

Introduction to Computer Vision Detection and segmentation

Dr. Sergio Escalera
University of Barcelona



UNIVERSITAT DE
BARCELONA

Computer Vision Tasks

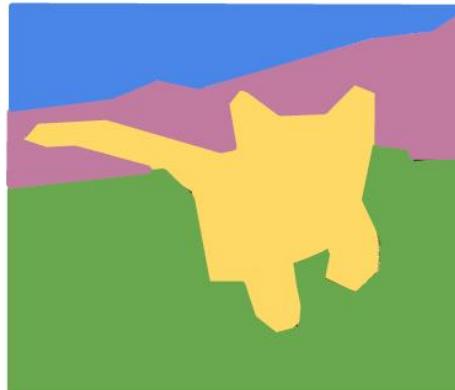
Classification



CAT

No spatial extent

Semantic Segmentation



GRASS, CAT,
TREE, SKY

No objects, just pixels

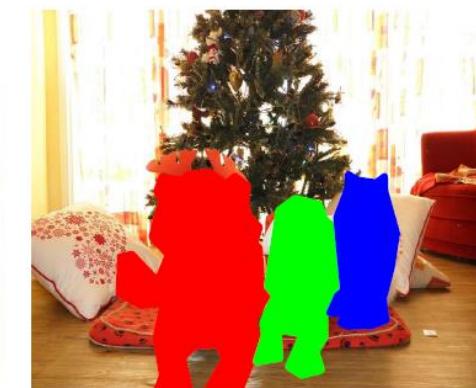
Object Detection



DOG, DOG, CAT

Multiple Object

Instance Segmentation



DOG, DOG, CAT

This image is CC0 public domain

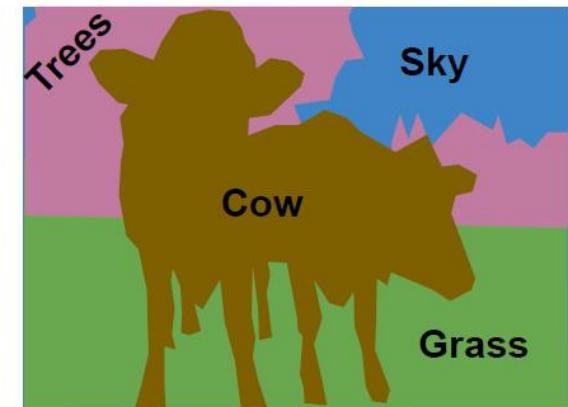
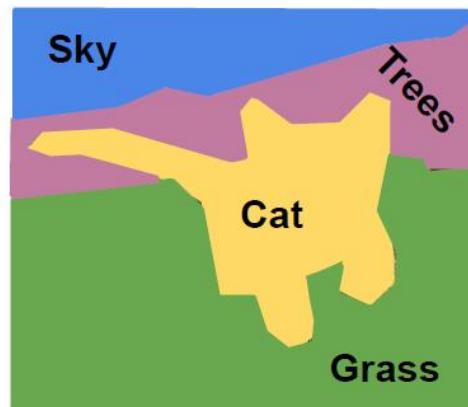
Semantic Segmentation

Label each pixel in the image with a category label

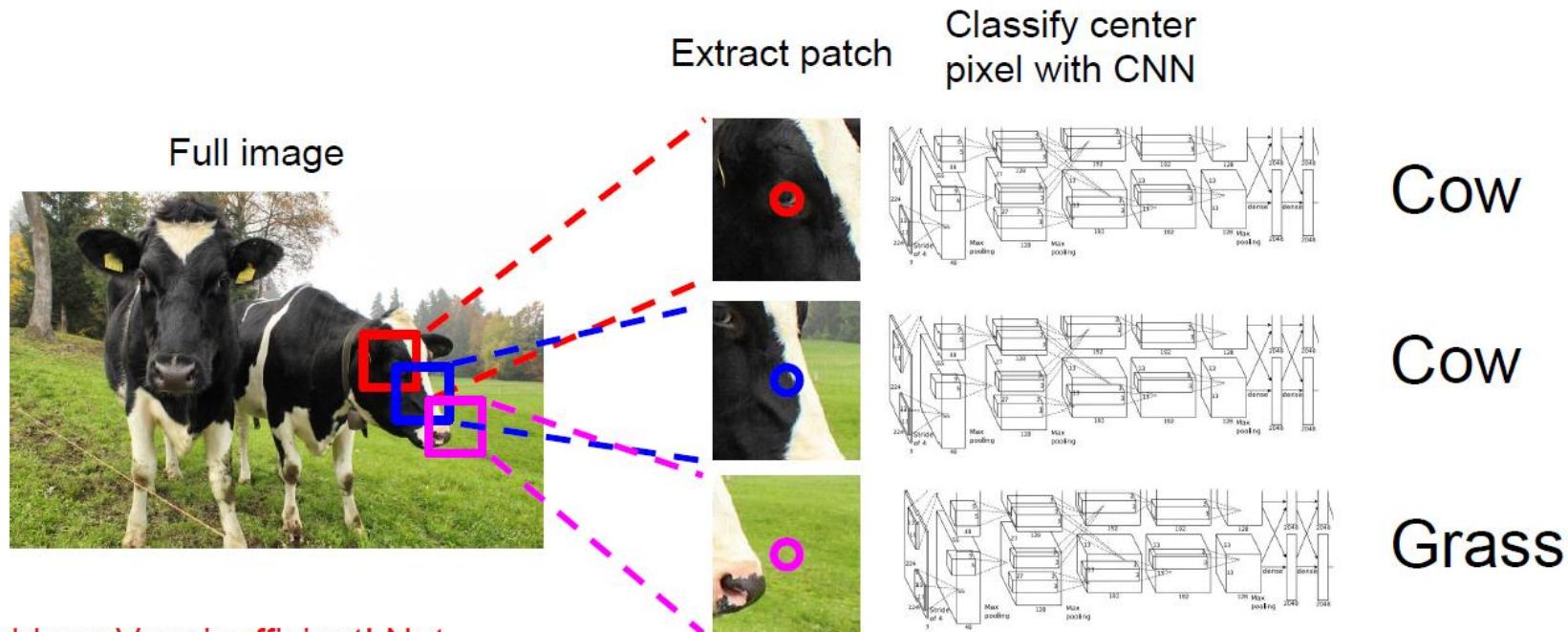
Don't differentiate instances, only care about pixels



This image is CC0 public domain



Semantic Segmentation Idea: Sliding Window

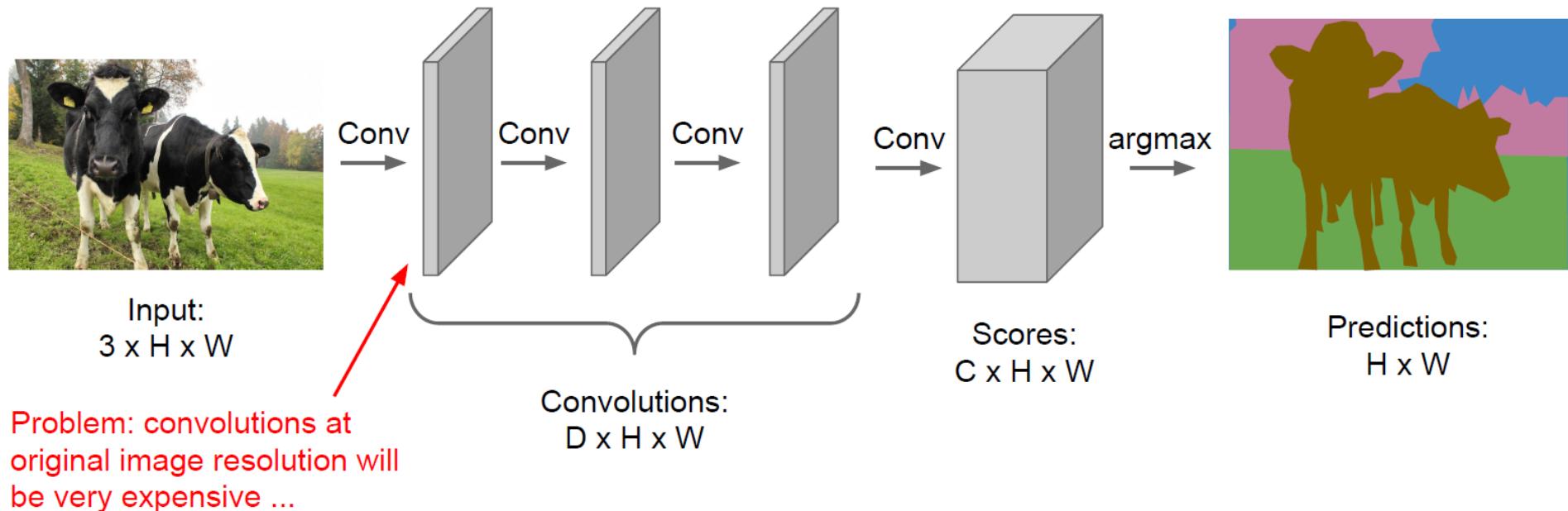


Problem: Very inefficient! Not reusing shared features between overlapping patches

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

Semantic Segmentation Idea: Fully Convolutional

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



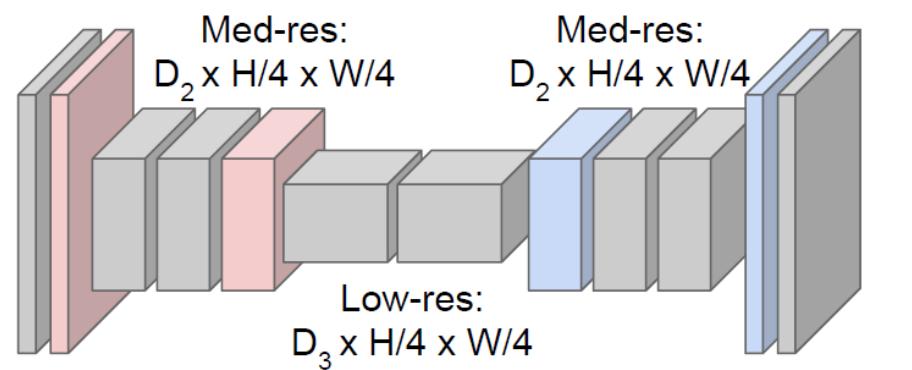
Semantic Segmentation Idea: Fully Convolutional

Downsampling:
Pooling, strided
convolution



Input:
 $3 \times H \times W$

High-res:
 $D_1 \times H/2 \times W/2$



Upsampling:
???



Predictions:
 $H \times W$

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”

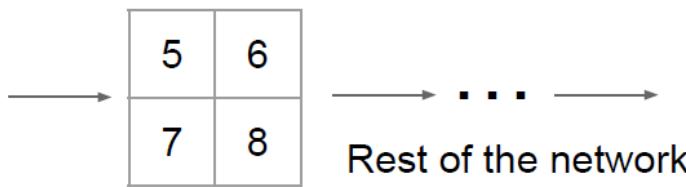
Max Pooling

Remember which element was max!

1	2	6	3
3	5	2	1
1	2	2	1
7	3	4	8

Input: 4 x 4

Output: 2 x 2



Max Unpooling

Use positions from pooling layer

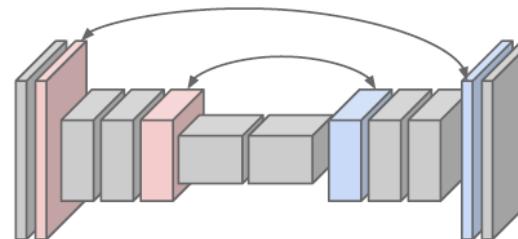
1	2
3	4

Input: 2 x 2

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

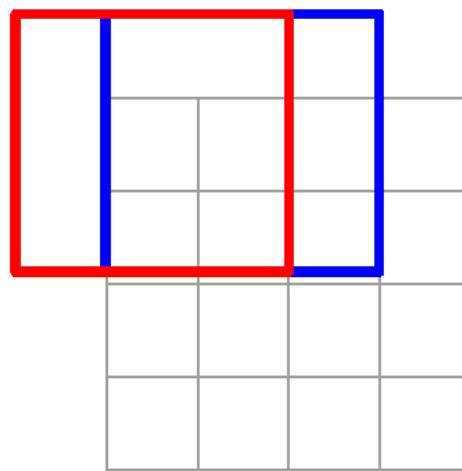
Output: 4 x 4

Corresponding pairs of
downsampling and
upsampling layers



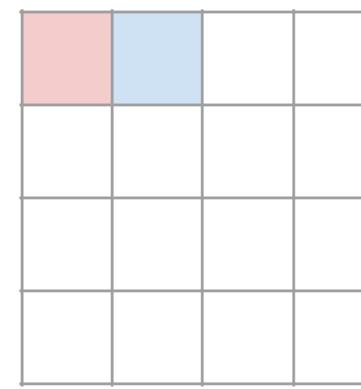
Learnable Upsampling: Transpose Convolution

