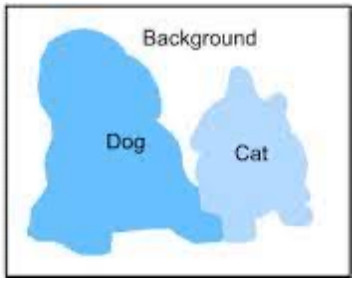
Computer Vision

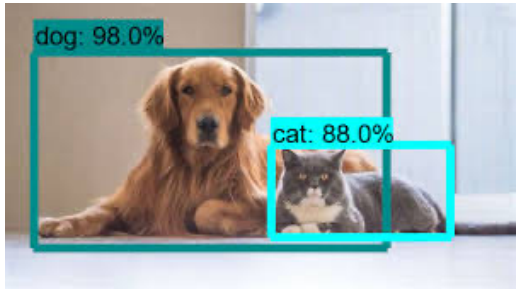
Structured Predictions
with Convolutional Neural Networks

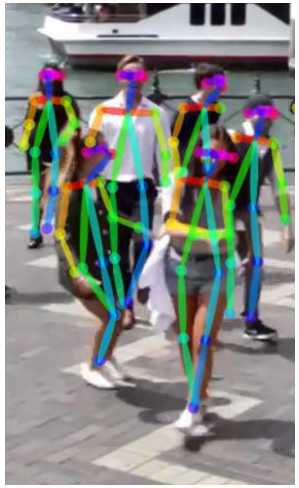
Luis F. Teixeira | M.EIC | FEUP

More complex outputs

- Attributes
- Image Output
  - e.g. colorization, semantic segmentation, super-resolution, stylization, depth estimation...)
- Object Detections
- Semantic Keypoints
- Text Captions

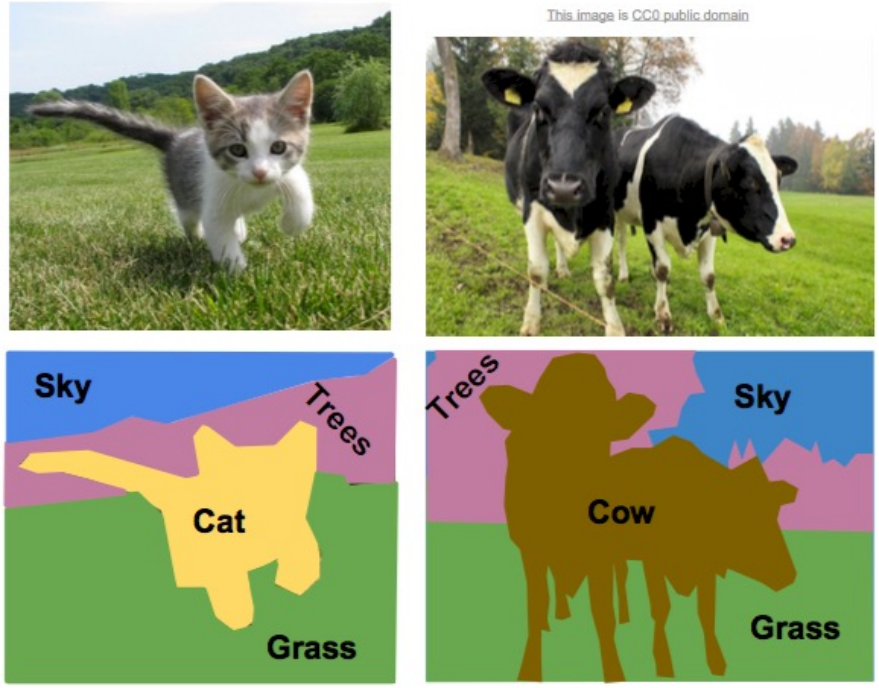






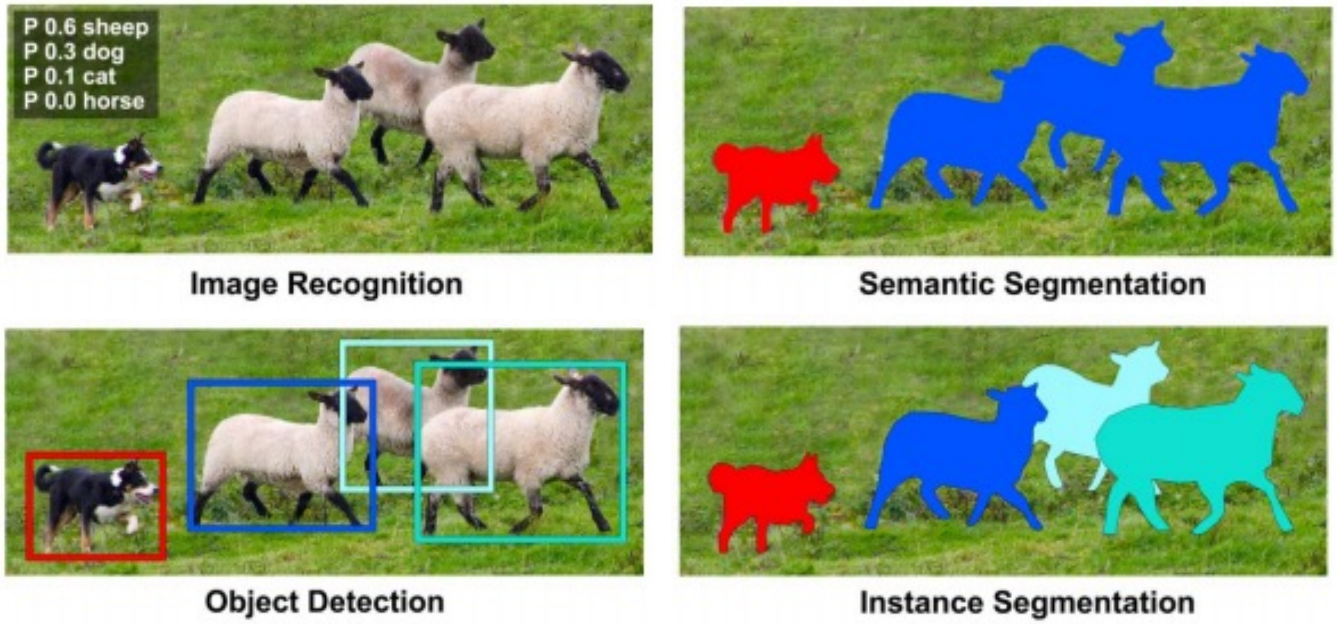
Segmentation

- Image segmentation partitions the image into meaningful regions → predict a label at every pixel



Segmentation

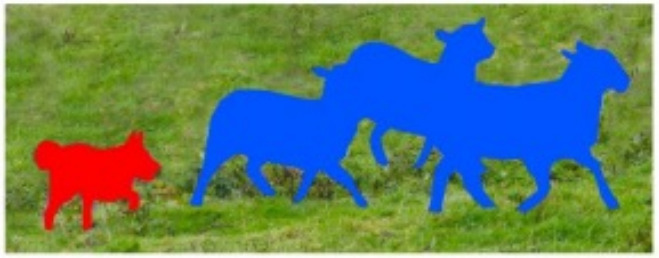
- **Semantic segmentation** assigns a pixel value or label to every concept in the image
- **Instance segmentation** assigns also a numeric value for every instance of the same semantic value



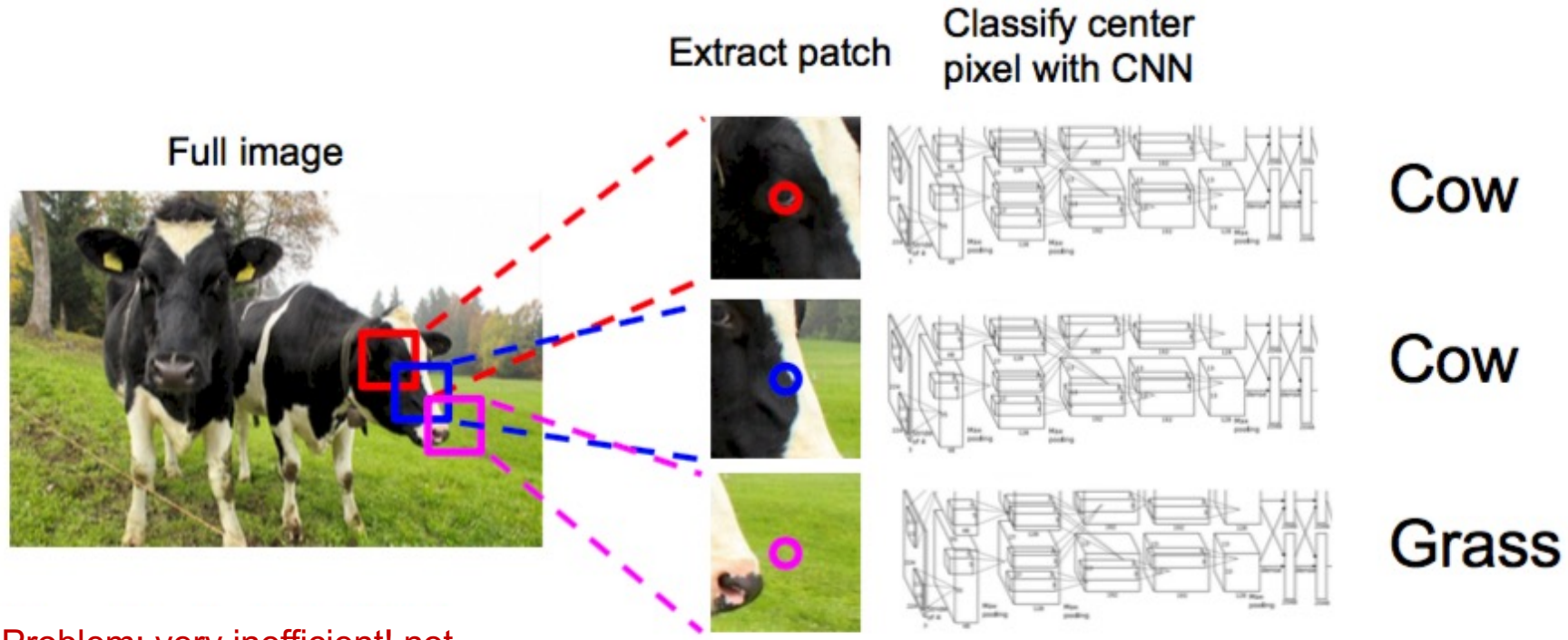
Semantic Segmentation

- How can we do semantic segmentation?
  - Until now a network gives us a prediction for the whole image

vs.



Sliding Window

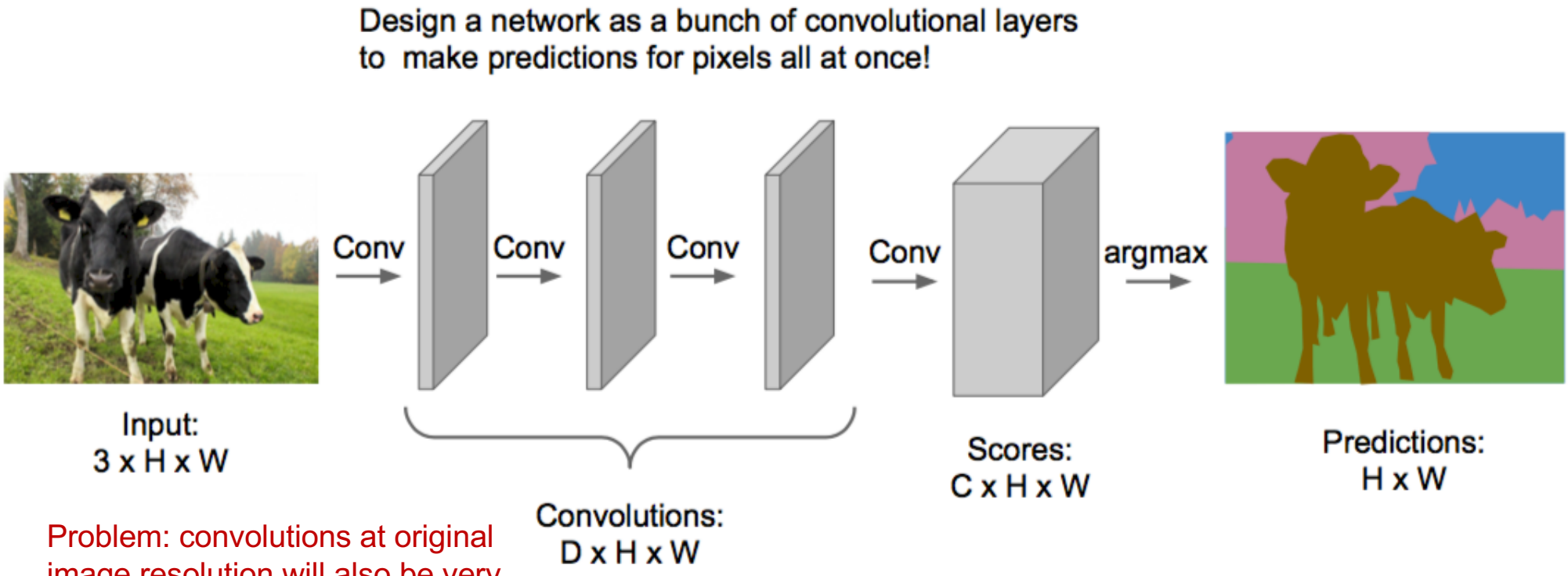


Problem: very inefficient! not reusing shared features between overlapping patches

