class score

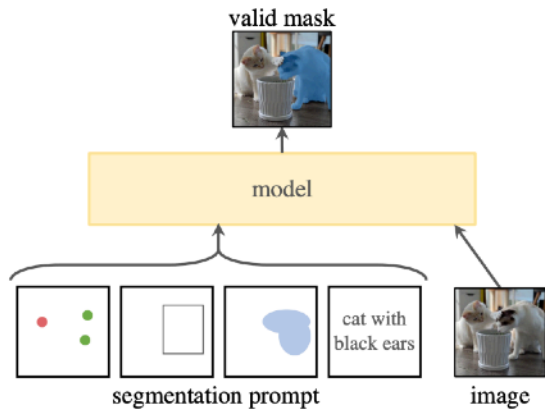
Chen, Xiaozhi, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. "Monocular 3d object detection for autonomous driving." CVPR 2016.

3D Shape Prediction: Mesh R-CNN

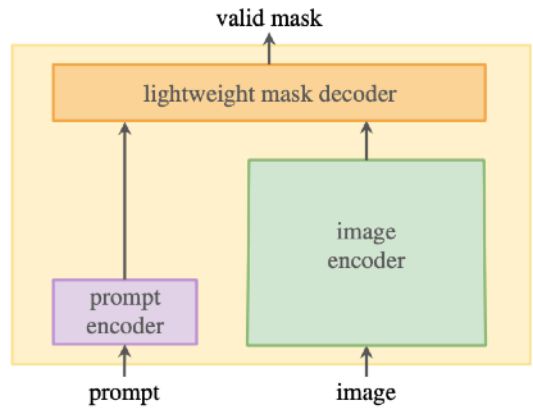


Gkioxari et al., Mesh RCNN, ICCV 2019

Segment Anything



(a) **Task:** promptable segmentation



(b) **Model:** Segment Anything Model (SAM)

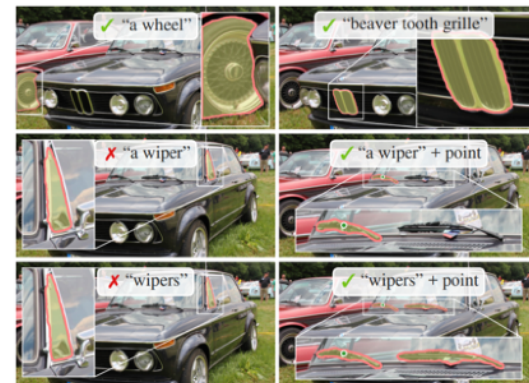


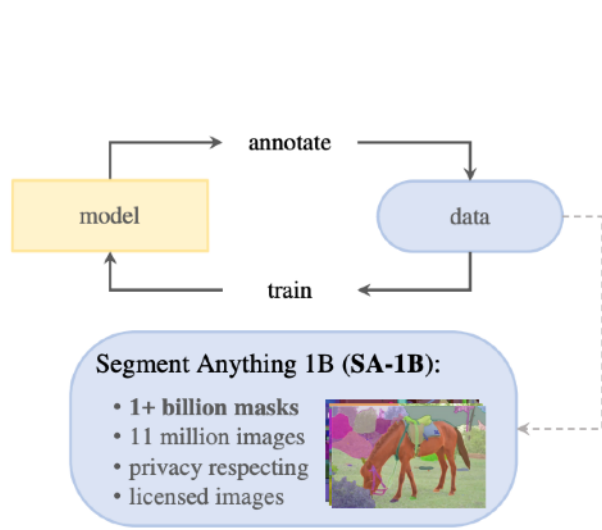
Figure 12: Zero-shot text-to-mask. SAM can work with simple and nuanced text prompts. When SAM fails to make a correct prediction, an additional point prompt can help.