Recall: Normal 3×3 convolution, stride 1 pad 1



Input: 4×4

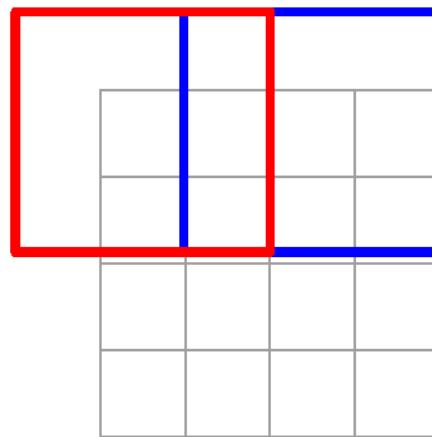
Dot product
between filter
and input



Output: 4×4

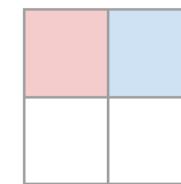
Learnable Upsampling: Transpose Convolution

Recall: Normal 3×3 convolution, stride 2 pad 1



Input: 4×4

Dot product
between filter
and input



Output: 2×2

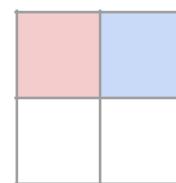
Filter moves 2 pixels in
the input for every one
pixel in the output

Stride gives ratio between
movement in input and
output

Learnable Upsampling: Transpose Convolution

Other names:

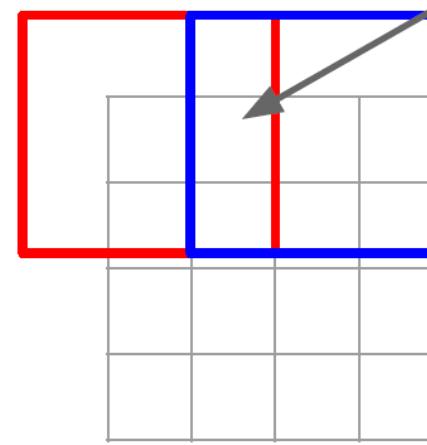
- Deconvolution (bad)
- Upconvolution
- Fractionally strided convolution
- Backward strided convolution



Input: 2 x 2

3 x 3 transpose convolution, stride 2 pad 1

Input gives weight for filter



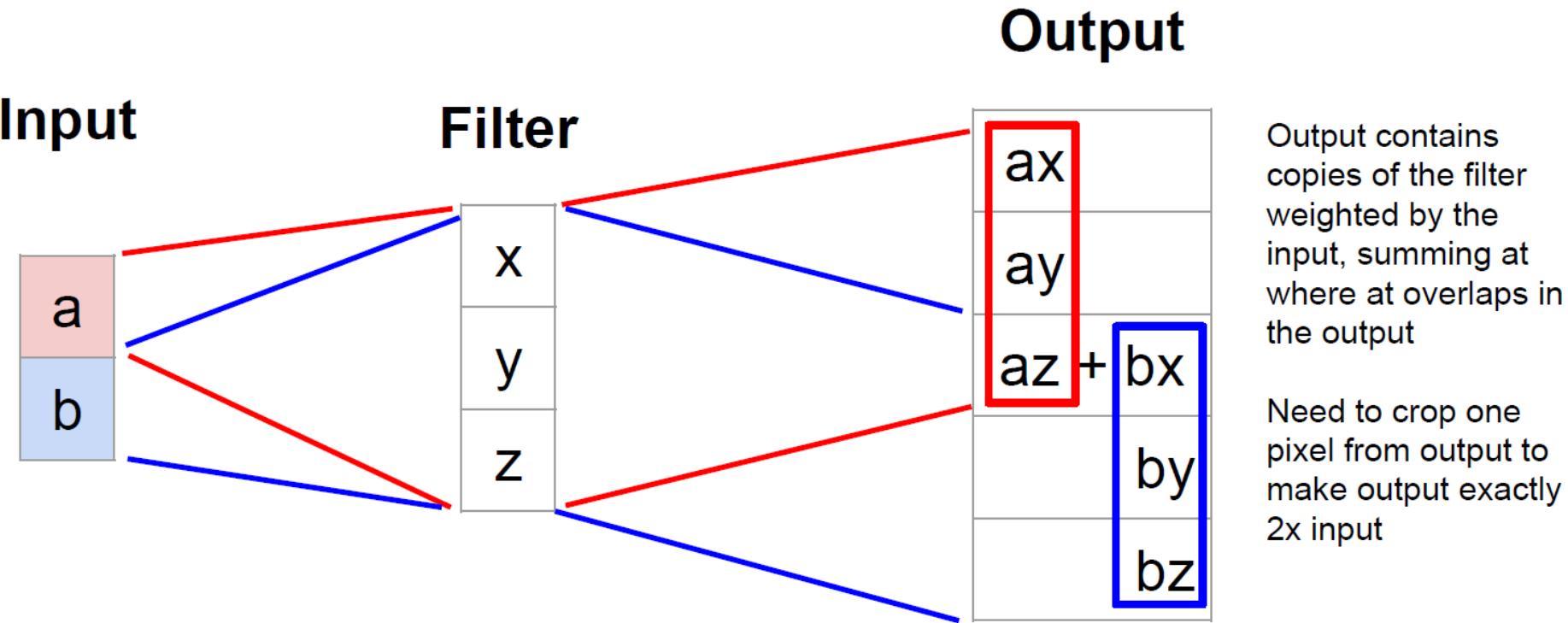
Output: 4 x 4

Sum where output overlaps

Filter moves 2 pixels in the output for every one pixel in the input

Stride gives ratio between movement in output and input

Learnable Upsampling: 1D Example



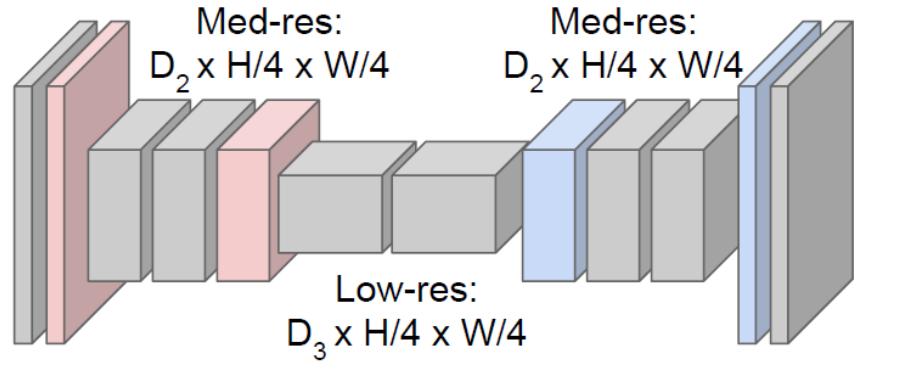
Semantic Segmentation Idea: Fully Convolutional

Downsampling:
Pooling, strided convolution



Input:
 $3 \times H \times W$

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



Upsampling:
Unpooling or strided transpose convolution



Predictions:
 $H \times W$

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015

Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Object Detection

Classification



CAT

No spatial extent

Semantic Segmentation



GRASS, CAT,
TREE, SKY

No objects, just pixels

Object
Detection



DOG, DOG, CAT

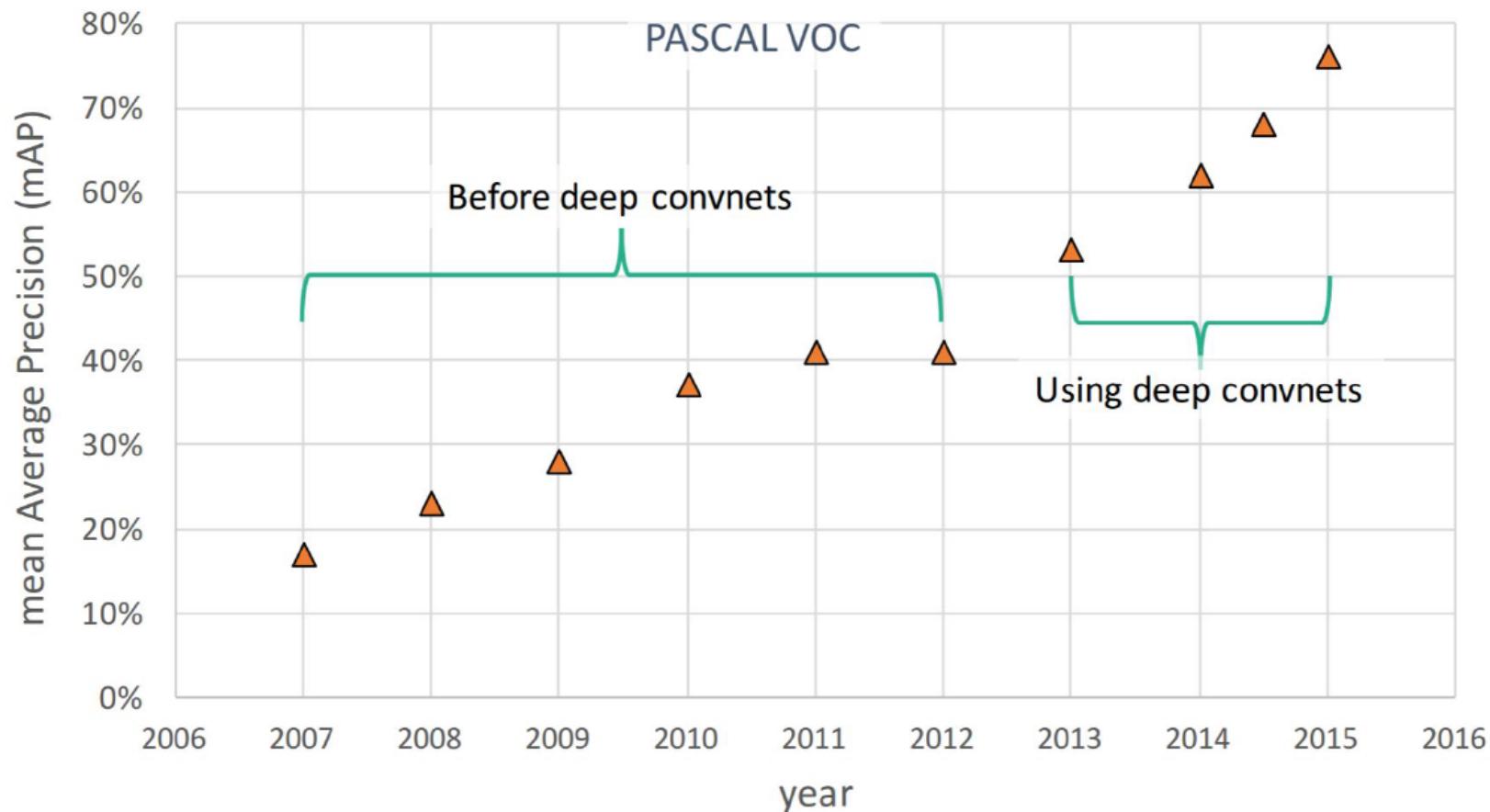
Multiple Object

Instance
Segmentation



DOG, DOG, CAT

Object Detection: Impact of Deep Learning

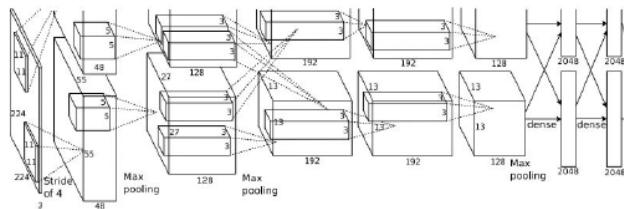


Object Detection: Single Object

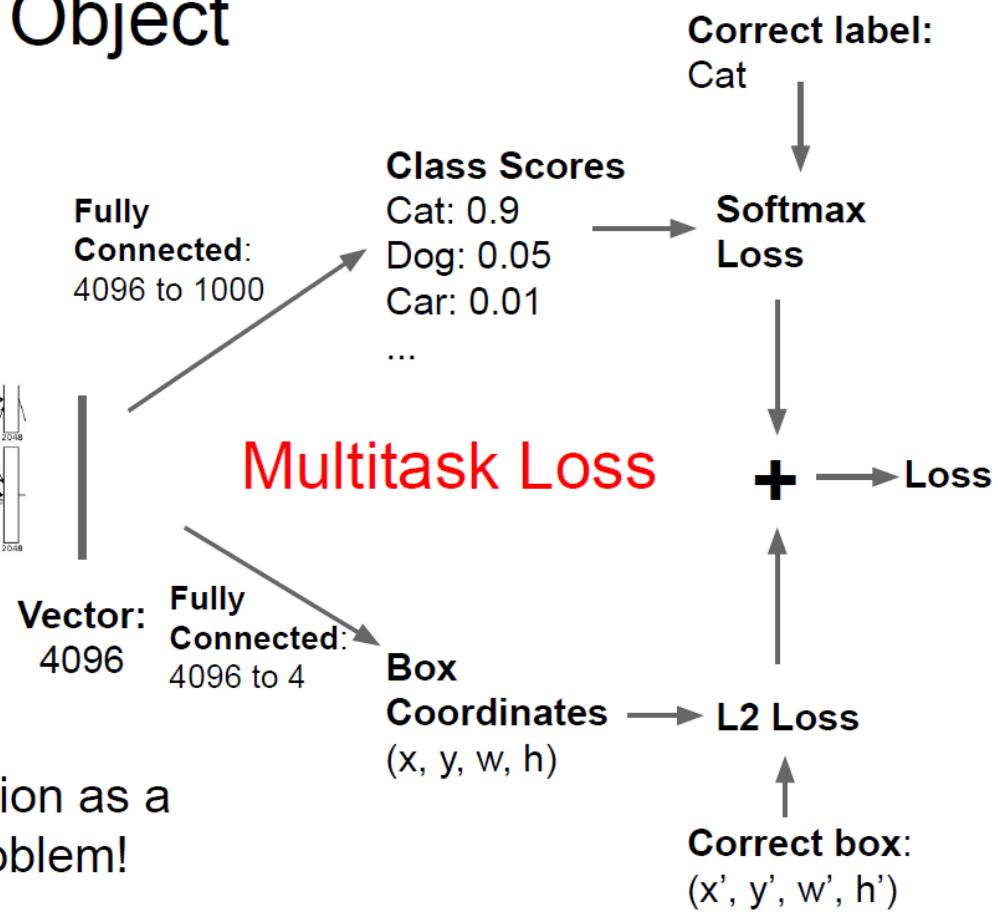
(Classification + Localization)



This image is CC0 public domain



Treat localization as a
regression problem!