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

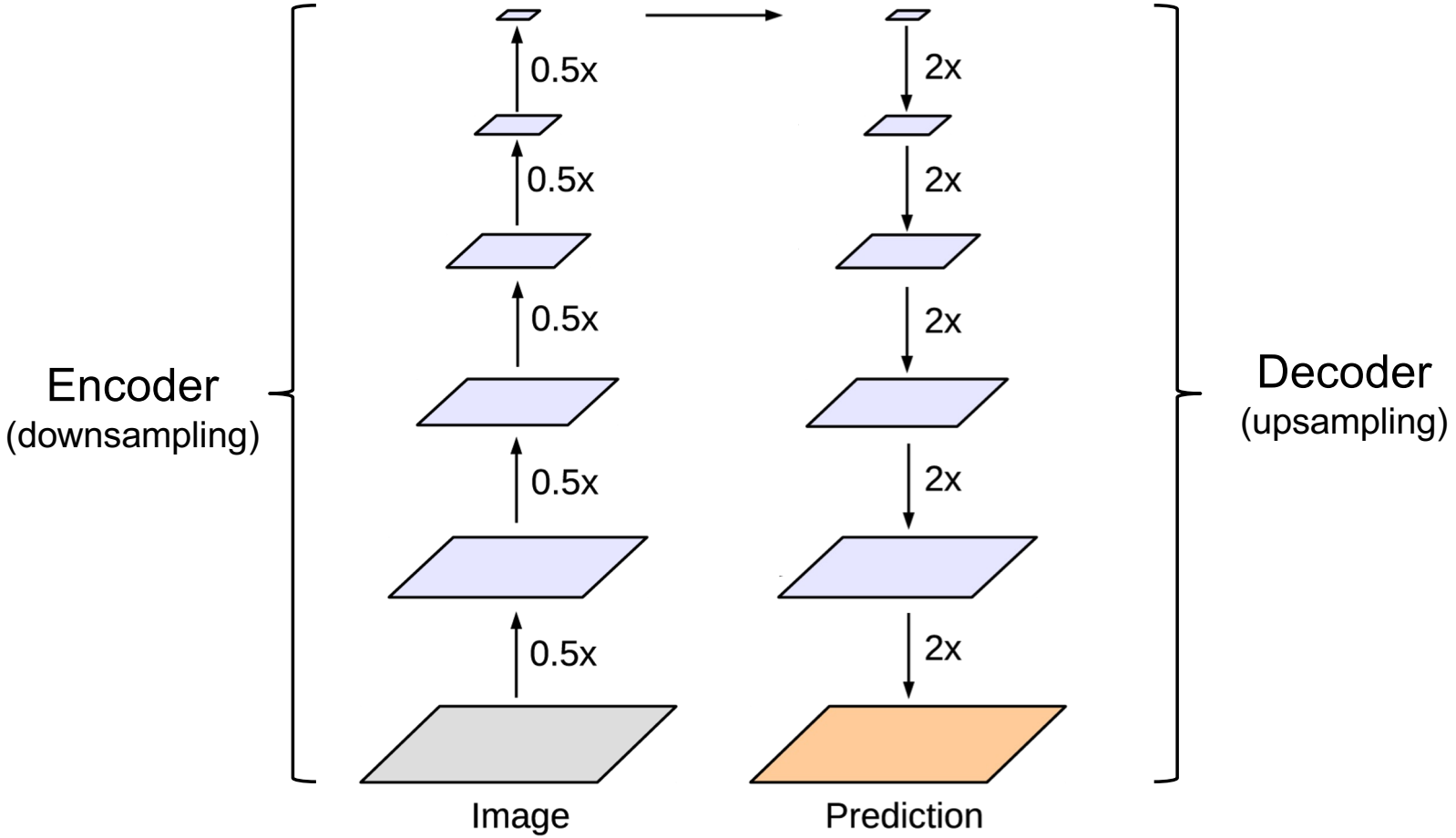
Fully Convolutional Network (FCN)

(no fully connected layer is used)



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

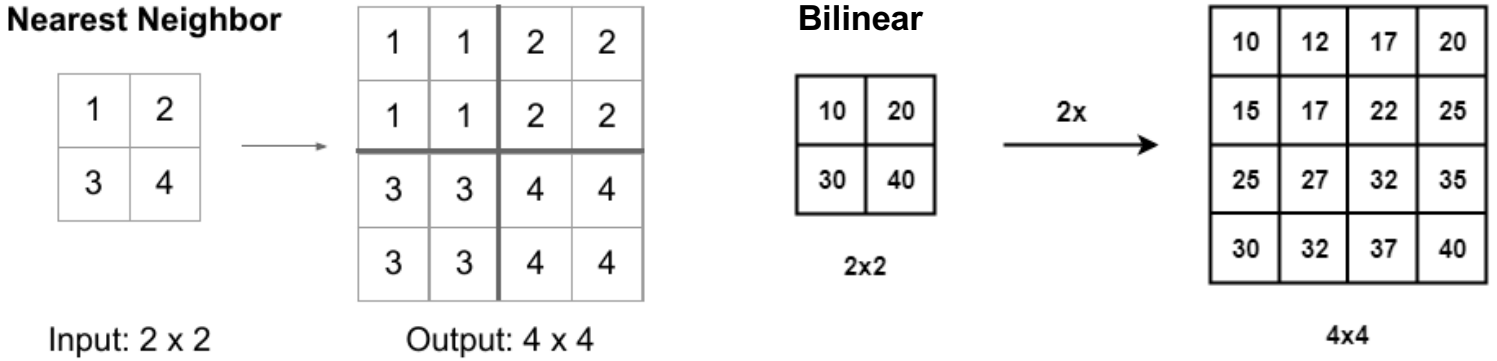
Fully Convolutional Network (FCN)



How can we do the decoding/upsampling?

Upsampling

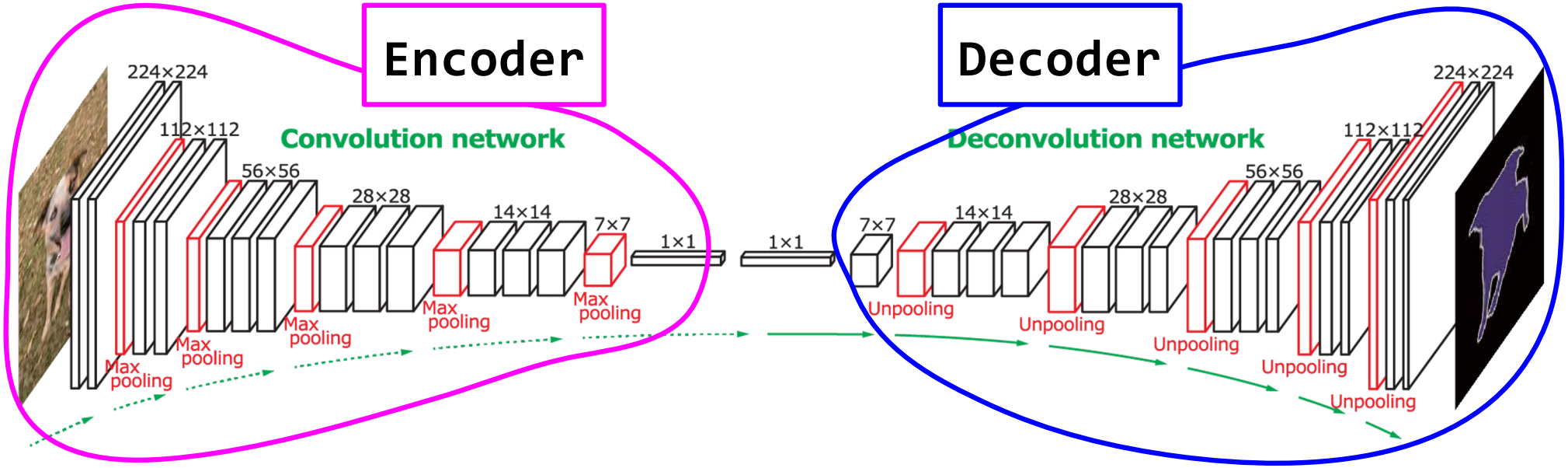
- Any kind of classical interpolation would be acceptable: nearest-neighbor, bilinear, bicubic, etc.



- However, the most common approach in deep learning is **transposed convolution** (sometimes called deconvolution)

  → The decoder will also be learned

‘Deconvolutional’ networks *learn to upsample*



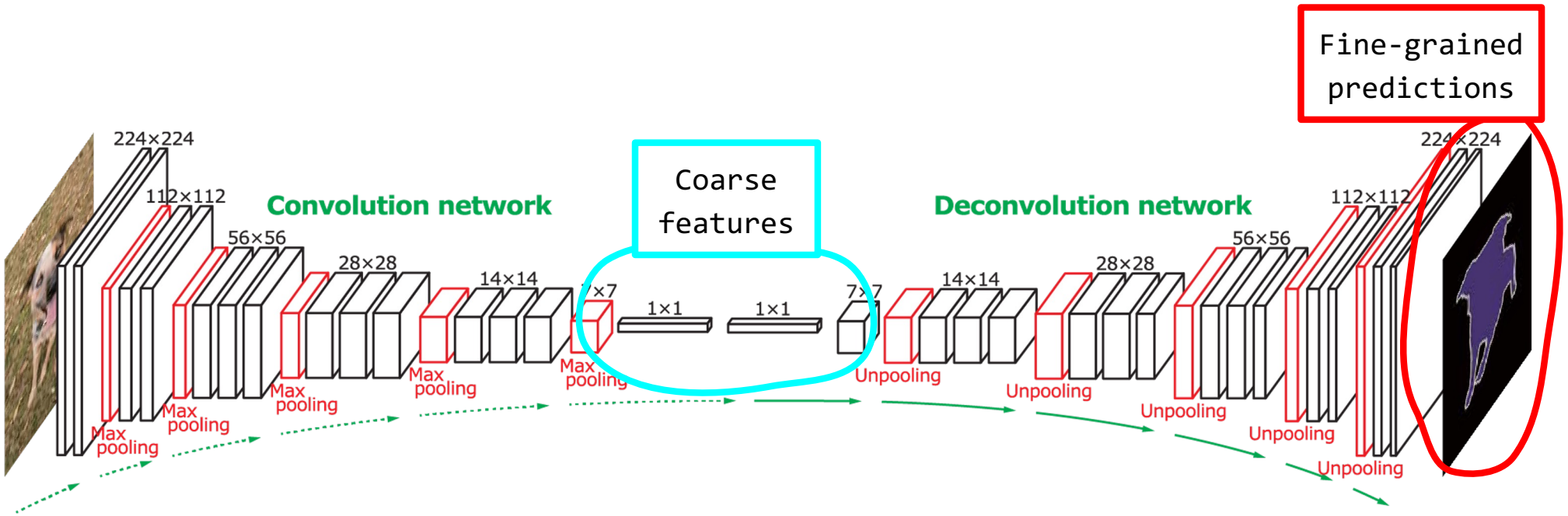
Zeiler et al., Deconvolutional Networks, CVPR 2010

Long et al., Fully Convolutional Models for Semantic Segmentation, CVPR 2015

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

Image source: https://arxiv.org/pdf/1505.04366.pdf

‘Deconvolutional’ networks *learn to upsample*

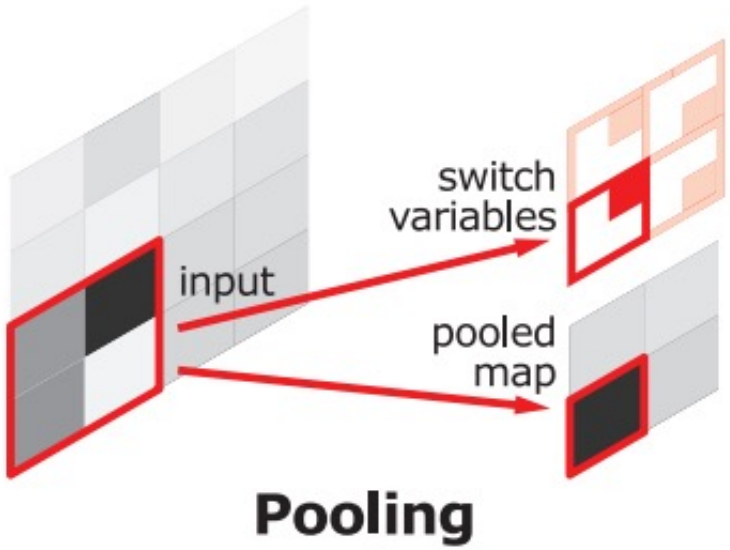


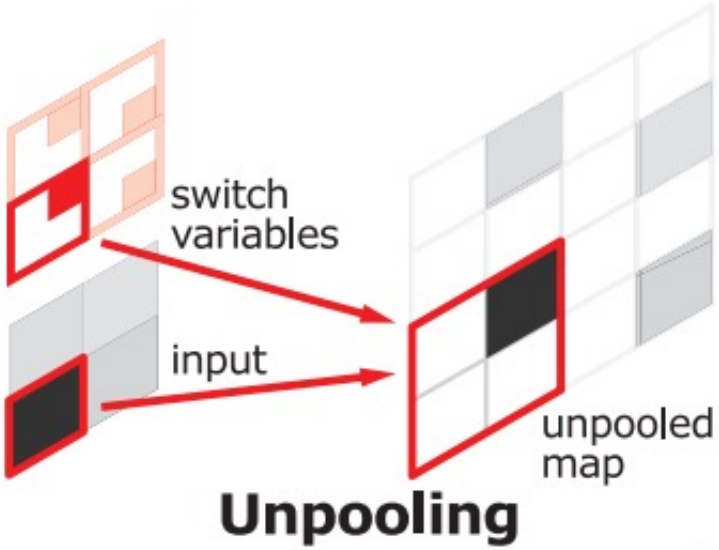
Zeiler et al., Deconvolutional Networks, CVPR 2010

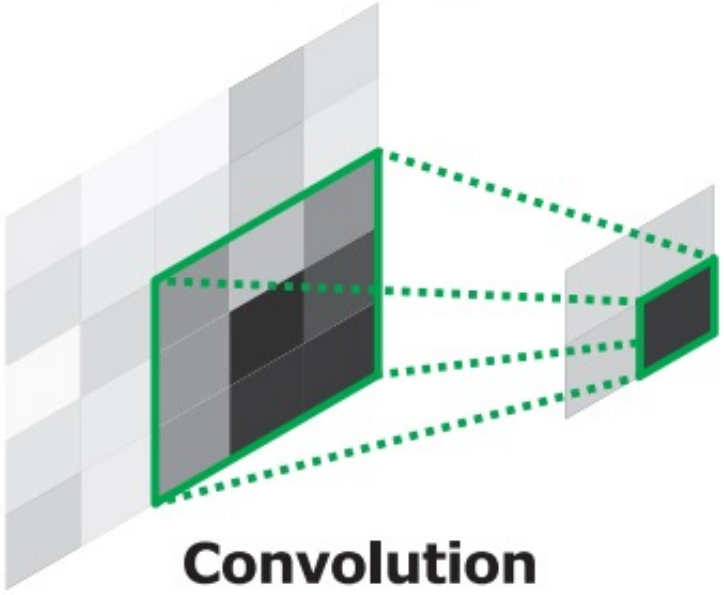
Long et al., Fully Convolutional Models for Semantic Segmentation, CVPR 2015