Segment Anything

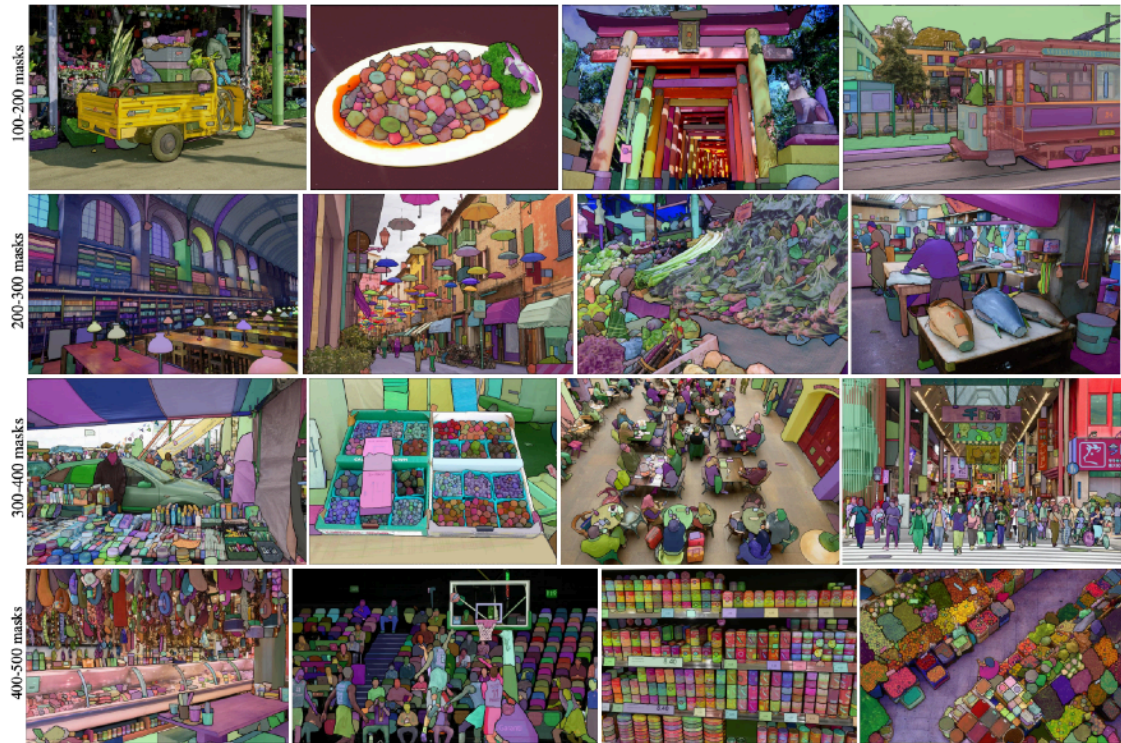
Alexander Kirillov^{1,2,4} Eric Mintun² Nikhila Ravi^{1,2} Hanzi Mao² Chloe Rolland³ Laura Gustafson³
Tete Xiao³ Spencer Whitehead Alexander C. Berg Wan-Yen Lo Piotr Dollár⁴ Ross Girshick⁴
¹project lead ²joint first author ³equal contribution ⁴directional lead

Meta AI Research, FAIR

Segment Anything — Training Set



(c) **Data:** data engine (top) & dataset (bottom)



Open Vocabulary Detection and Segmentation

Align image representations to language, clicks, etc.

Region CLIP, GLIP, Grounding Dino, Grounding SAM,

SAM, SAM2, OpenSeg

Recap: Lots of computer vision tasks!

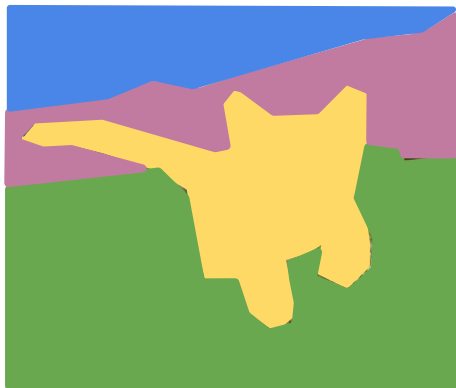
Classification



CAT

No spatial extent

Semantic Segmentation



**GRASS, CAT,
TREE, SKY**

No objects, just pixels

Object Detection



DOG, DOG, CAT

Multiple Object

Instance Segmentation



DOG, DOG, CAT

[This image](#) is [CC0 public domain](#)