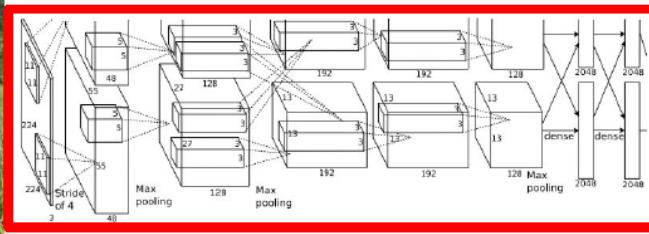


Object Detection: Single Object

(Classification + Localization)

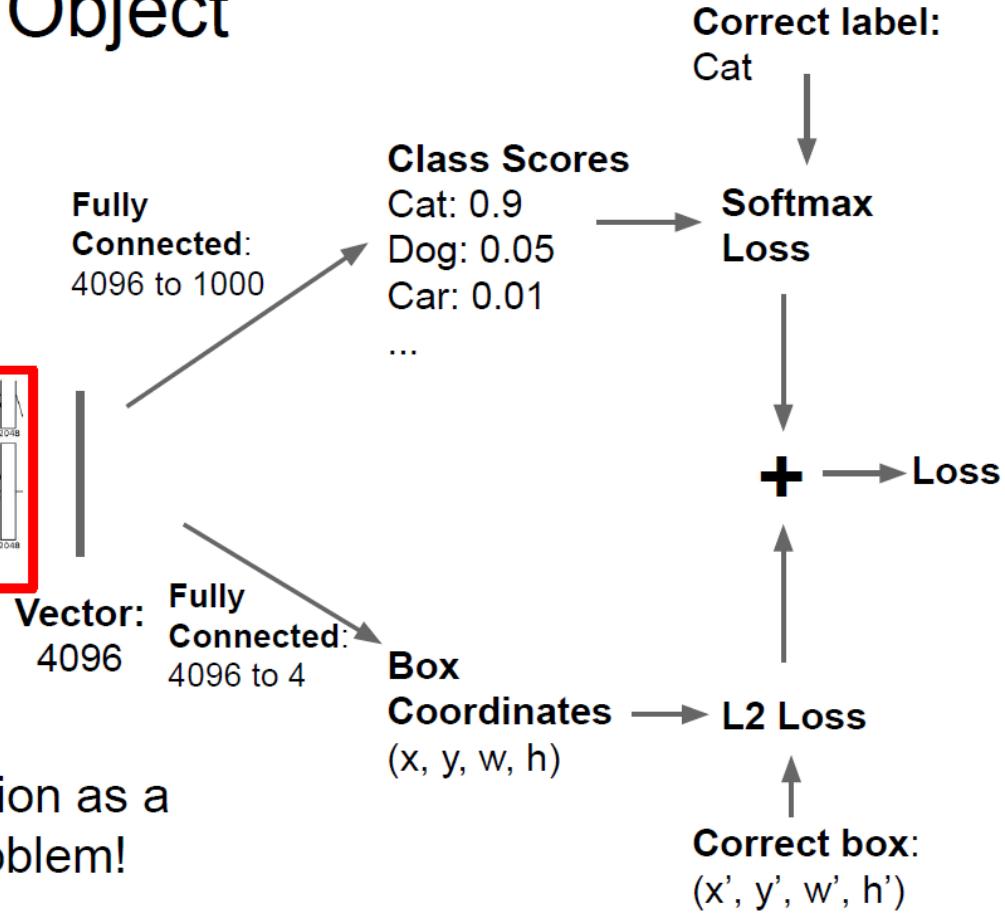


This image is CC0 public domain



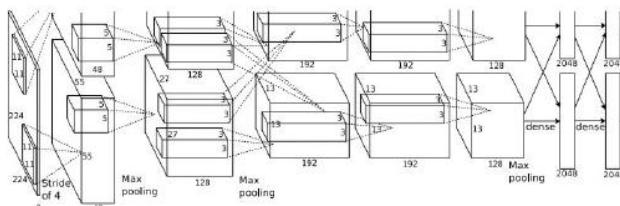
Often pretrained on ImageNet
(Transfer learning)

Treat localization as a
regression problem!

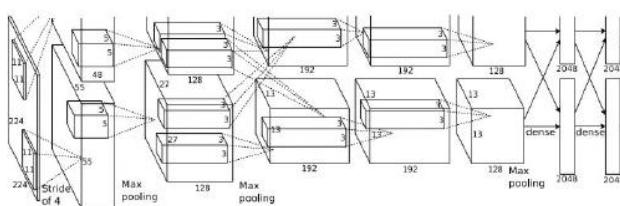


Object Detection: Multiple Objects

Each image needs a different number of outputs!

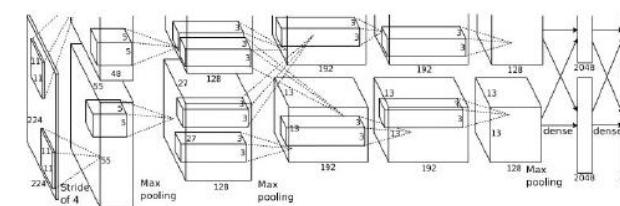


CAT: (x, y, w, h) 4 numbers



DOG: (x, y, w, h)
DOG: (x, y, w, h) 16 numbers

CAT: (x, y, w, h)

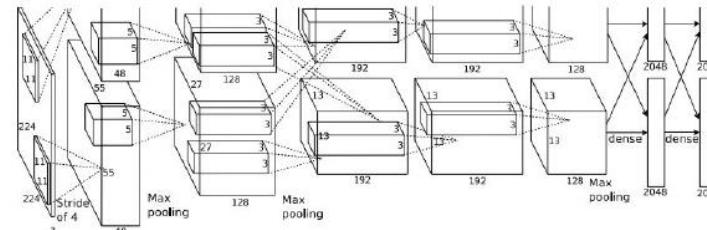


DUCK: (x, y, w, h)
DUCK: (x, y, w, h) Many numbers!

...

Object Detection: Multiple Objects

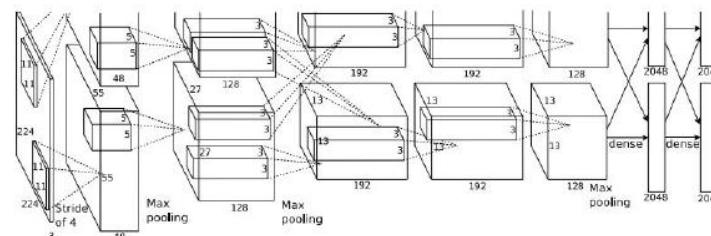
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES
Cat? NO
Background? NO

Object Detection: Multiple Objects

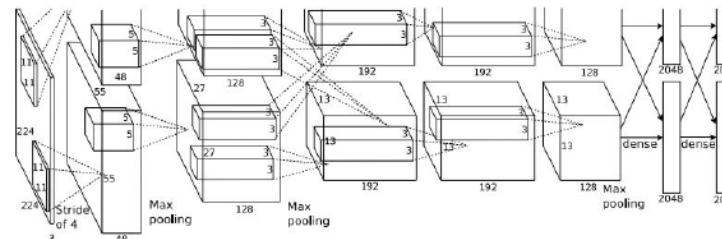
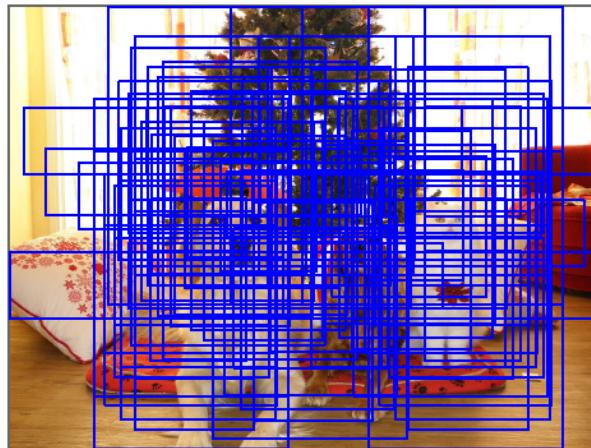
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? YES
Background? NO

Object Detection: Multiple Objects

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

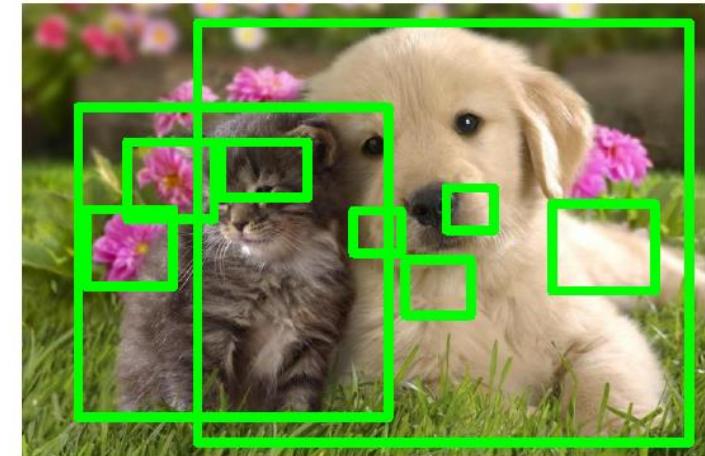


Dog? NO
Cat? YES
Background? NO

Problem: Need to apply CNN to huge number of locations, scales, and aspect ratios, very computationally expensive!

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

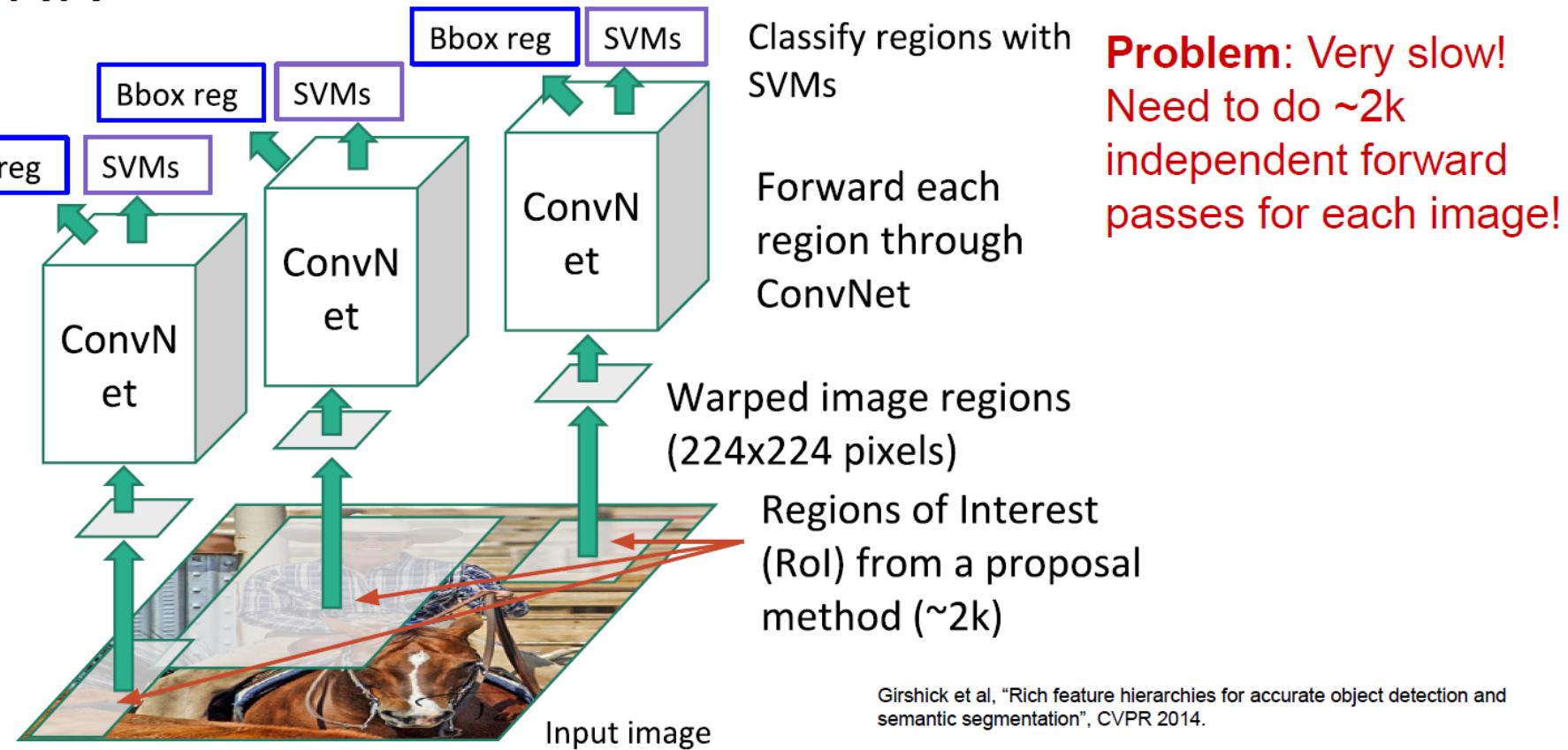
Cheng et al, "BING: Binarized normed gradients for objectness estimation at 300fps", CVPR 2014