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

Image source: https://arxiv.org/pdf/1505.04366.pdf







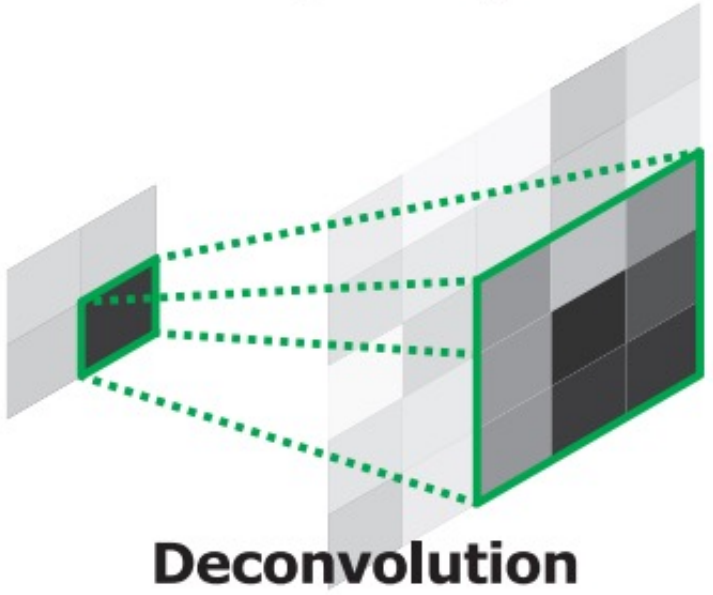


Image source: https://arxiv.org/pdf/1505.04366.pdf

Upsampling by “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

5	6
7	8

Output: 2 x 2

Rest of the network

Max Unpooling
Use positions from pooling layer

1	2
3	4

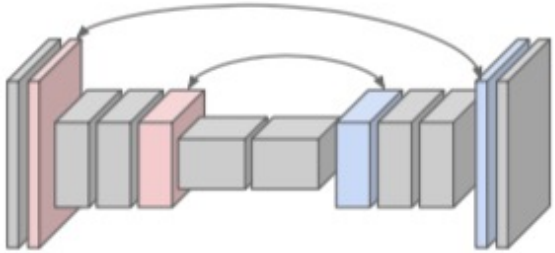
Input: 2 x 2

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

(switch variables)

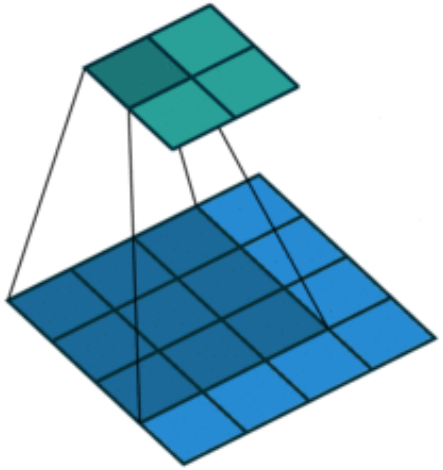


The output of an unpooling layer is an **enlarged, yet sparse** activation map → deconvolution layers **densify** the sparse activations

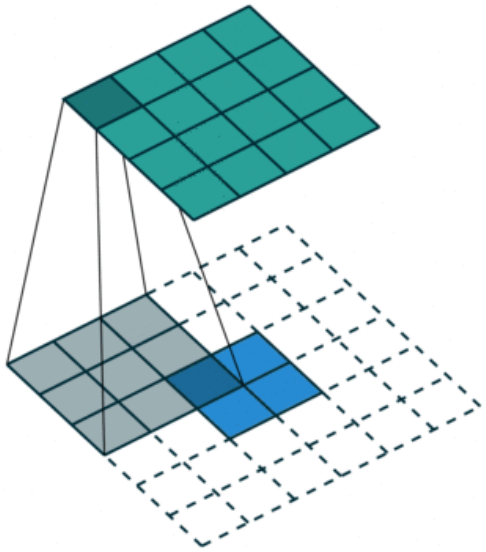
Adapted from Fei-Fei Li et al.

Learnable Upsampling - Transposed Convolution

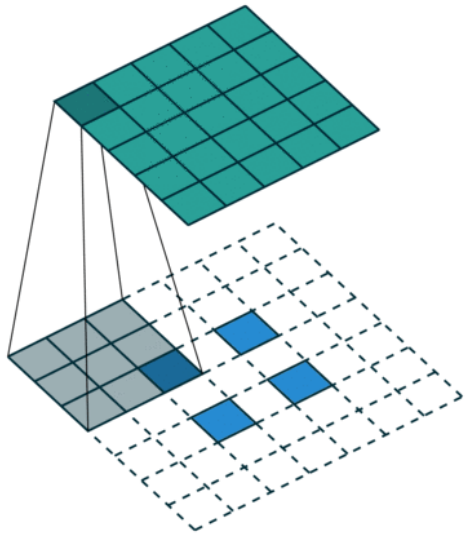
Convolution



Transposed convolution = padding/striding smaller image then weighted sum of input x filter: ‘stamping’ kernel



2x2, stride 1, 3x3 kernel,
upsample to 4x4



2x2, stride 2, 3x3 kernel,
upsample to 5x5.

Image Source: https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d

Kernel

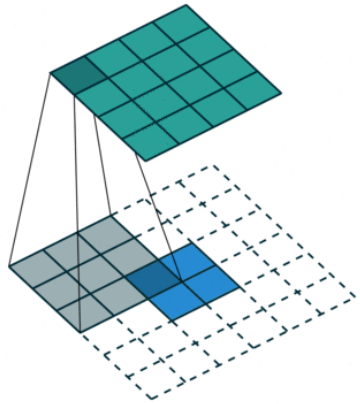
1	1	1
1	1	1
1	1	1

Feature map

1	2
3	4

Padded feature map

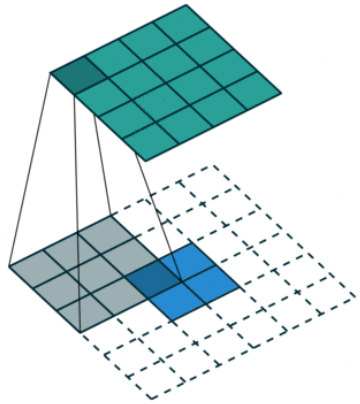
		1	2		
		3	4		



James Tompkin (inspired by andriys)

Kernel

1	1	1
1	1	1
1	1	1



Input feature map

1	2
3	4

Output feature map

1	1	1			
1	1	1			
1	1	1			

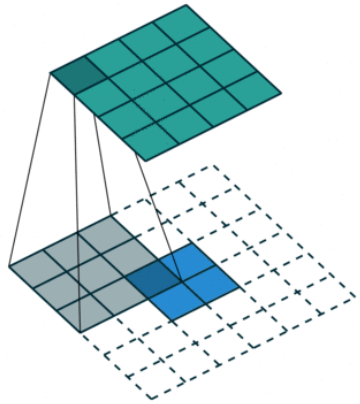
Padded input feature map

		1	2		
		3	4		

James Tompkin (inspired by andriys)

Kernel

1	1	1
1	1	1
1	1	1



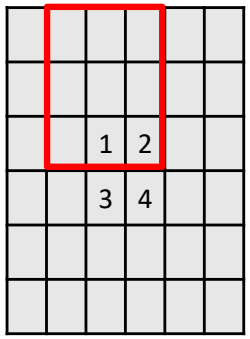
Input feature map

1	2
3	4

Output feature map

1	4	4	3		
1	4	4	3		
1	4	4	3		

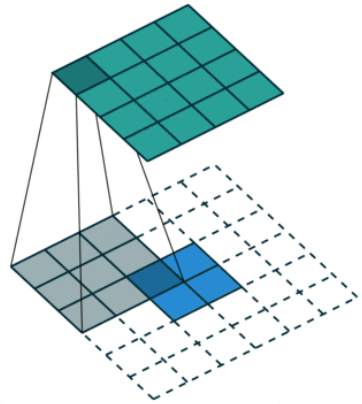
Padded input feature map



James Tompkin (inspired by andriys)

Kernel

1	1	1
1	1	1
1	1	1



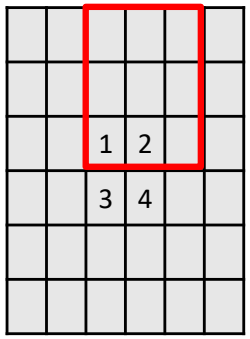
Input feature map

1	2
3	4

Output feature map

1	4	7	6	3	
1	4	7	6	3	
1	4	7	6	3	

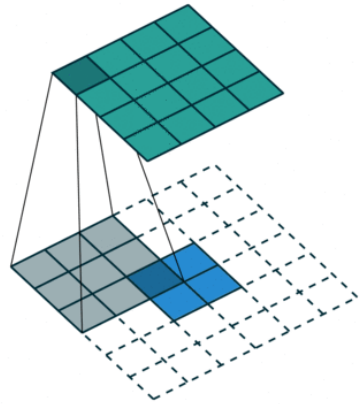
Padded input feature map



James Tompkin (inspired by andriys)

Kernel

1	1	1
1	1	1
1	1	1



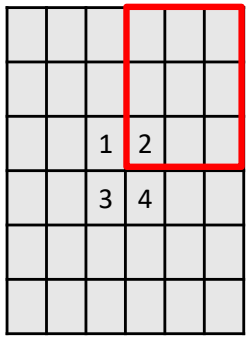
Input feature map

1	2
3	4

Output feature map

1	4	7	8	5	2
1	4	7	8	5	2
1	4	7	8	5	2

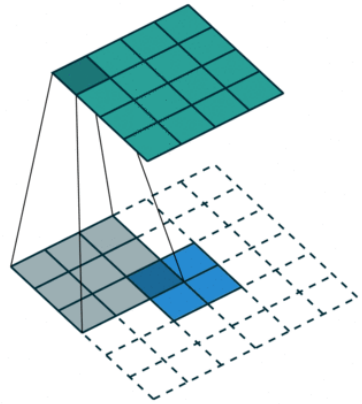
Padded input feature map



James Tompkin (inspired by andriys)

Kernel

1	1	1
1	1	1
1	1	1



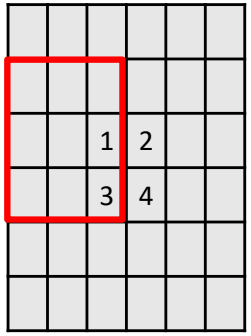
Input feature map

1	2
3	4

Output feature map

1	4	7	8	5	2
5	8	11	8	5	2
5	8	11	8	5	2
4	4	4			

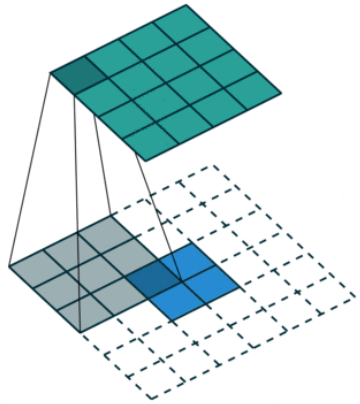
Padded input feature map



James Tompkin (inspired by andriys)

Kernel

1	1	1
1	1	1
1	1	1



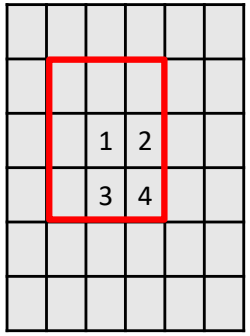
Input feature map

1	2
3	4

Output feature map

1	4	7	8	5	2
5	18	21	18	5	2
5	18	21	18	5	2
4	14	14	10		

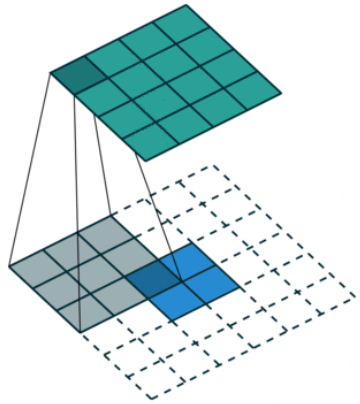
Padded input feature map



James Tompkin (inspired by andriys)

Kernel

1	1	1
1	1	1
1	1	1



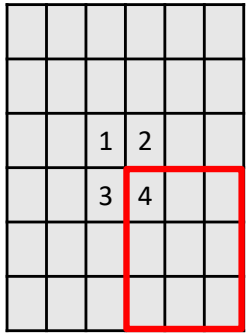
Input feature map

1	2
3	4

Output feature map

1	4	7	8	5	2
5	18	31	34	21	8
9	32	55	60	37	14
11	38	66	64	43	16
7	24	41	44	27	10
3	10	17	18	11	4

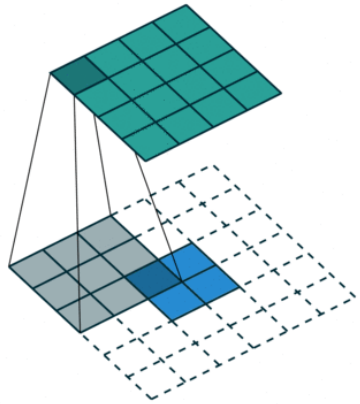
Padded input feature map



James Tompkin (inspired by andriys)