Zitnick and Dollar, "Edge boxes: Locating object proposals from edges", ECCV 2014

Convolutional neural networks

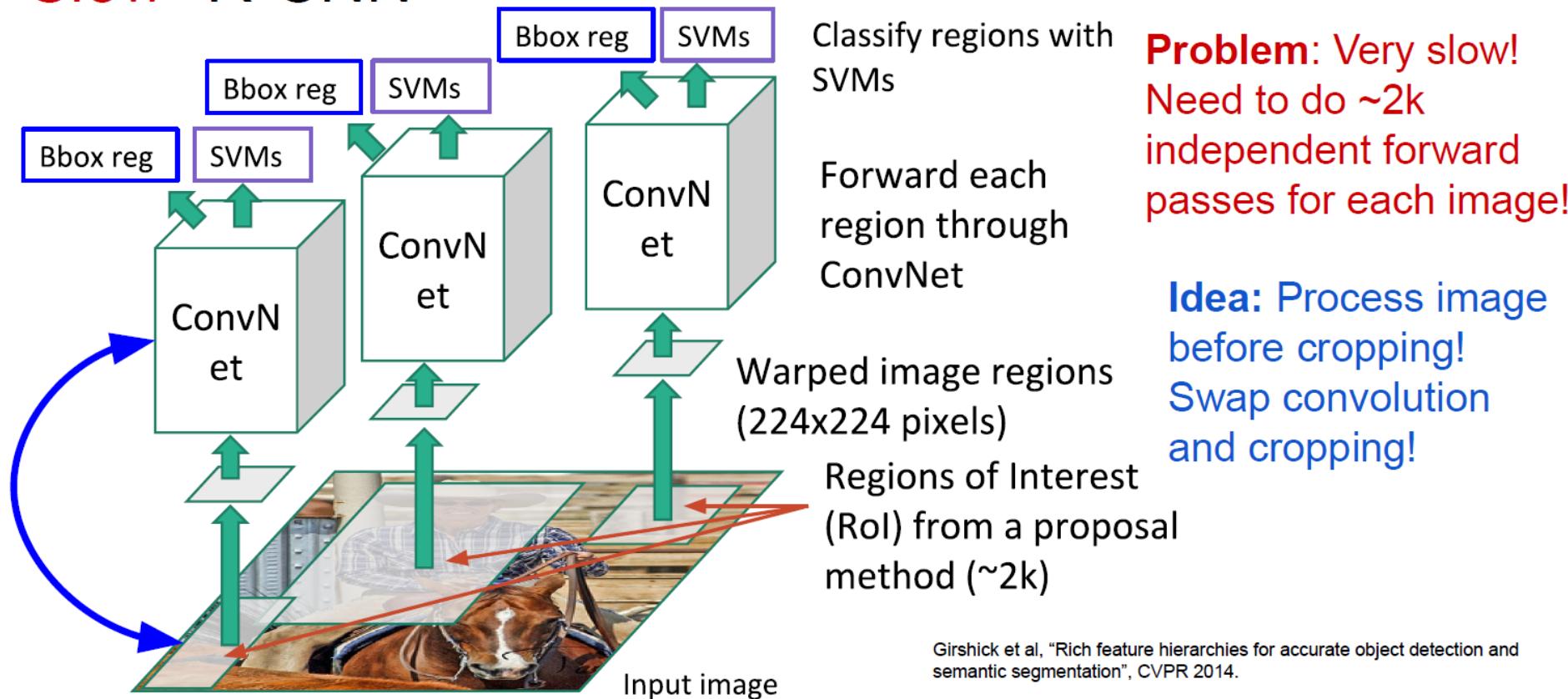
R-CNN

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



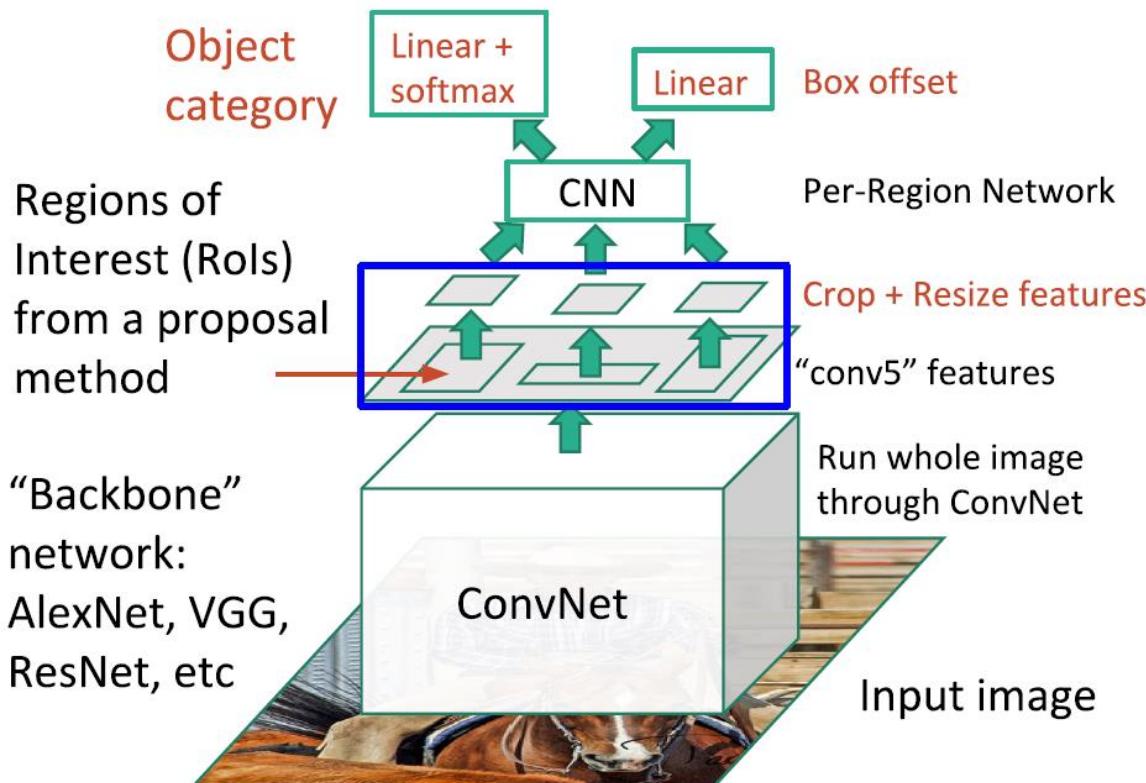
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

“Slow” R-CNN

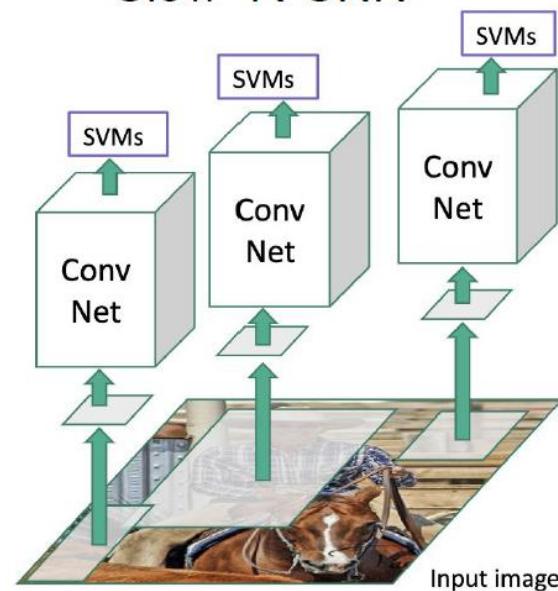


Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

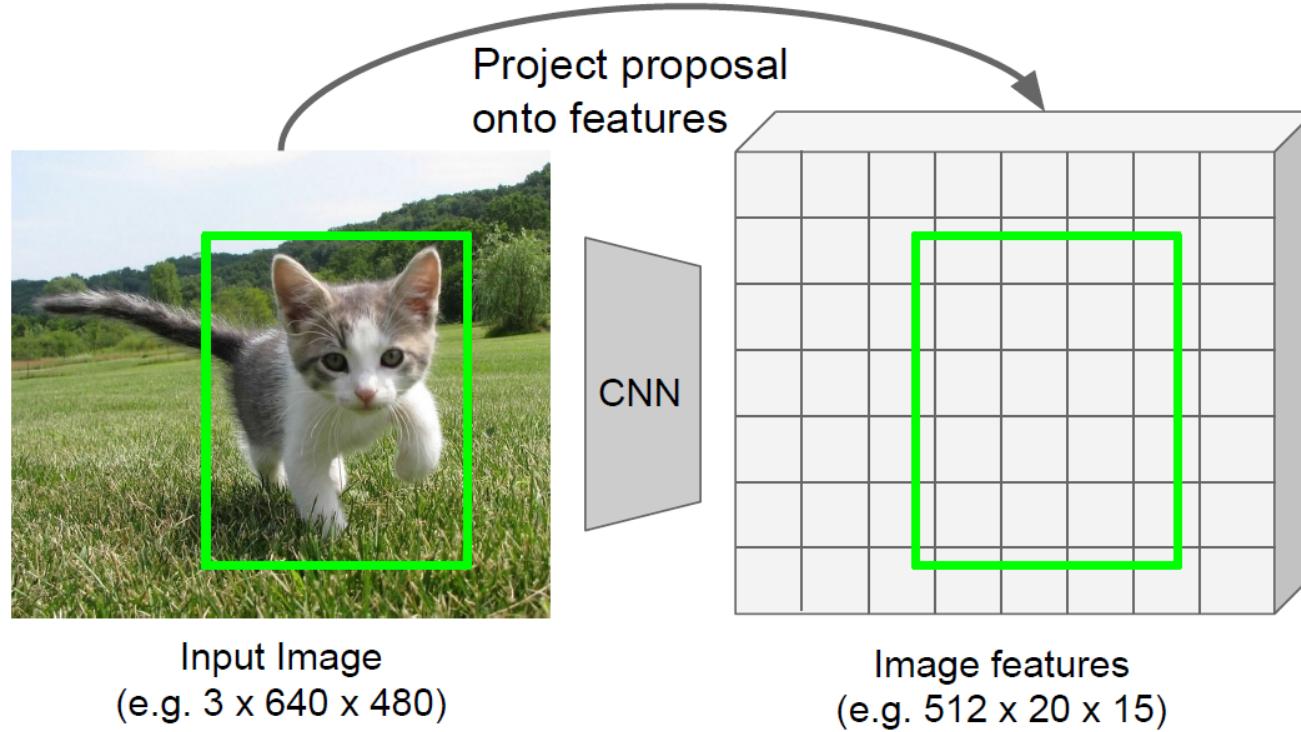
Fast R-CNN



“Slow” R-CNN



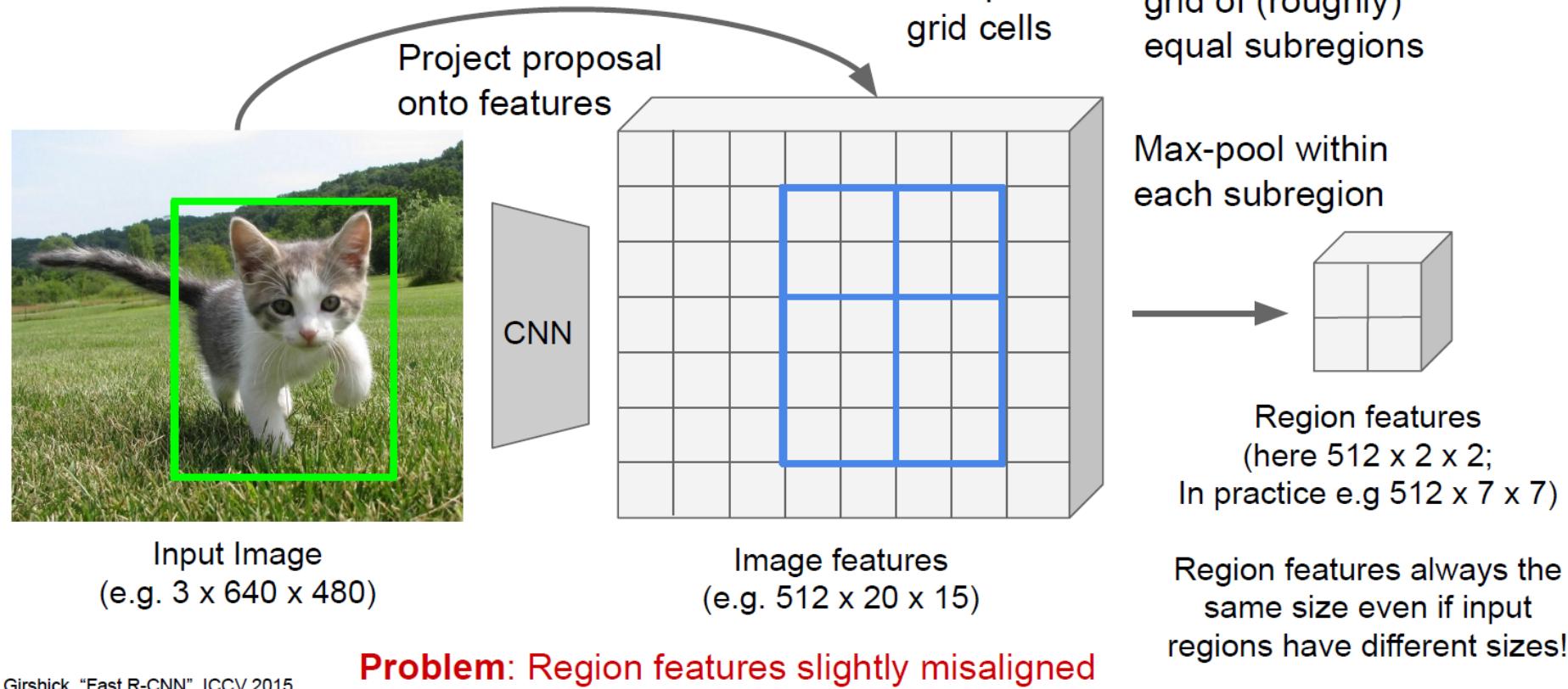
Cropping Features: RoI Pool



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

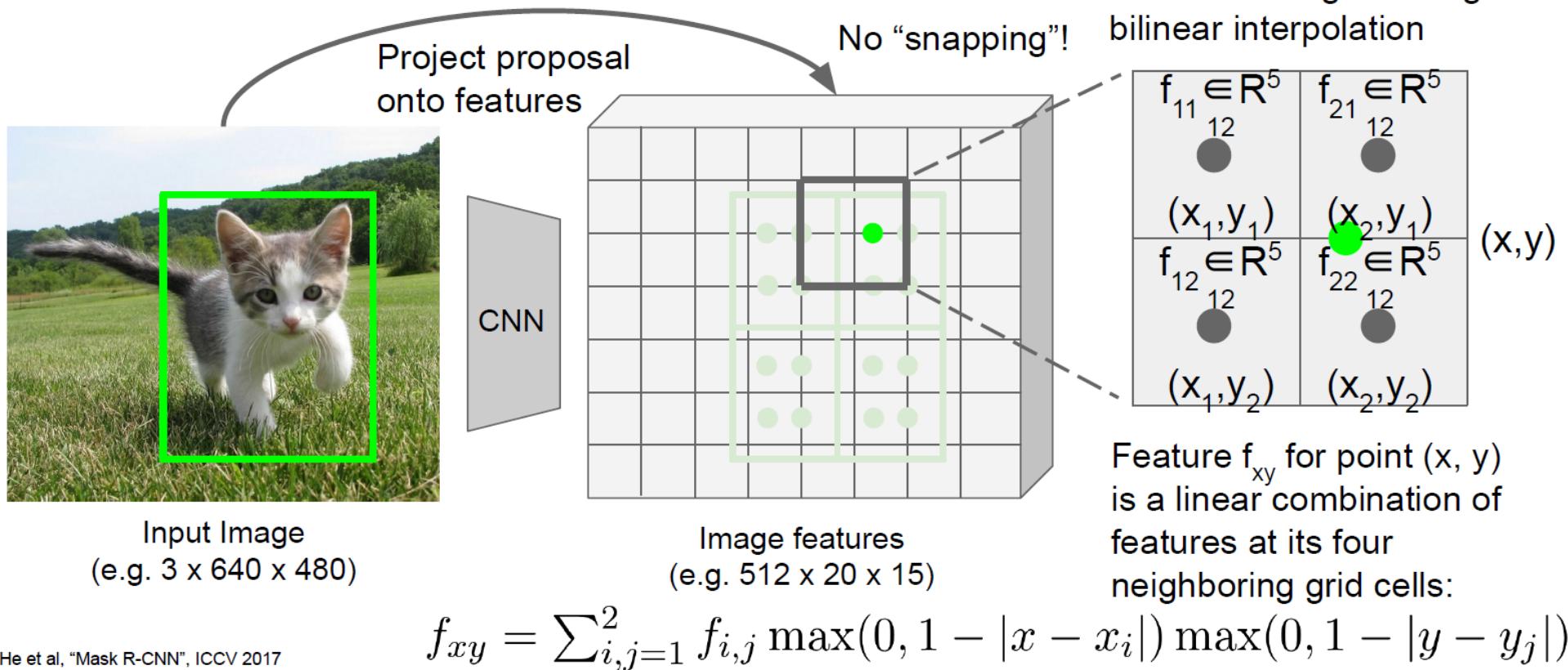
Girshick, "Fast R-CNN", ICCV 2015.

Cropping Features: RoI Pool



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

Cropping Features: RoI Align



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

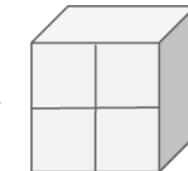
Cropping Features: RoI Align

Project proposal
onto features

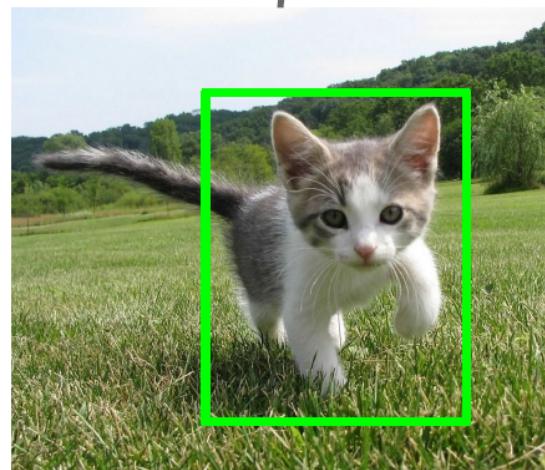
No “snapping”!

Sample at regular points
in each subregion using
bilinear interpolation

Max-pool within
each subregion



Region features
(here $512 \times 2 \times 2$;
In practice e.g $512 \times 7 \times 7$)



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

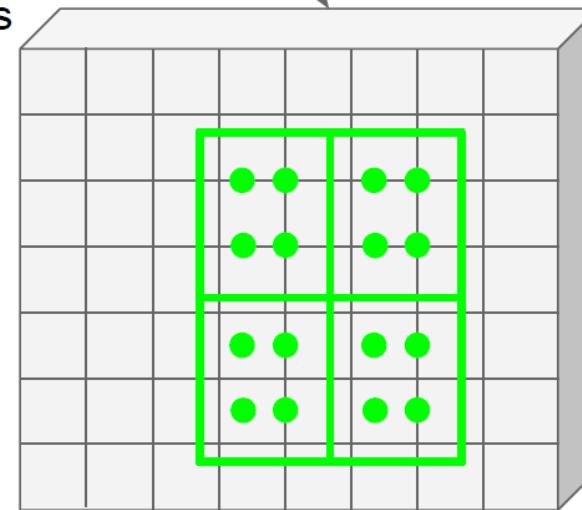
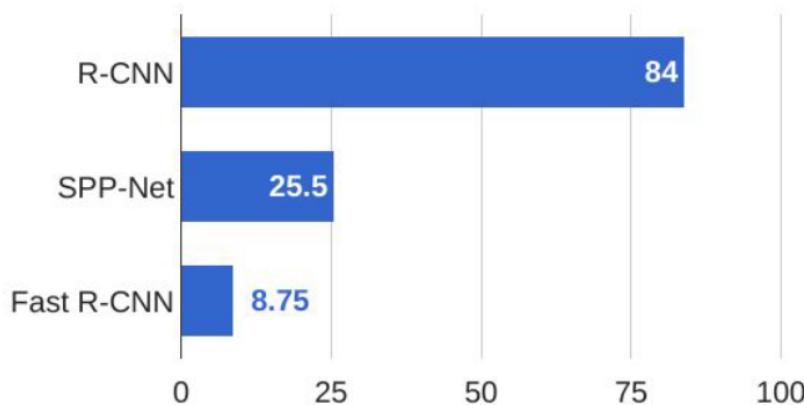


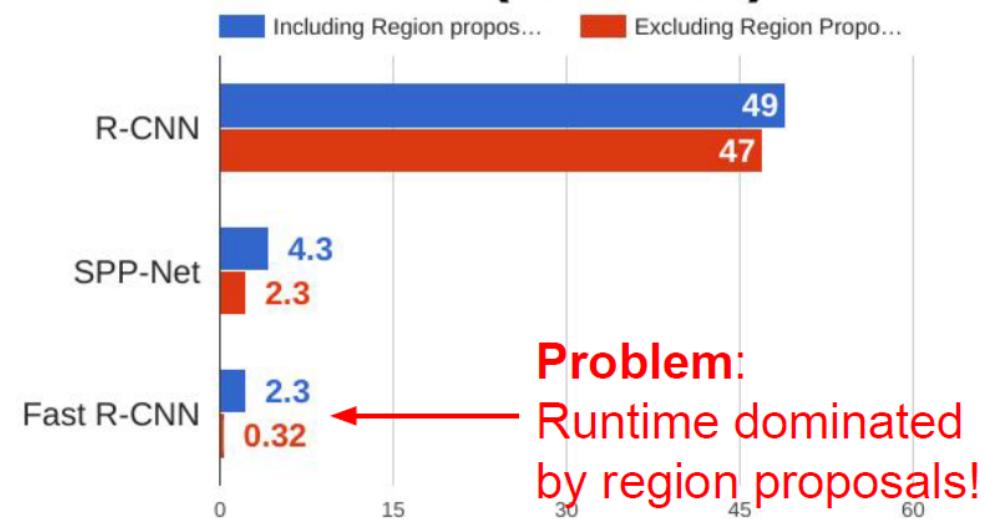
Image features
(e.g. $512 \times 20 \times 15$)