Kernel

1	1	1
1	1	1
1	1	1



Input feature map

1	2
3	4

Cropped output feature map

18	31	34	21
32	55	60	37
38	66	64	43
24	41	44	27

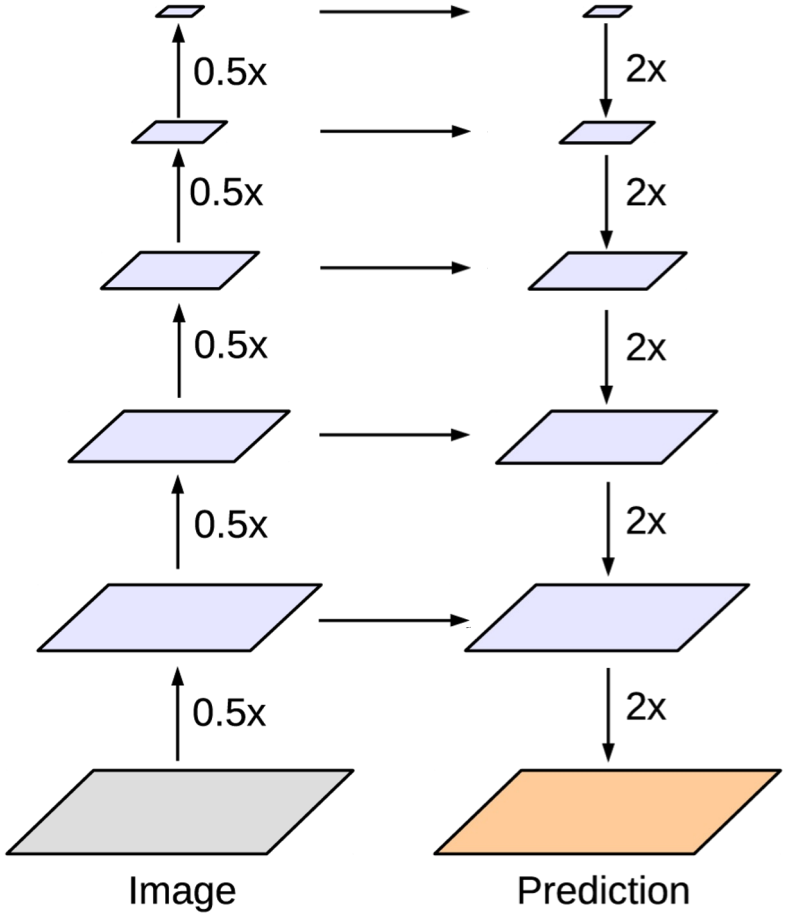
Padded input feature map

		1	2		
		3	4		

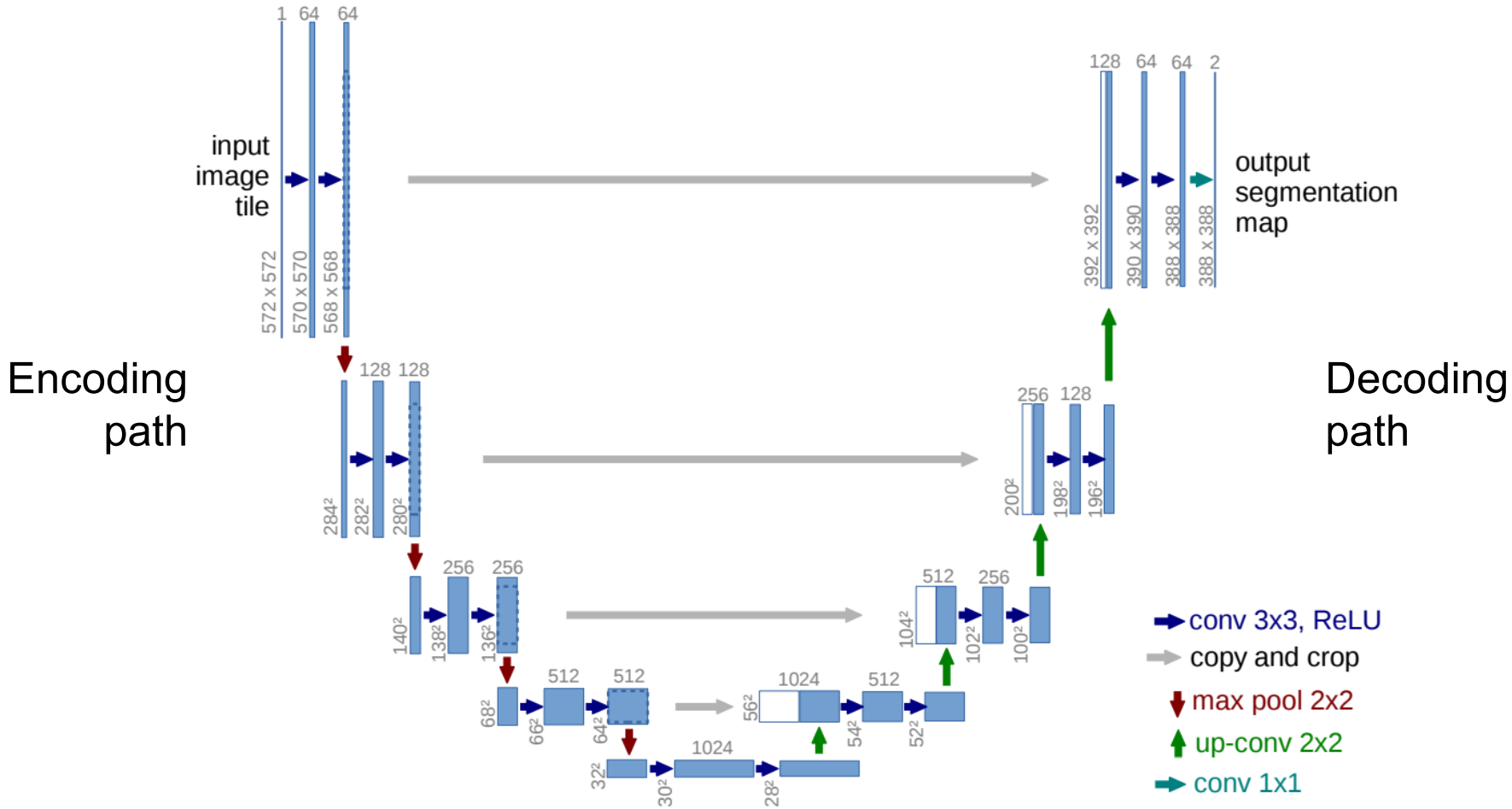
James Tompkin (inspired by andriys)

FCN with Connections

An alternative to the learnable upsampling, is to use **connections** between the encoder and the decoder



U-Net

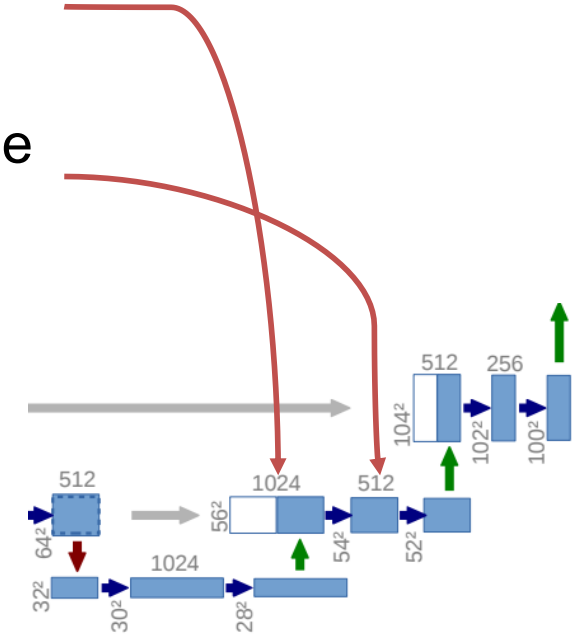


Ronneberger et al., U-net: Convolutional networks for biomedical image segmentation. *MICCAI 2015*
https://arxiv.org/abs/1505.04597

U-Net

Connections are needed to allow **recovering the high resolution** lost in the encoding path

- After upsampling, the symmetrical layer in the encoding path is **concatenated**
- The next convolution in the same level **recovers the fine detail** from the concatenated part

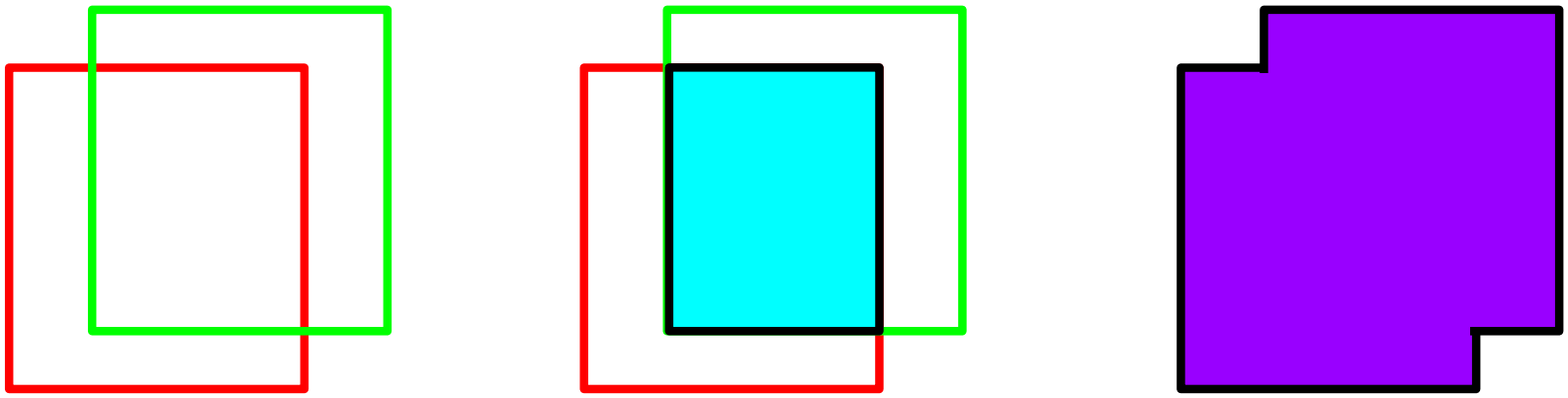


Scoring segmentation

- **Pixel accuracy**
  - Not very useful, due to class imbalance (objects vs. background)
- **Jaccard index** or **Jaccard similarity coefficient**

$$J = IoU$$ Intersection over Union

Intersection: Ground truth ∩ prediction



Union: Ground truth ∪ prediction

Scoring segmentation

- **Dice's coefficient** or **Dice similarity coefficient** (DSC)

$$DSC = \frac{2 * intersection}{total}$$

$$DSC = \frac{2J}{1+J}$$

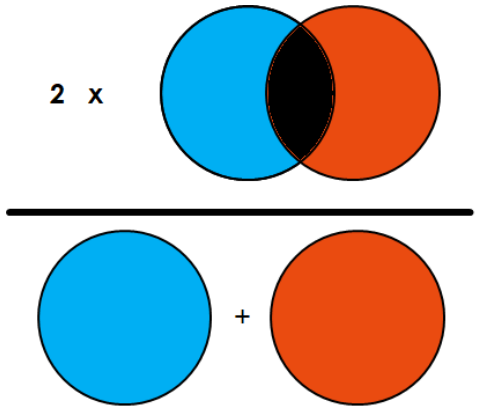
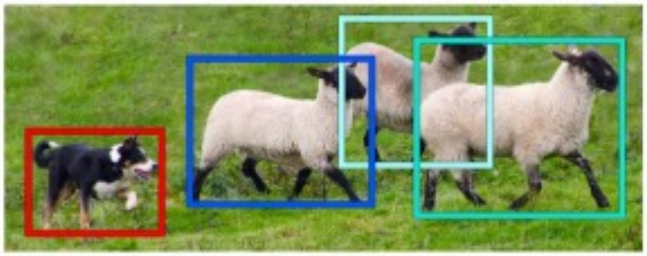


Image source: https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2

Object Detection

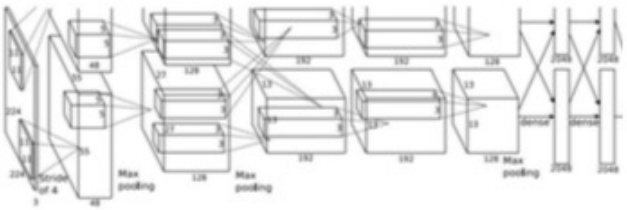
- What about object detection?
  - The goal is to predict a location (defined by a bounding box) of the objects



Object Detection

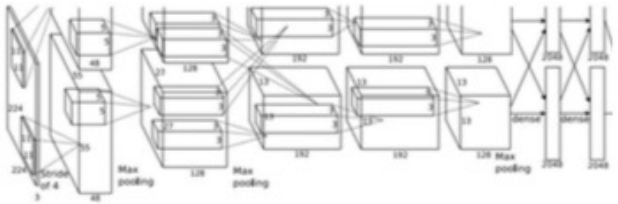
Object Detection as Regression?

Each image needs a different number of outputs



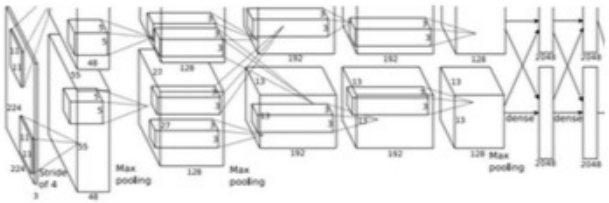
CAT: (x, y, w, h)

4 numbers



DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)

12 numbers

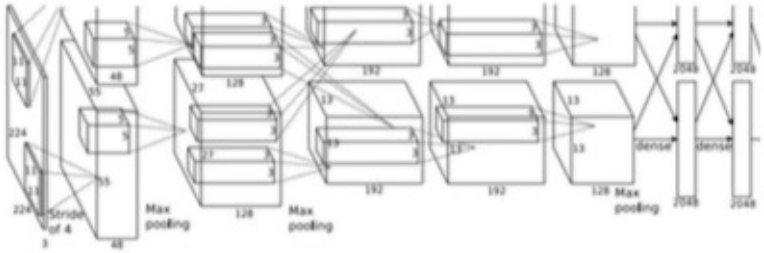


DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
....