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

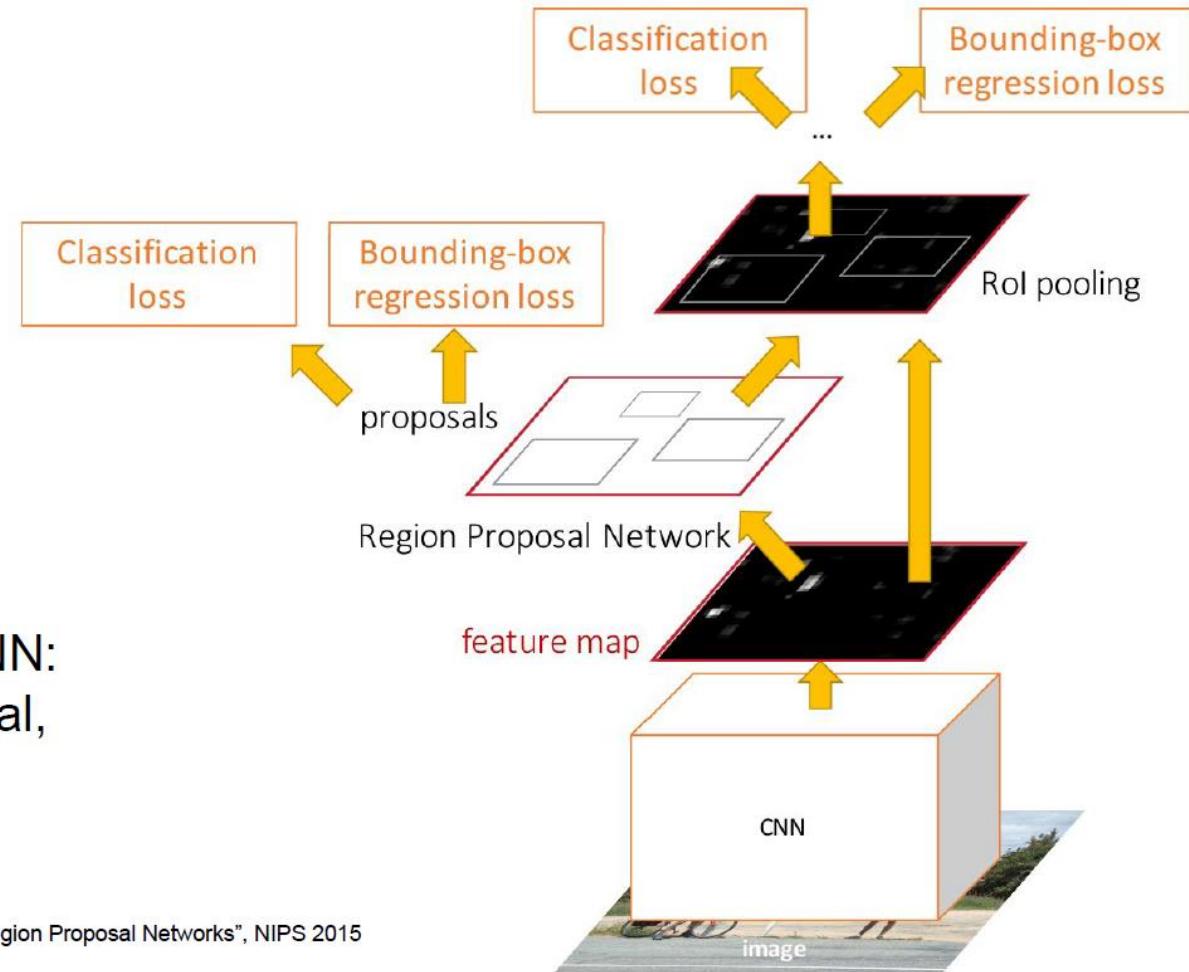
Girshick, "Fast R-CNN", ICCV 2015

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

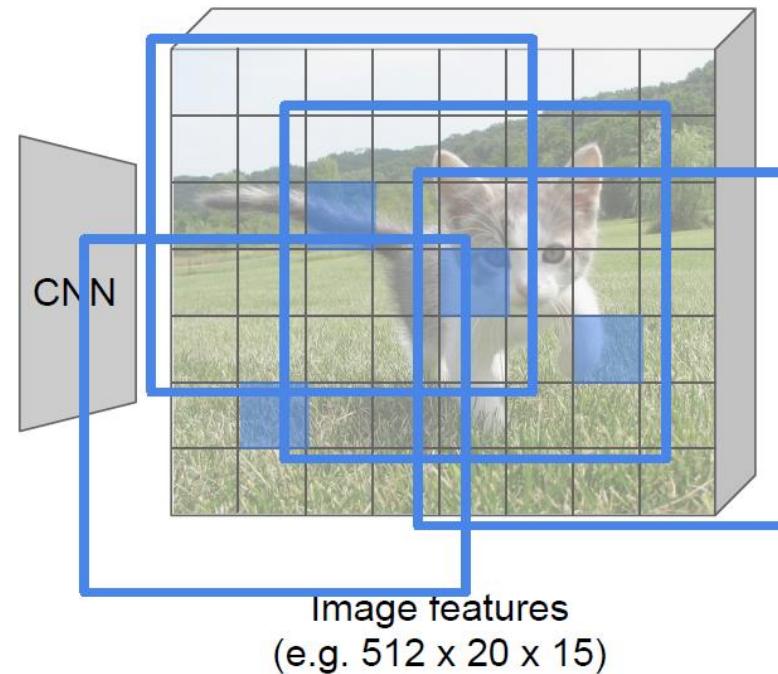


Ren et al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015
Figure copyright 2015, Ross Girshick; reproduced with permission

Region Proposal Network



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

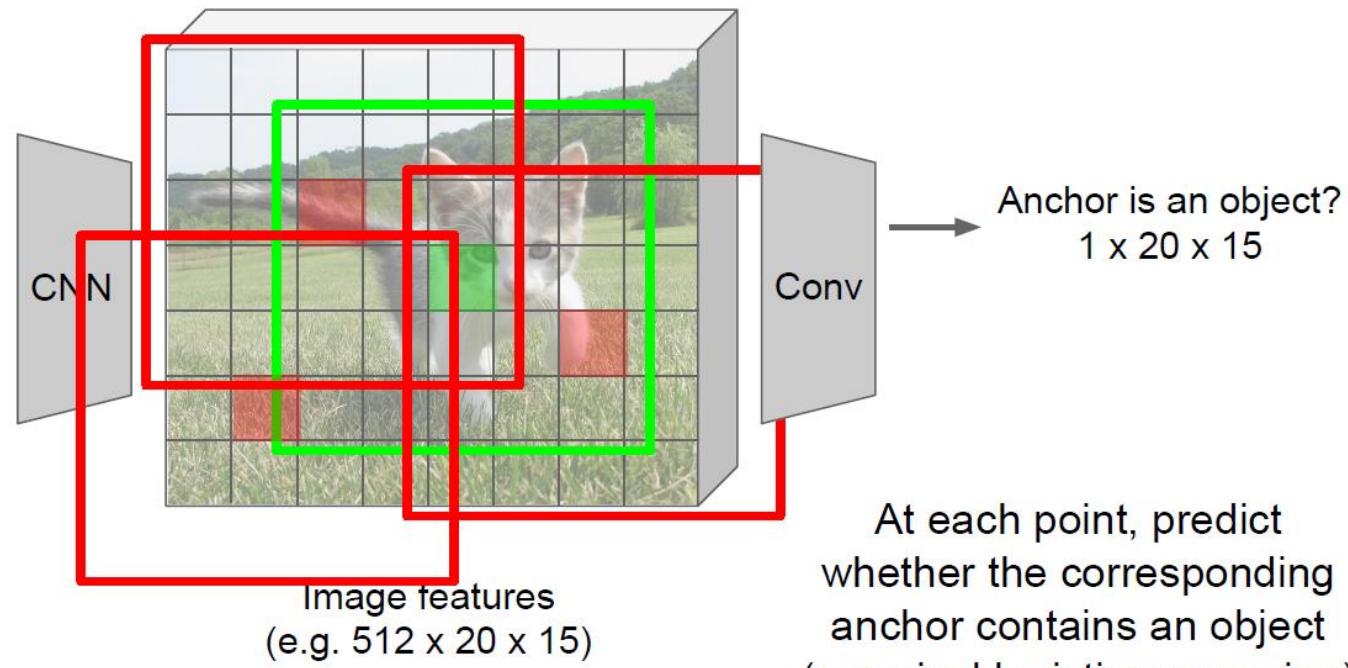


Imagine an **anchor box** of fixed size at each point in the feature map

Region Proposal Network



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



Region Proposal Network



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

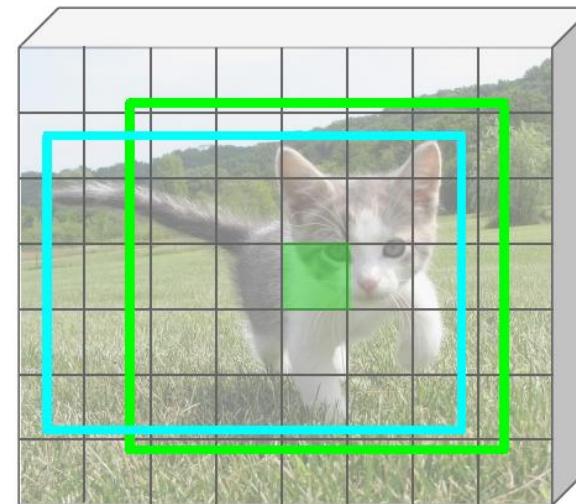


Image features
(e.g. $512 \times 20 \times 15$)



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

Anchor is an object?
 $1 \times 20 \times 15$

Box transforms
 $4 \times 20 \times 15$

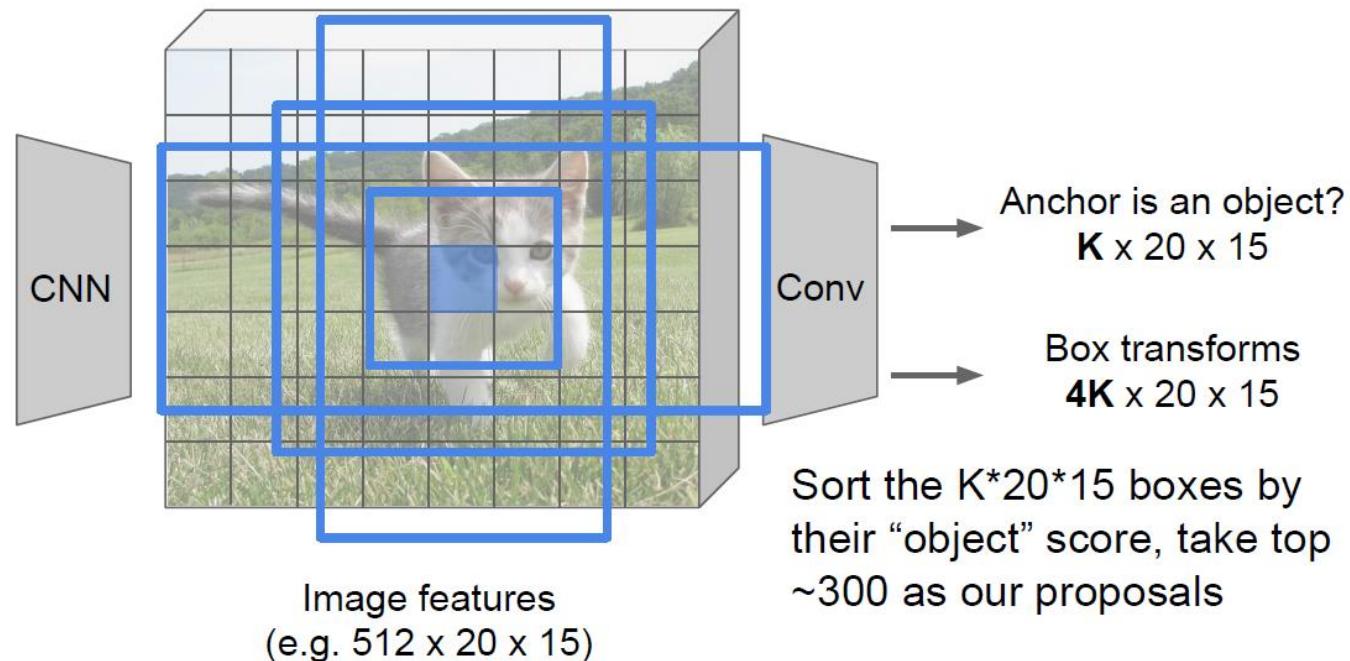
For positive boxes, also predict a transformation from the anchor to the ground-truth box (regress 4 numbers per pixel)

Region Proposal Network

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



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

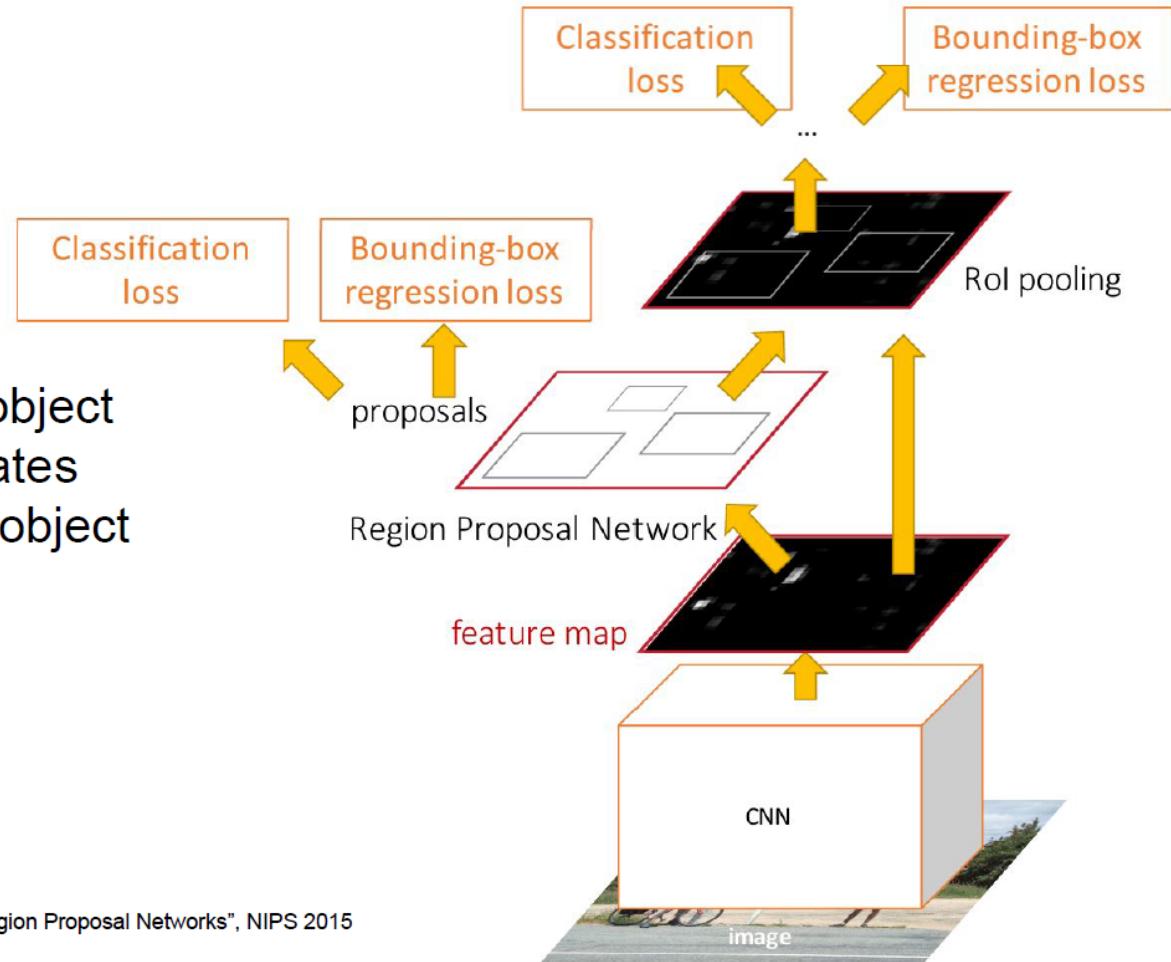


Faster R-CNN:

Make CNN do proposals!

Jointly train with 4 losses:

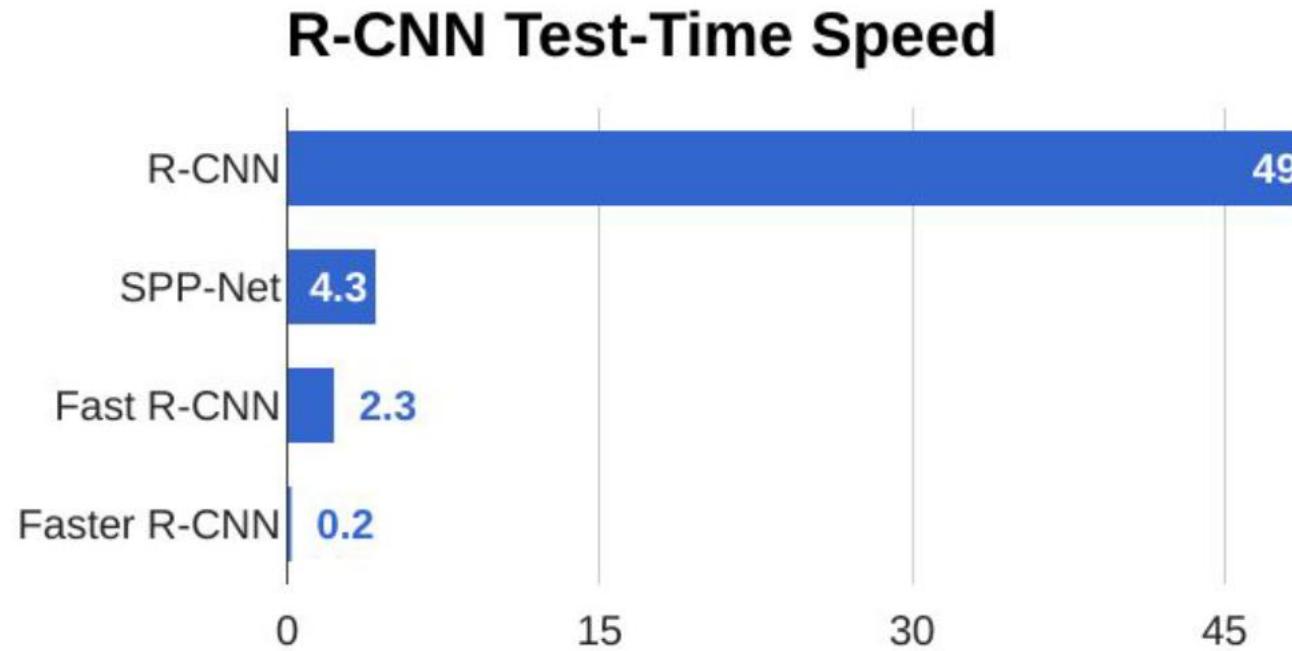
1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates



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

Faster R-CNN:

Make CNN do proposals!



Faster R-CNN:

Make CNN do proposals!

Faster R-CNN is a
Two-stage object detector

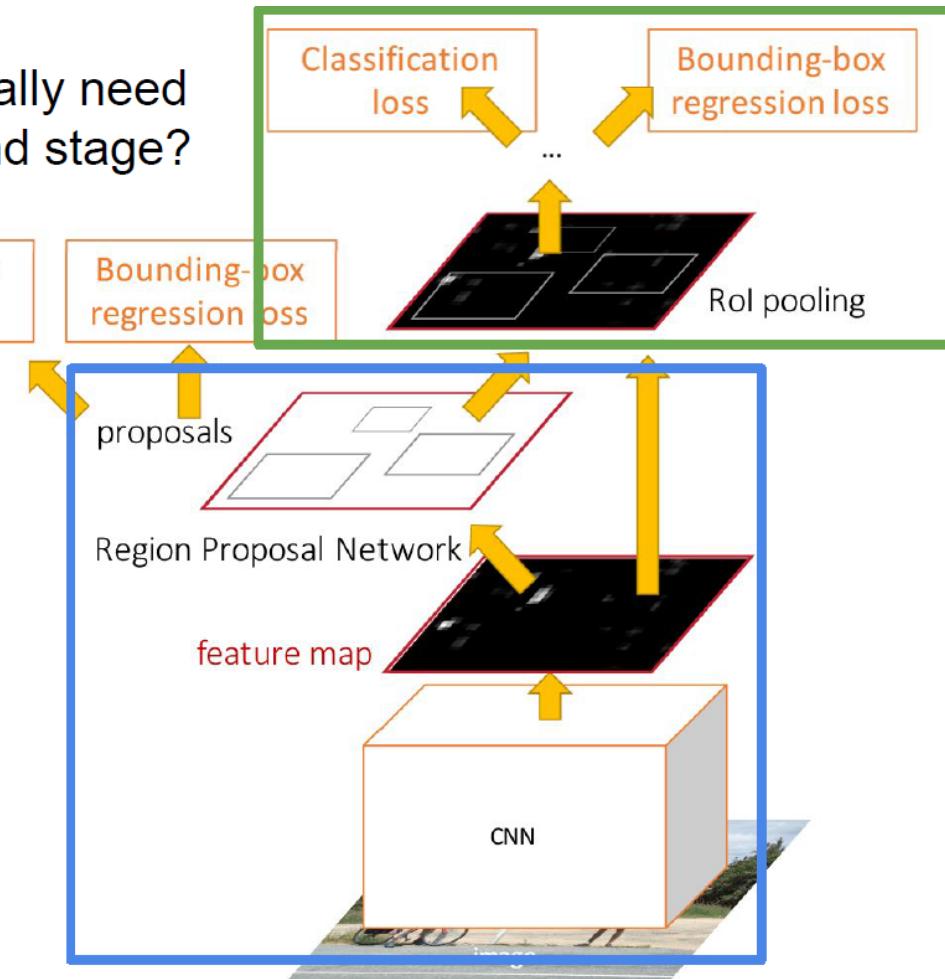
First stage: Run once per image

- Backbone network
- Region proposal network

Second stage: Run once per region

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

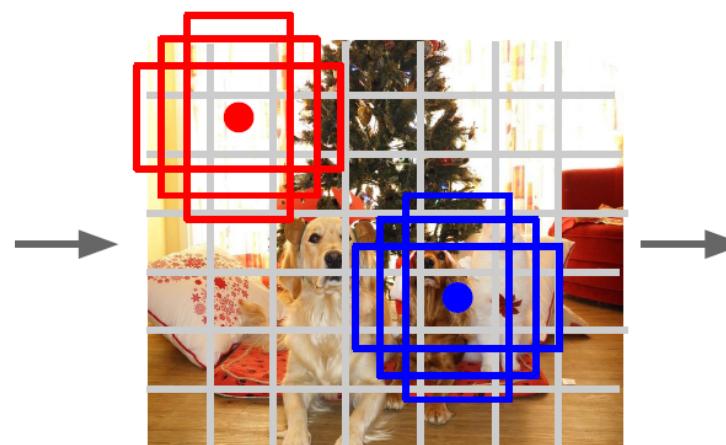
Do we really need
the second stage?



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: $(dx, dy, dh, dw, \text{confidence})$
- 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