? numbers

Sliding Window

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

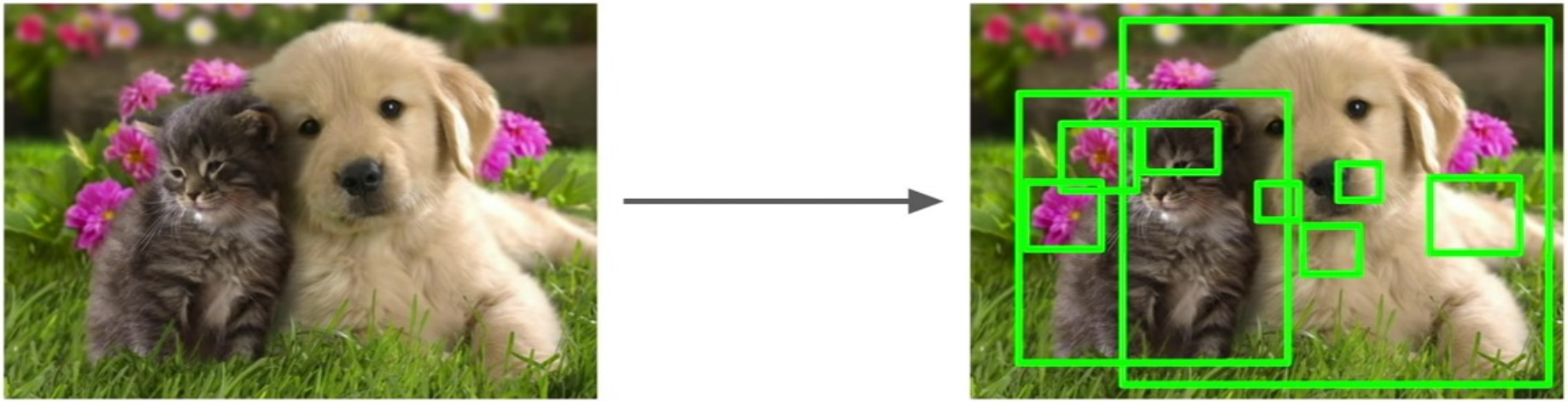


Dog? YES
Cat? NO
Background? NO

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

Region Proposal

- Find image regions that are likely to contain objects, for example using **Selective Search**

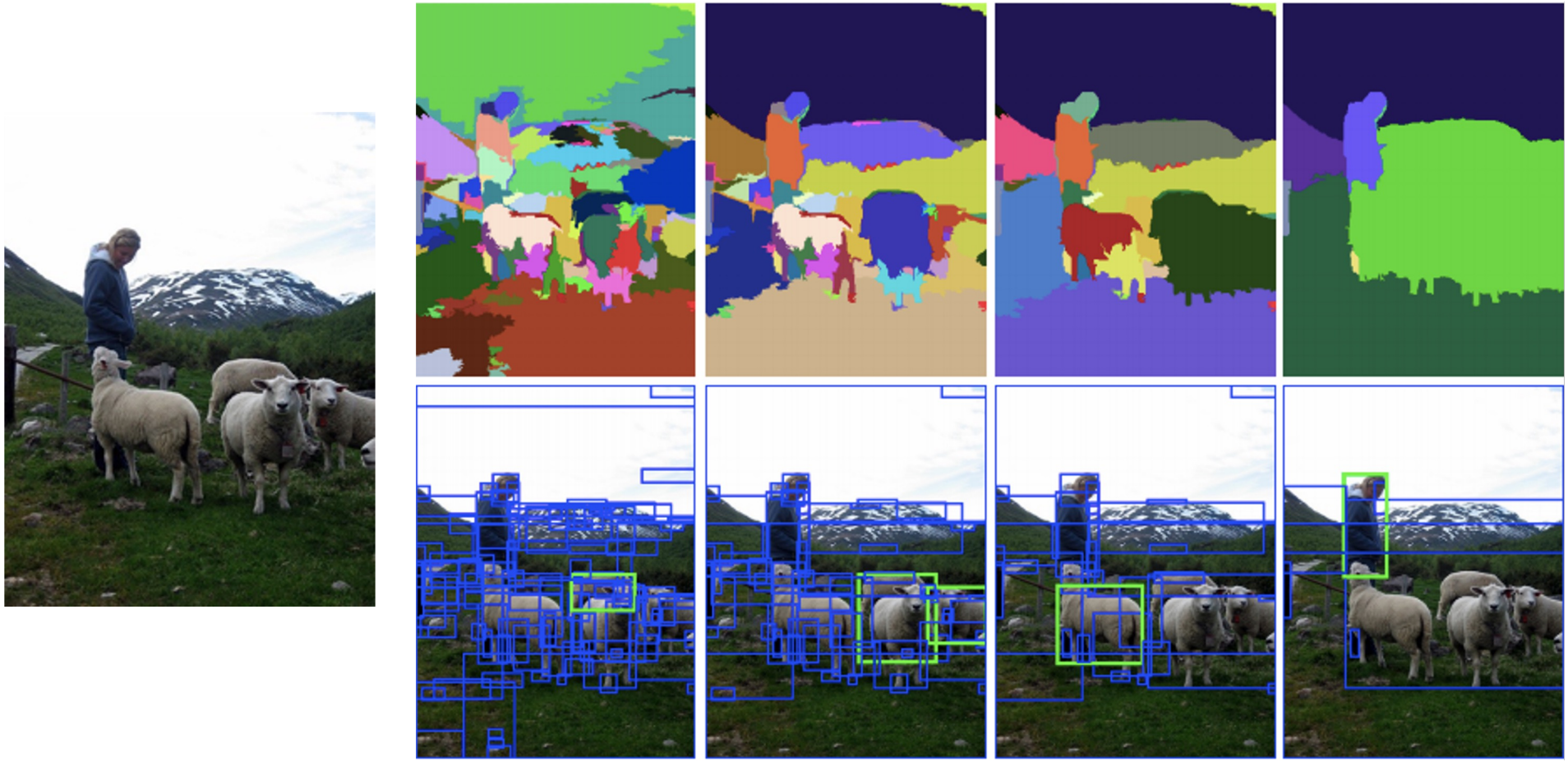


Alexe et al, "Measuring the objectness of image windows", TPAMI 2012
Uijlings et al, "Selective Search for Object Recognition", IJCV 2013
Cheng et al, "BING: Binarized normed gradients for objectness estimation at 300fps", CVPR 2014
Zitnick and Dollar, "Edge boxes: Locating object proposals from edges",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

adapted from Fei-Fei Li et al.

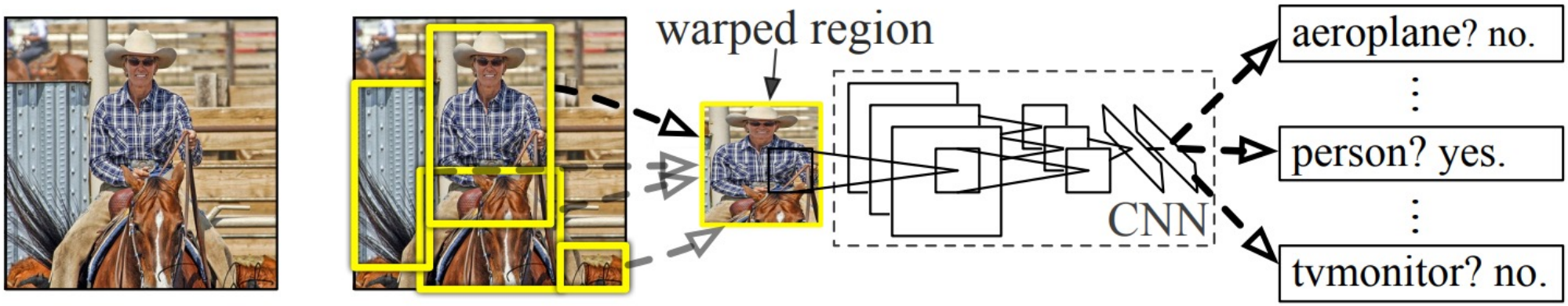
Region Proposal

Regions defined by similarity on color, texture, …



Ali Farhadi, Joseph Redmon

R-CNN: Region-based CNN



1. Input image

2. Extract region proposals (~2k)

3. Compute CNN features

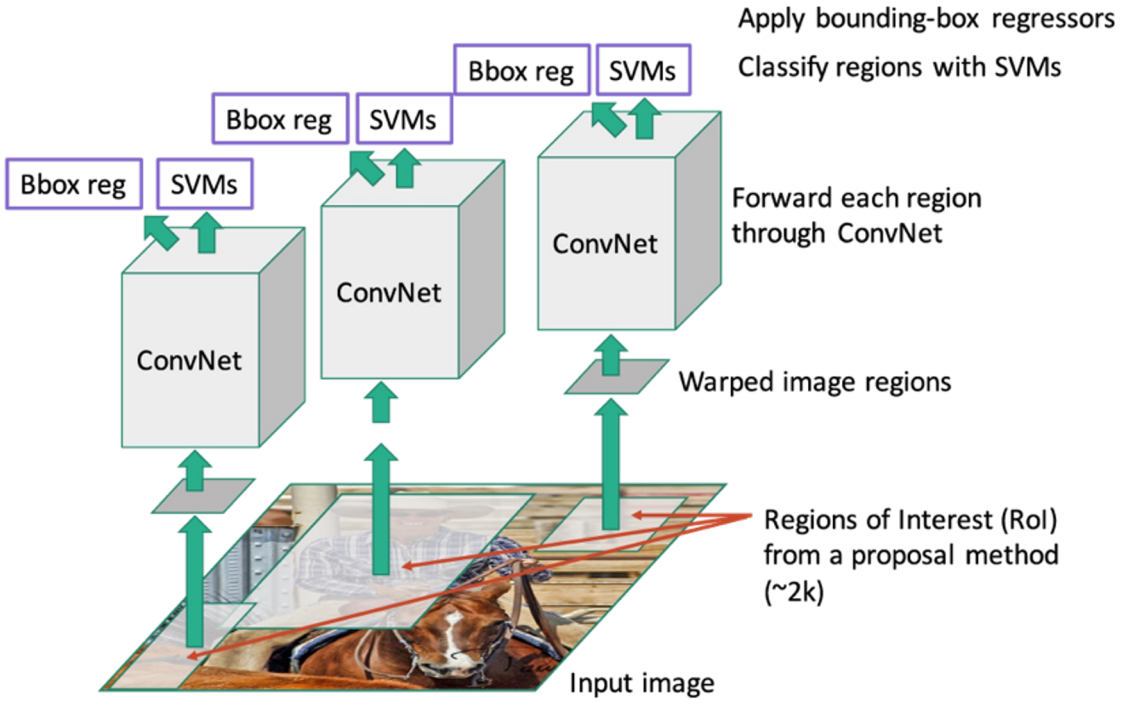
4. Classify regions

+bbox regression

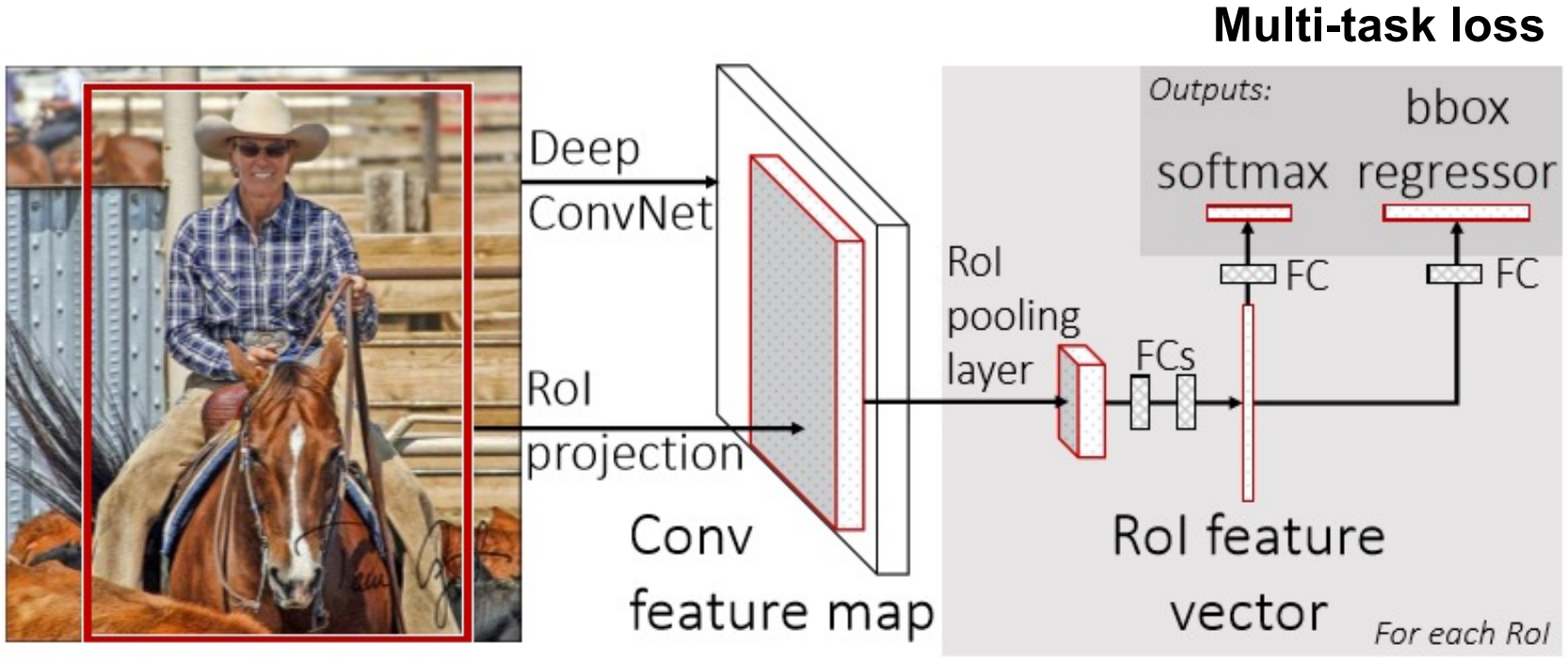
Girshick et al.

R-CNN is slow

Run CNN independently over every ROI (~50s/image)



Fast R-CNN



- Run CNN once, extract features using ROI pooling
- ROI Pool:

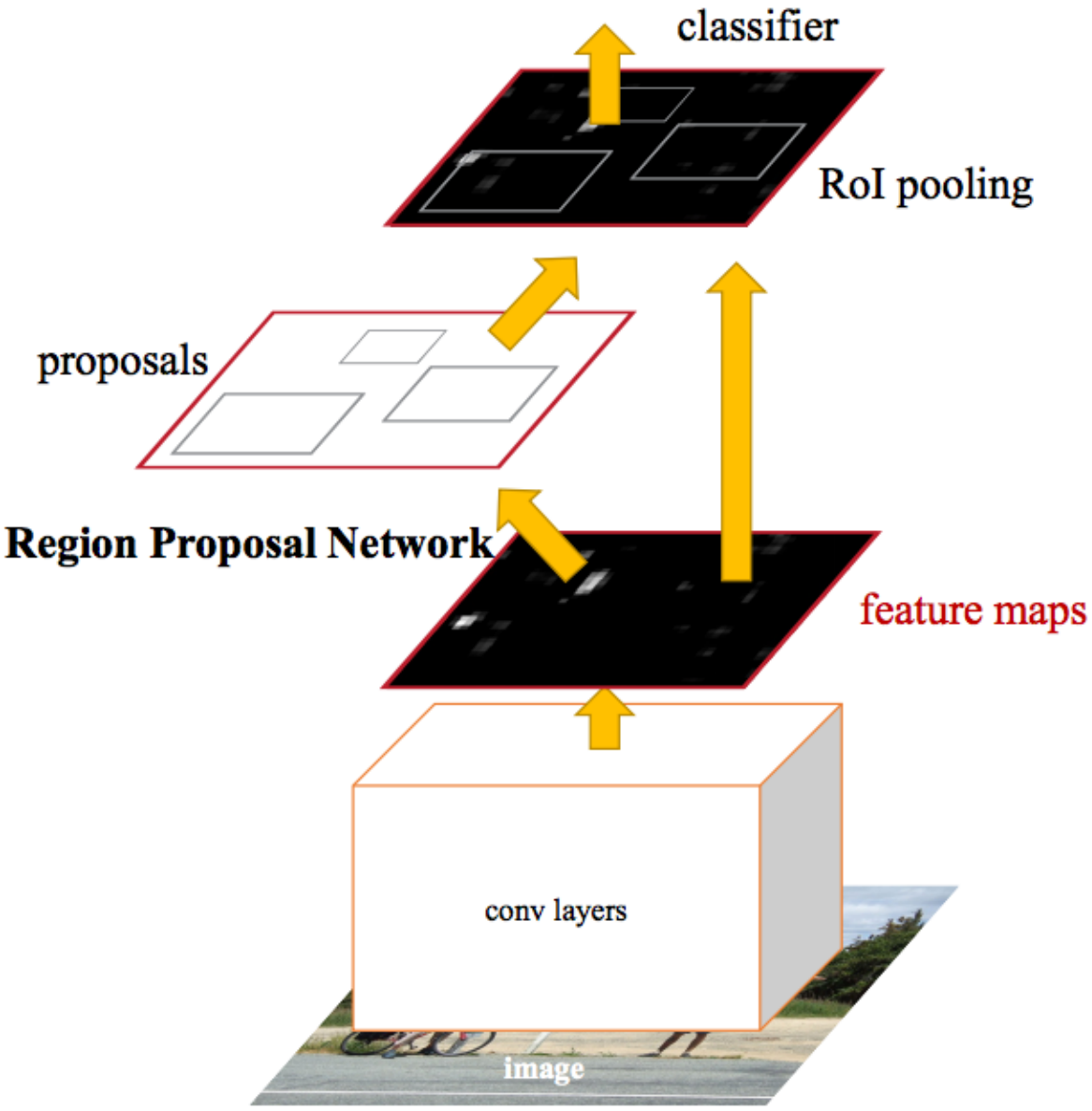
  Convert variable sized ROI to fixed size output
- Much faster, no independent network evaluations (except last linear layer)
- Still slow region proposer, selective search takes ~2 sec