Instance Segmentation

Classification



CAT

No spatial extent

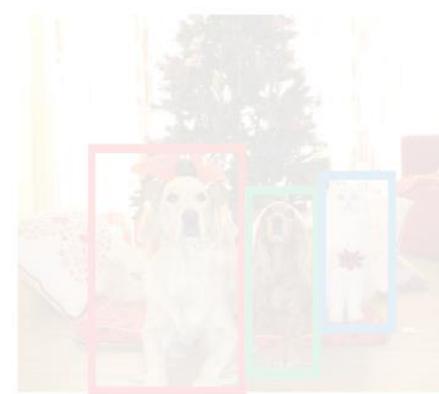
Semantic
Segmentation



GRASS, CAT,
TREE, SKY

No objects, just pixels

Object
Detection



DOG, DOG, CAT

Instance
Segmentation



DOG, DOG, CAT

Multiple Object

Instance Segmentation: Mask R-CNN

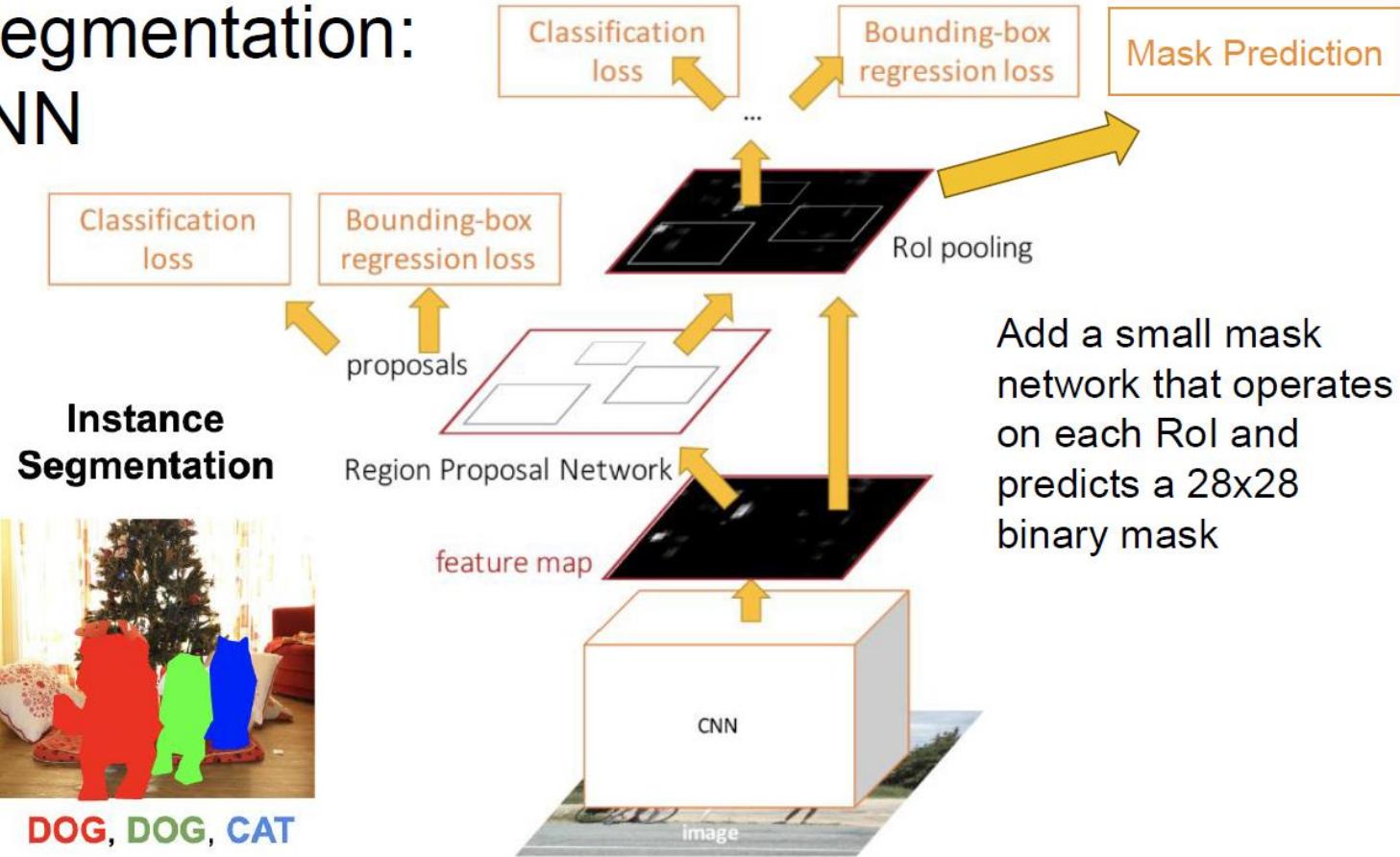


Object
Detection



Instance
Segmentation

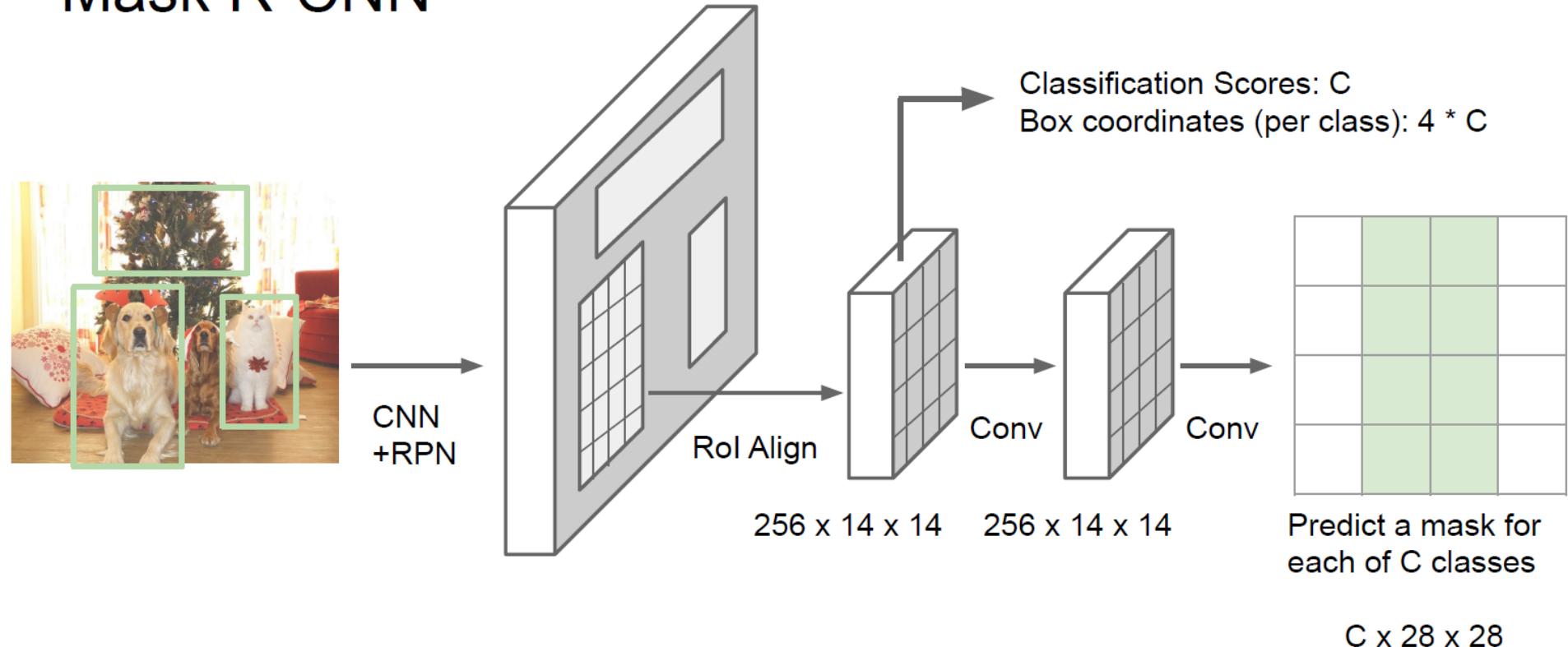
DOG, DOG, CAT



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

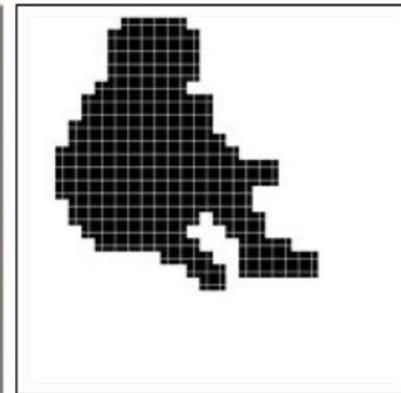
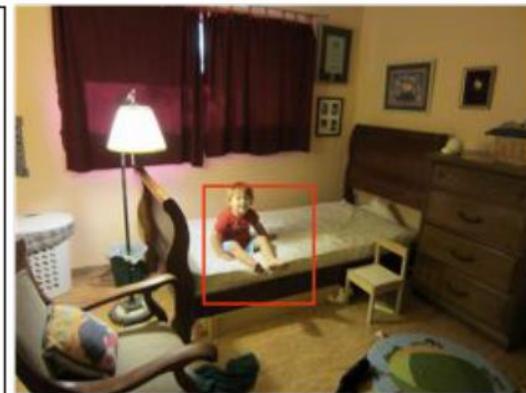
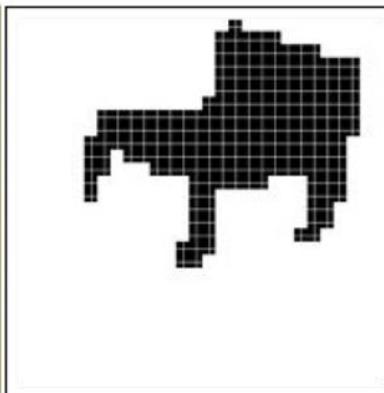
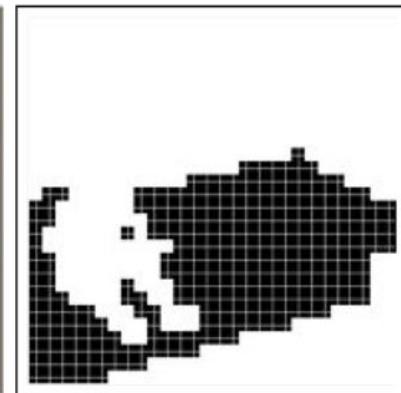
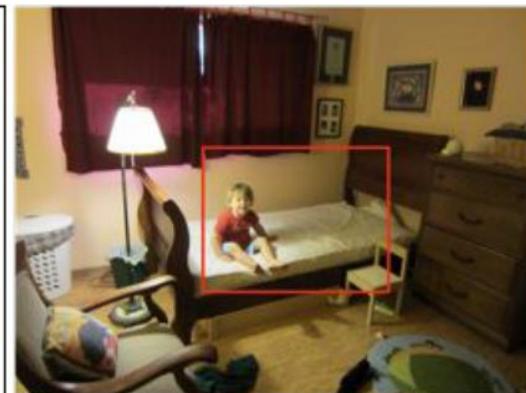
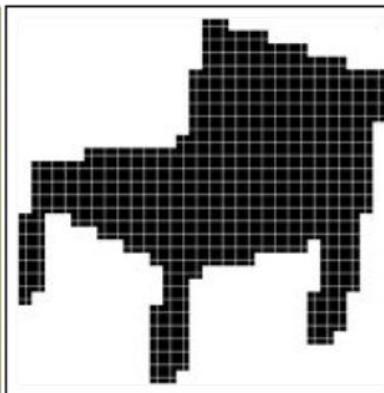
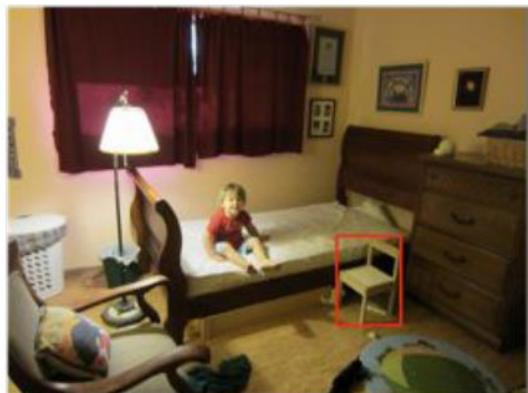
Convolutional neural networks

Mask R-CNN

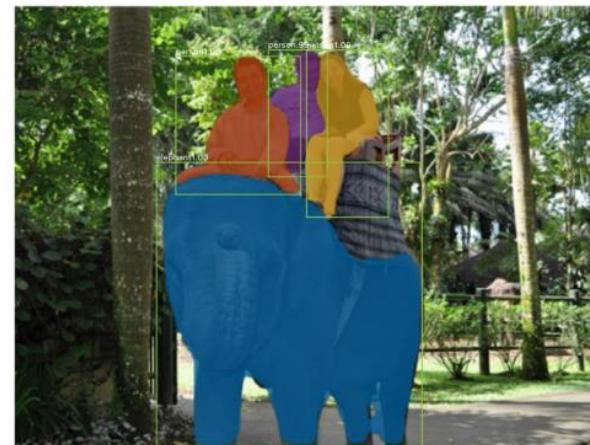


He et al, "Mask R-CNN", arXiv 2017

Mask R-CNN: Example Mask Training Targets



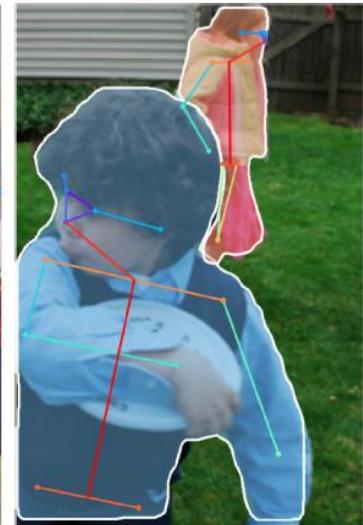
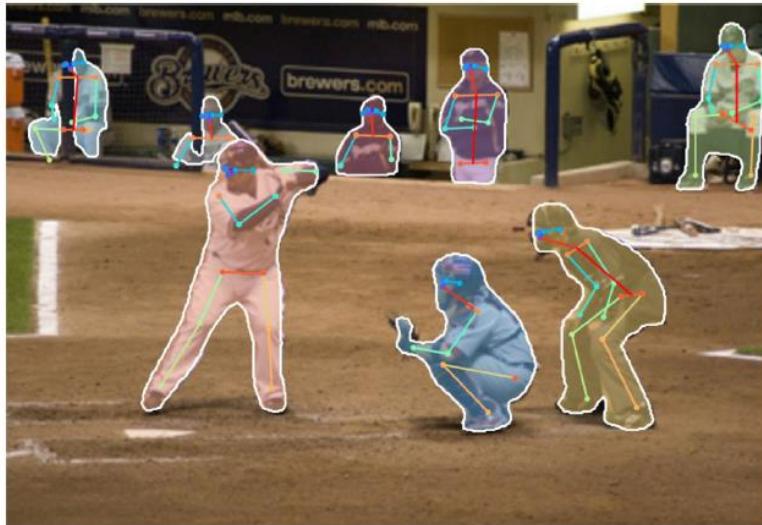
Mask R-CNN: Very Good Results!



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

Mask R-CNN

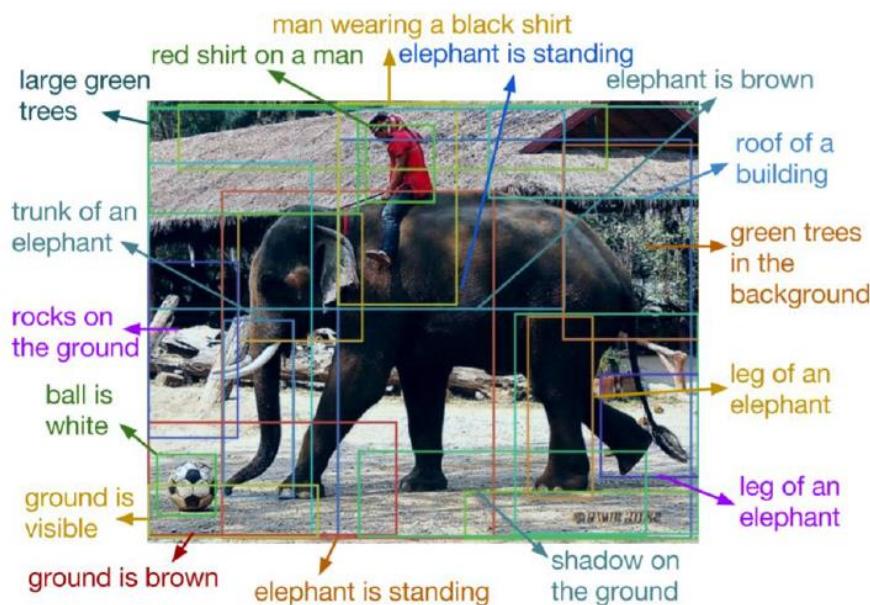
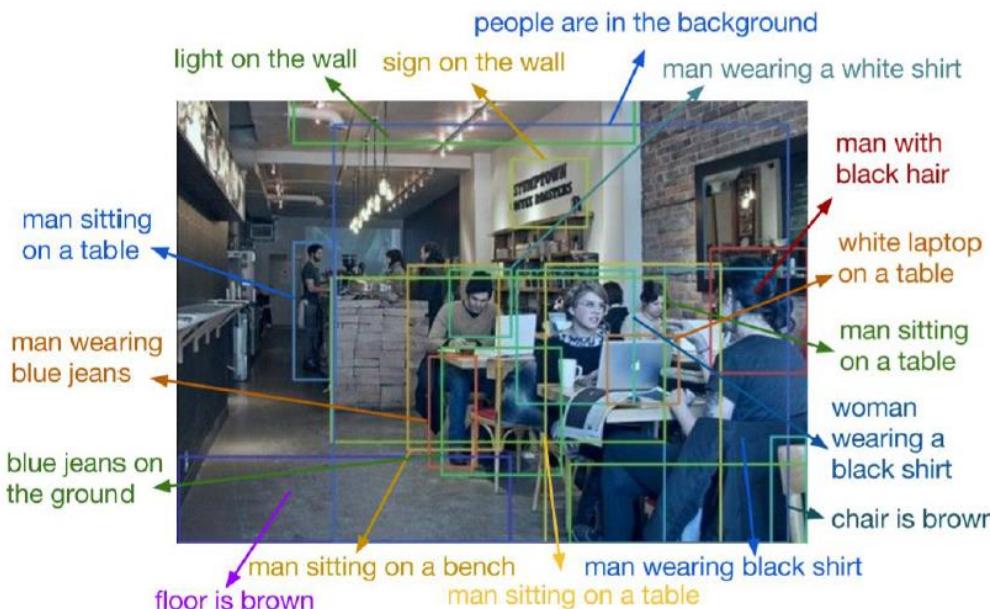
Also does pose



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

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.