Faster R-CNN

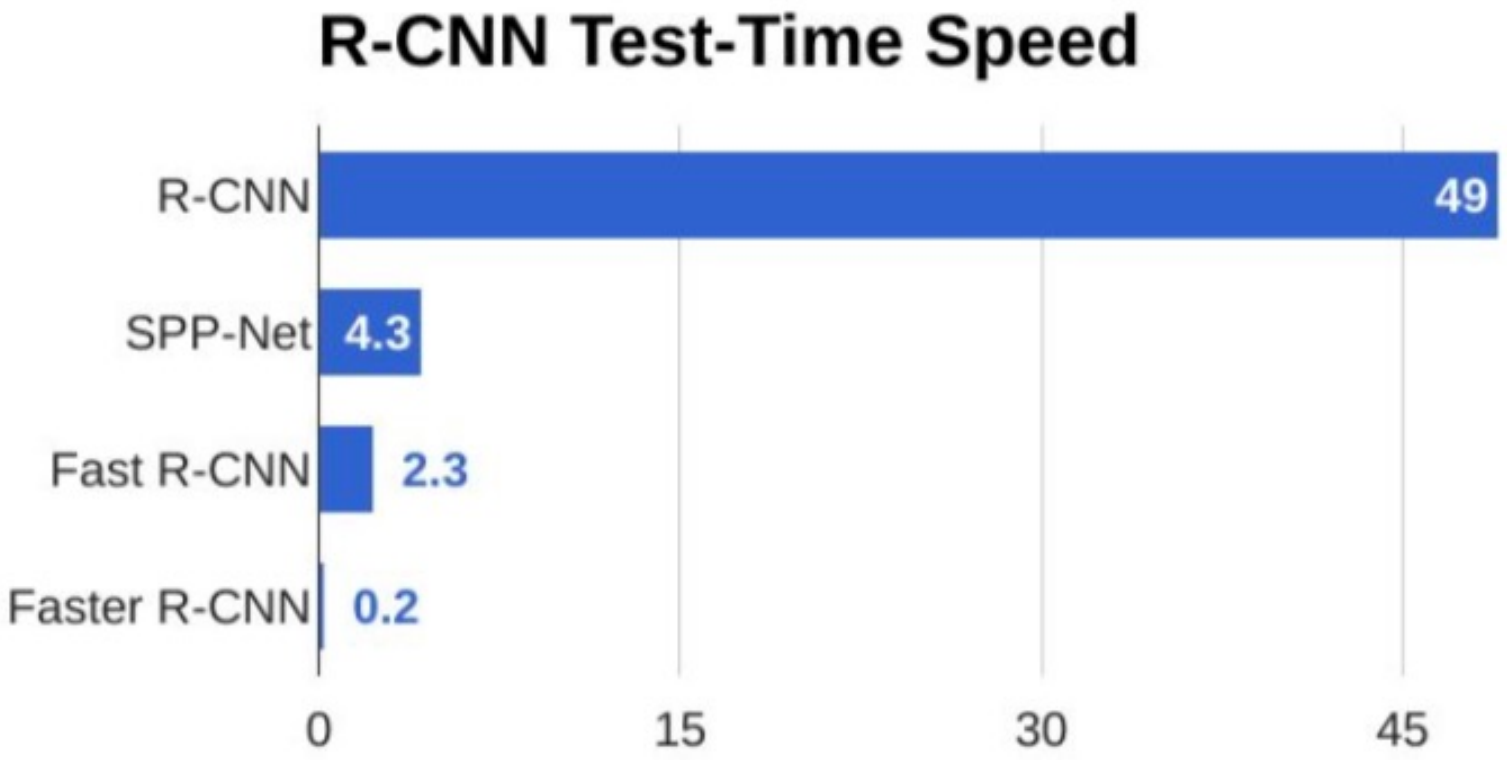
Region Proposal Network uses CNN feature maps.

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

End to end object detection.

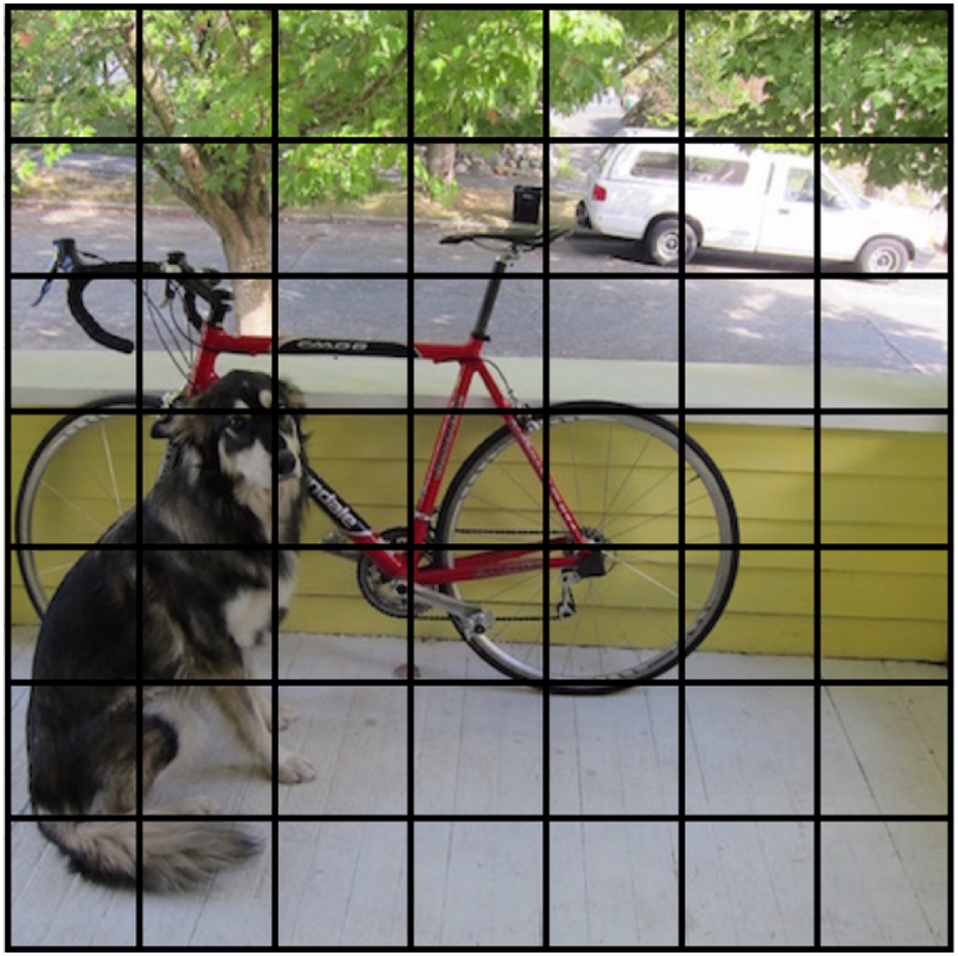


Ren et al. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, 2016
https://arxiv.org/abs/1506.01497


R-CNN Test-Time Speed
R-CNN
SPP-Net
Fast R-CNN
Faster R-CNN
49
4.3
2.3
0.2
0
15
30
45

YOLO (You Only Look Once)

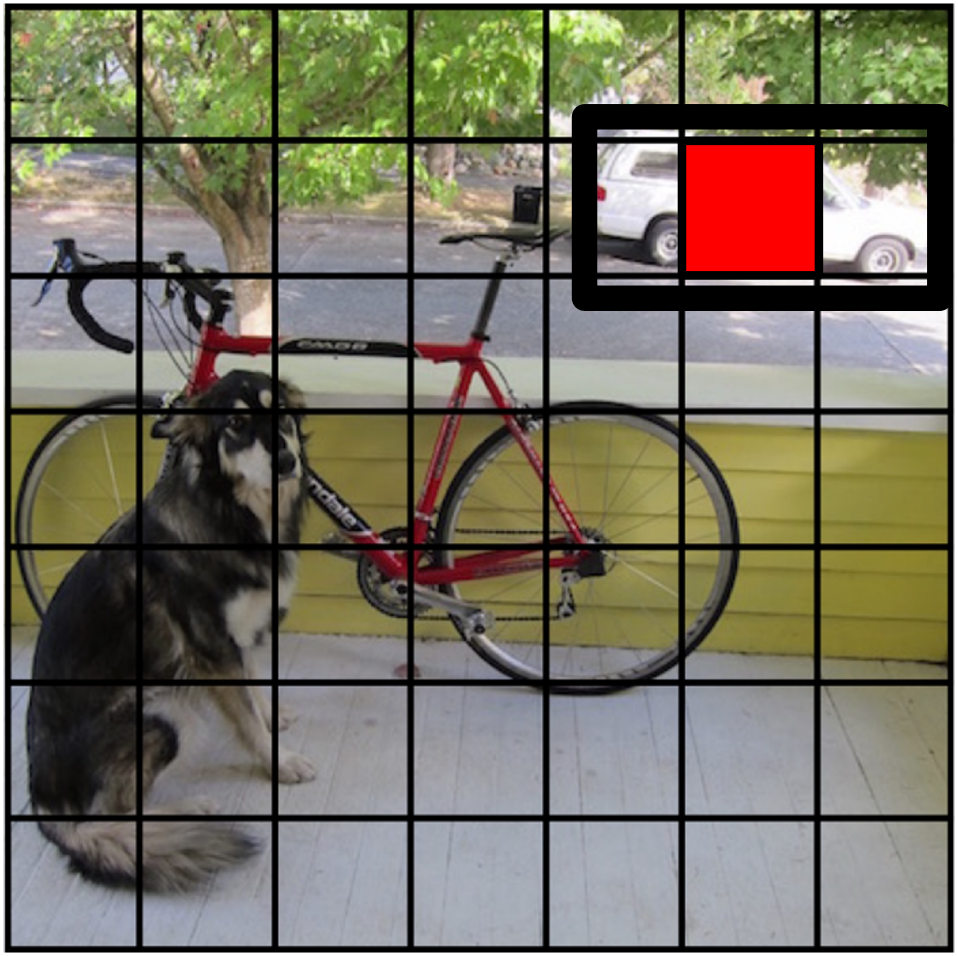
Split image into a grid



Ali Farhadi, Joseph Redmon

YOLO

Each cell predicts B bboxes(x,y,w,h) and confidences of each box: P(Object)



Ali Farhadi, Joseph Redmon

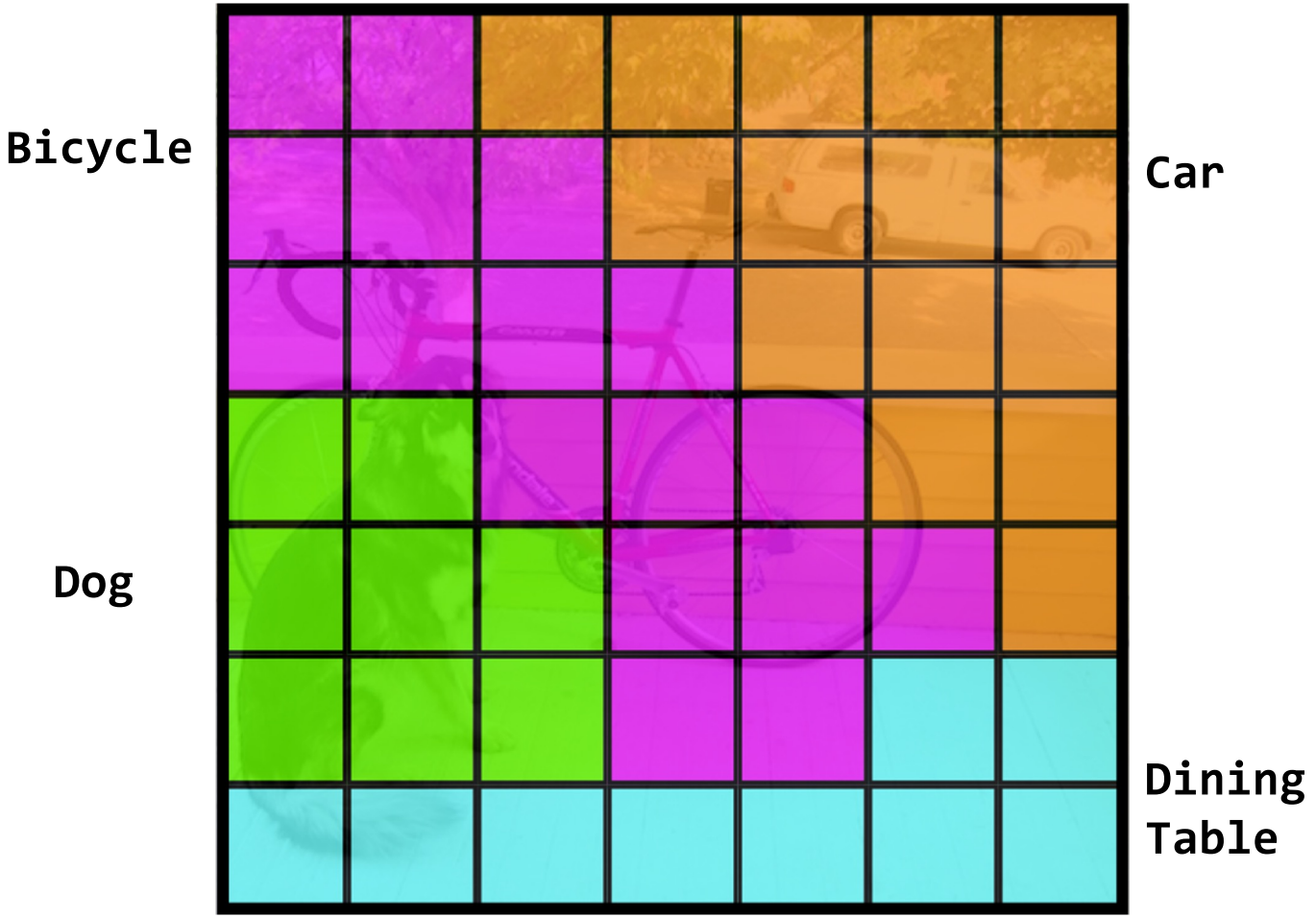
YOLO

Each cell predicts B bboxes(x,y,w,h) and confidences of each box: P(Object)

Ali Farhadi, Joseph Redmon

YOLO

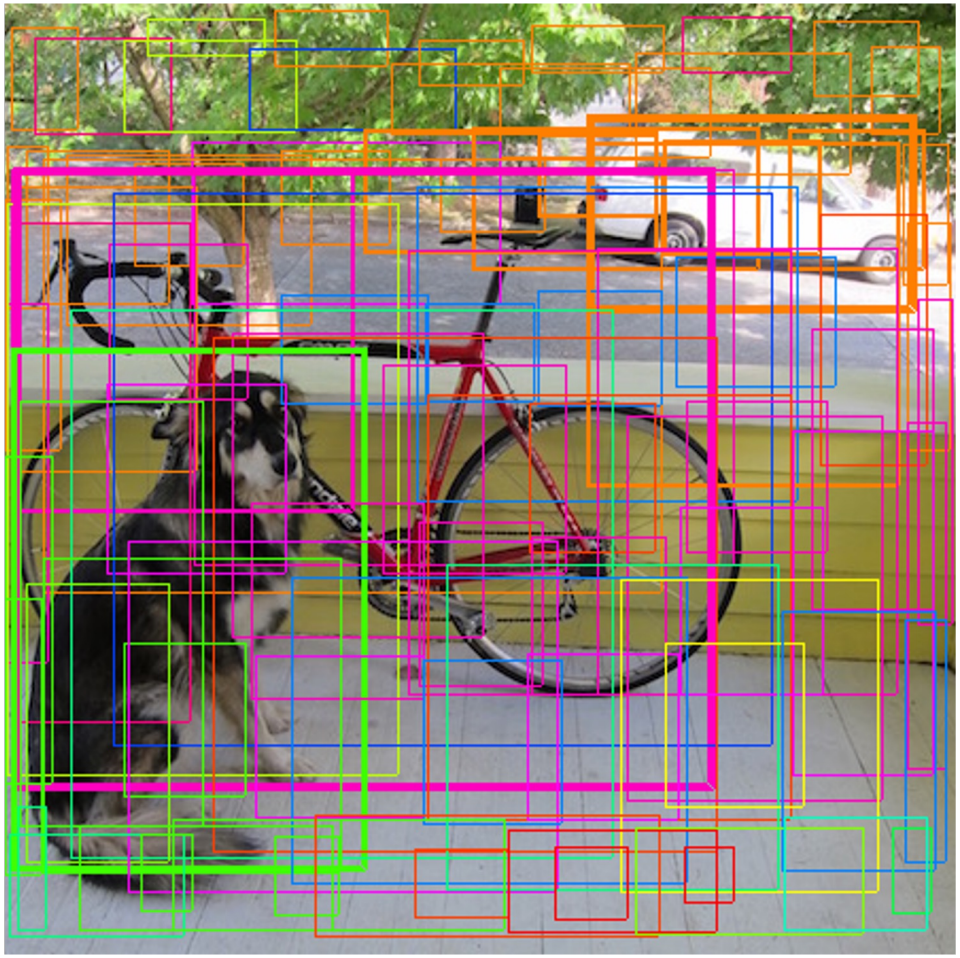
Each cell also predicts a class probability



Ali Farhadi, Joseph Redmon

YOLO

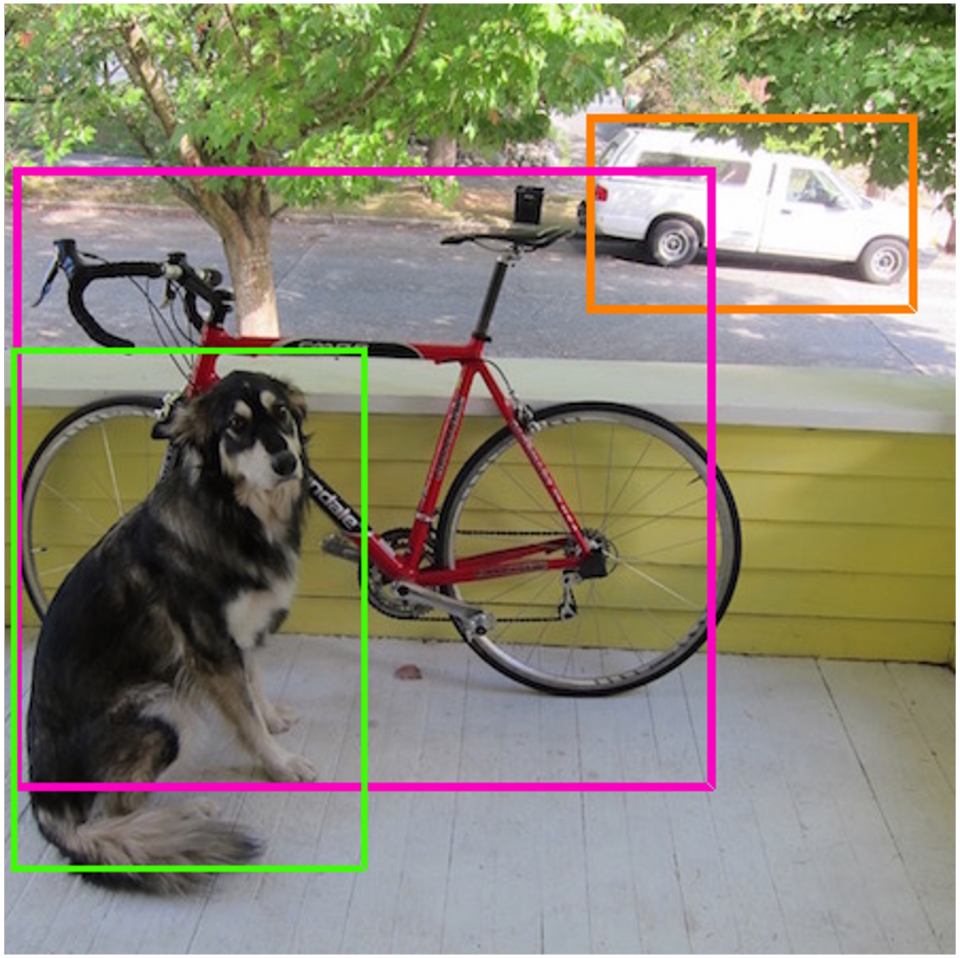
Then combine the bboxes and class predictions



Ali Farhadi, Joseph Redmon

YOLO

Finally, do threshold and non-maximum suppression

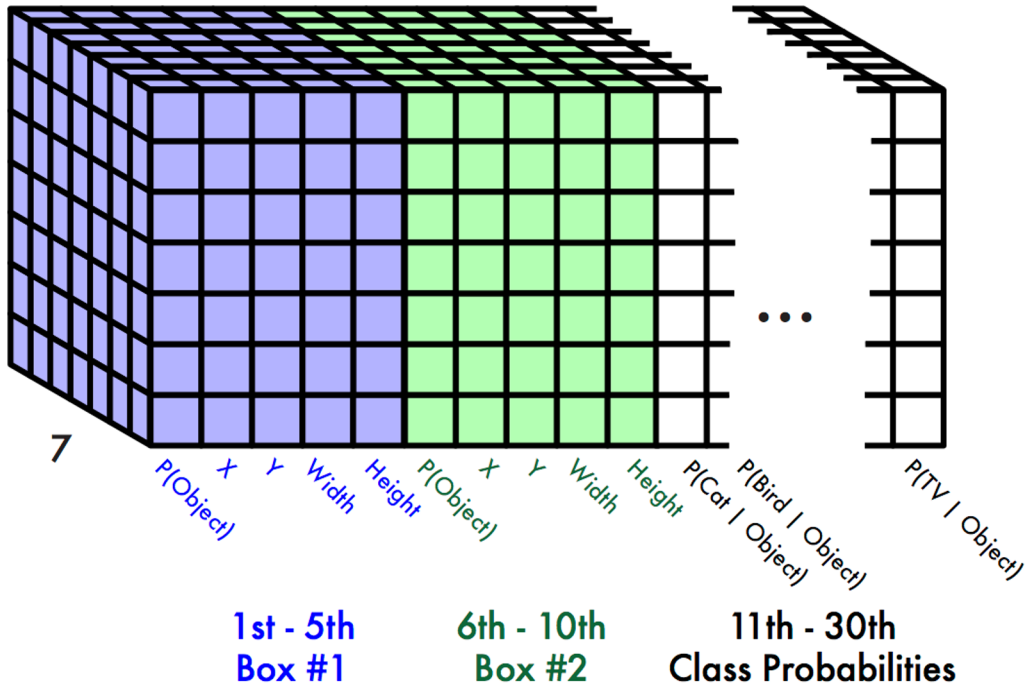


Ali Farhadi, Joseph Redmon

YOLO

Each cell predicts:

- For each bounding box:
  - 4 coordinates (x, y, w, h)
  - 1 confidence value
- Some number of class probabilities

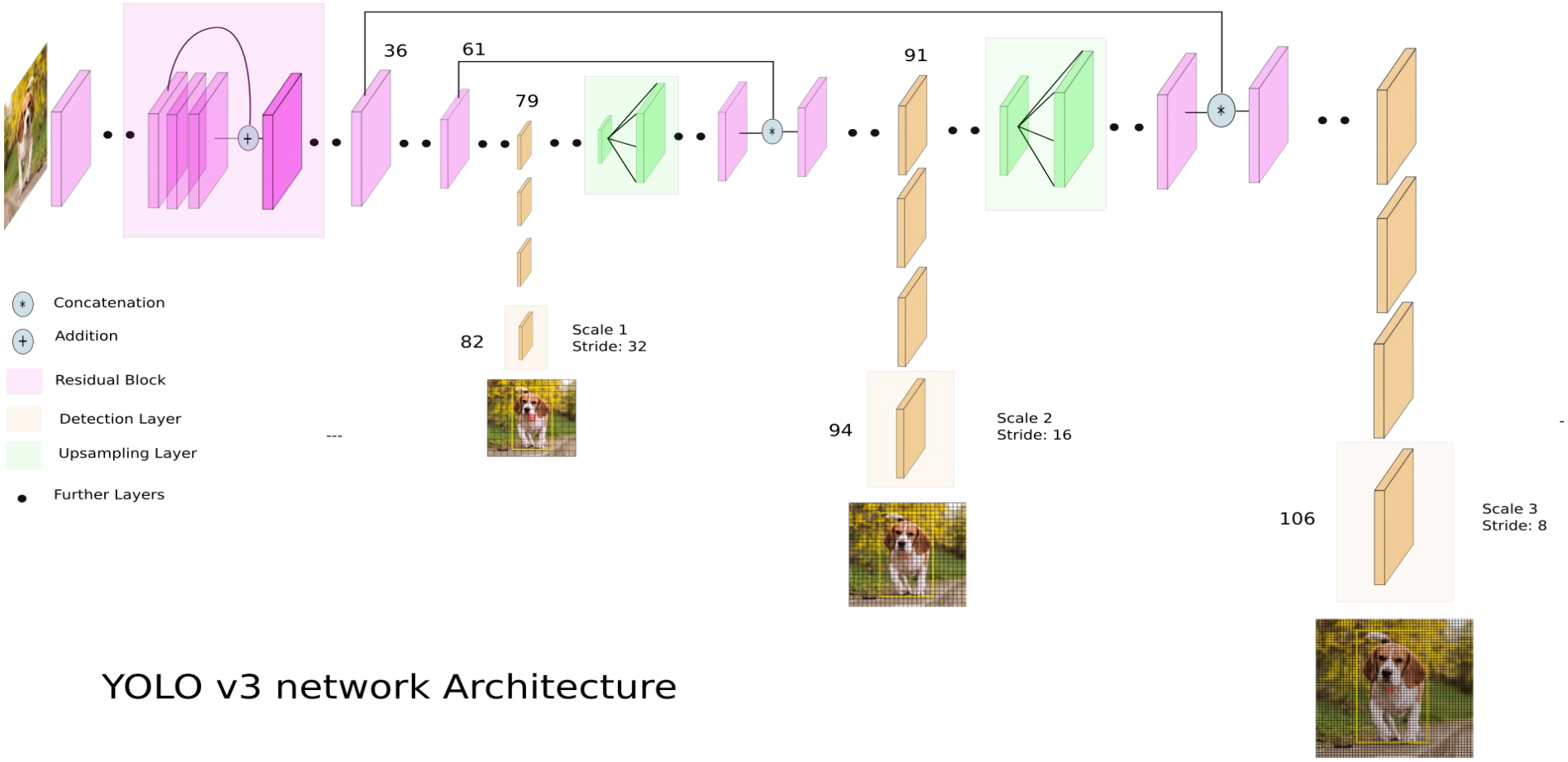


For Pascal VOC dataset:

- 7x7 grid
- 2 bounding boxes / cell
- 20 classes

7 x 7 x (2 x 5 + 20) = 7 x 7 x 30 tensor = **1470 outputs**

YOLOv3



YOLO v3 network Architecture

Recent updates by different authors:
YOLOv4 (April 2020) https://arxiv.org/abs/2004.10934
YOLOv5 (June 2020)

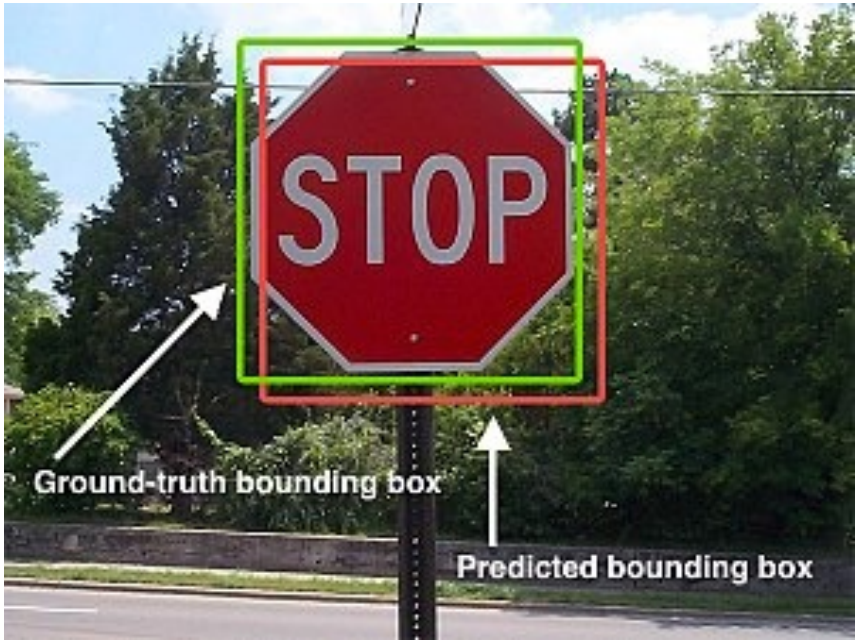
Image source: https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b

	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
YOLO	63.4	45 FPS	22 ms/img
YOLOv2	78.6	40 FPS	25 ms/img
YOLOv3	83.1	30 FPS	33 ms/img
Fast R-CNN	70.0	.5 FPS	2 s/img
Faster R-CNN	73.2	7 FPS	140 ms/img
R-FCN	83.6	6 FPS	170 ms/img

YOLOv4 (April 2020) https://arxiv.org/abs/2004.10934

Ali Farhadi, Joseph Redmon

Scoring object detection



Correct object detection?

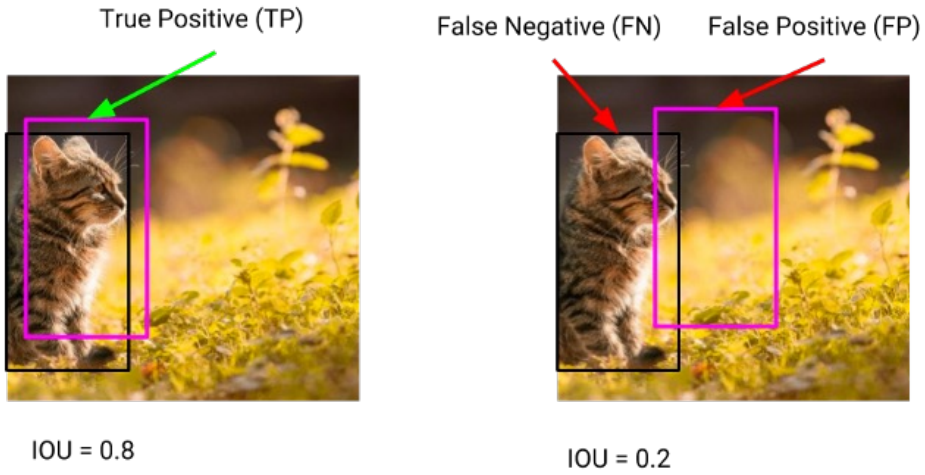
A measure of similarity is needed. If this measure is larger than a **threshold**, we consider it correct

Image source: https://commons.wikimedia.org/wiki/File:Intersection_over_Union_-_object_detection_bounding_boxes.jpg

Scoring object detection

"Correct" bounding box:

IoU = Intersection / Union > 0.5 (or a different threshold)



Typically, mean average precision (mAP) is used

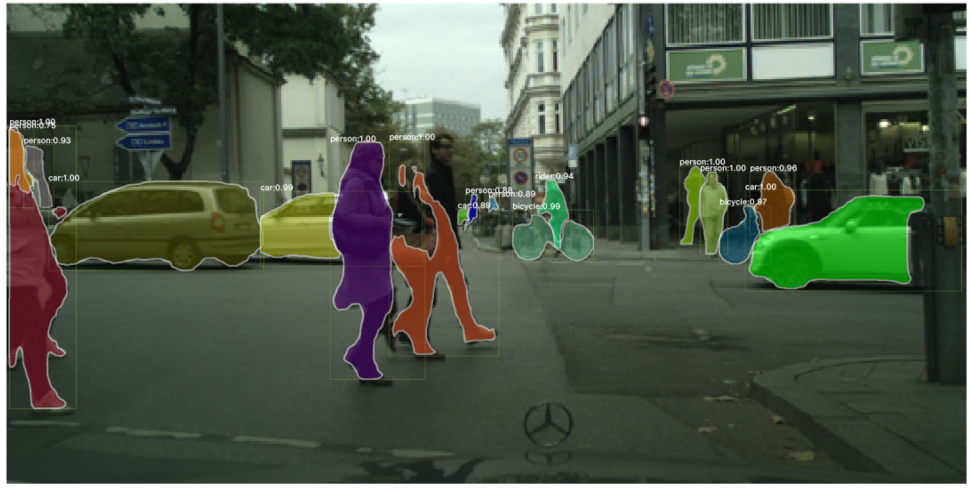
- AP is the area under the PR curve for different thresholds
- mAP is the average AP over the different classes

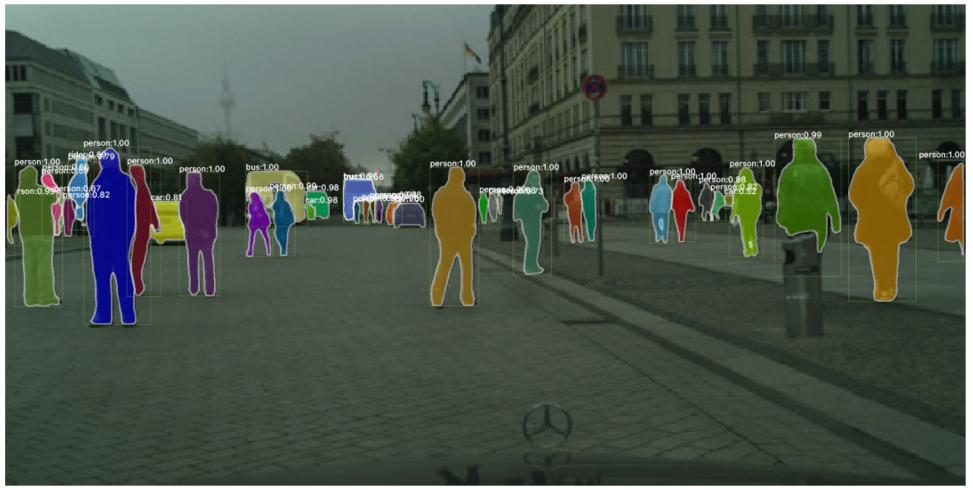
adapted from Ali Farhadi, Joseph Redmon

Instance Segmentation

Given an image produce instance-level segmentation

- Which class does each pixel belong to
- Also which instance

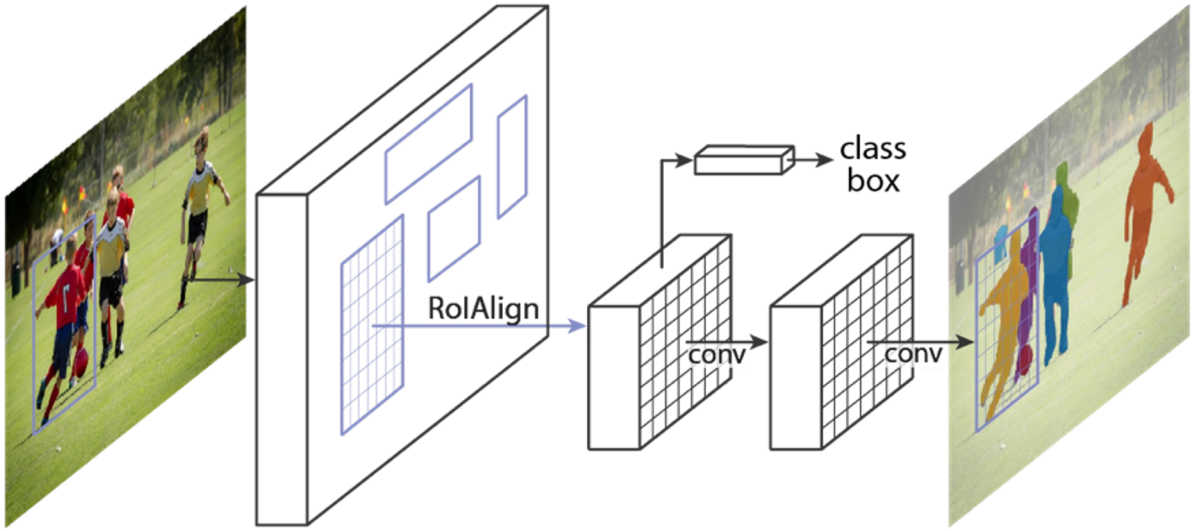




Ali Farhadi, Joseph Redmon

Mask R-CNN

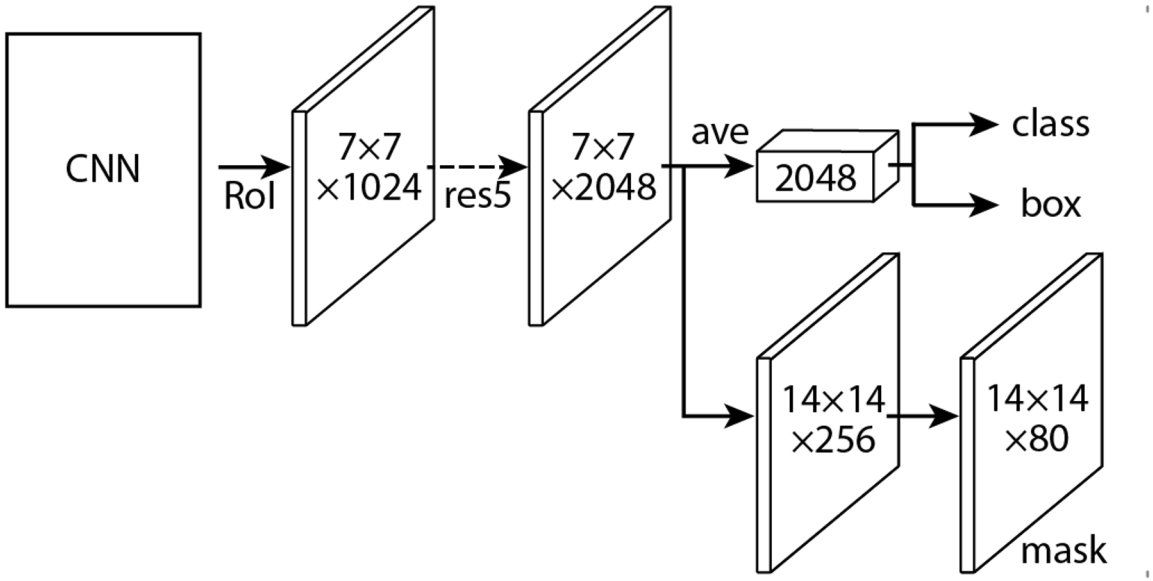
Similar to R-CNN but predicts mask *as well as* bounding box



He et al., Mask R-CNN
https://arxiv.org/pdf/1703.06870.pdf

Mask R-CNN

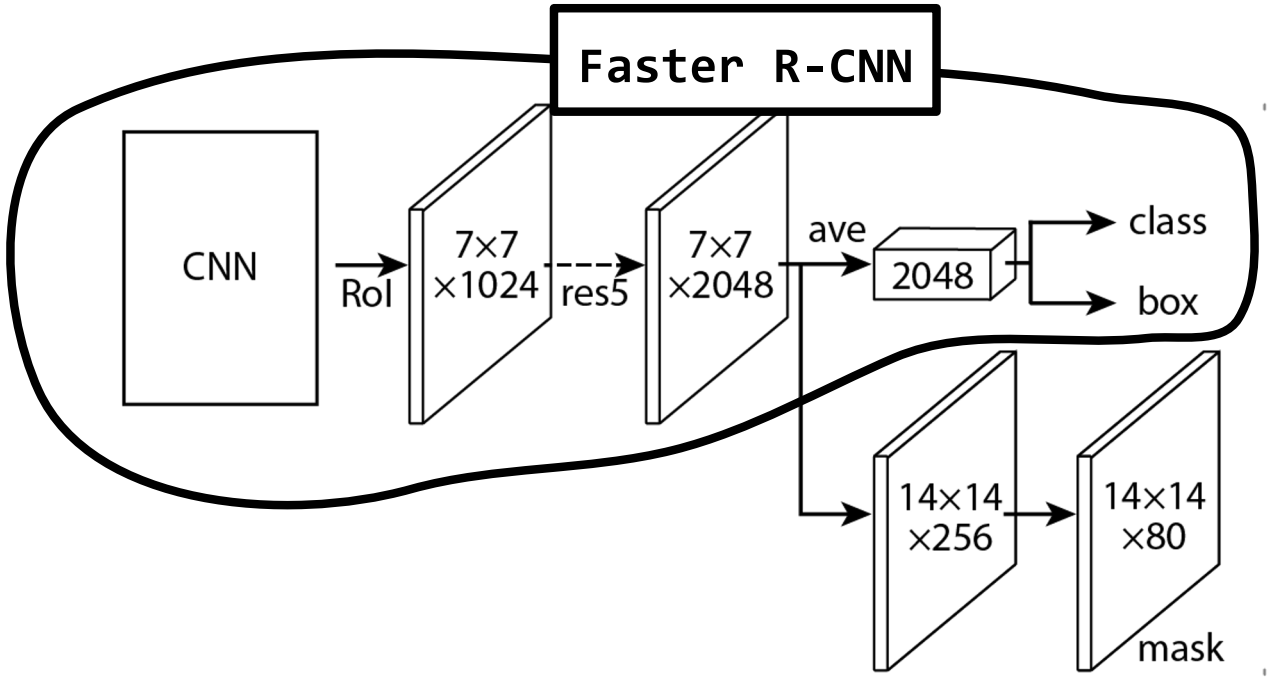
Similar to R-CNN but predicts mask *as well as* bounding box



https://arxiv.org/pdf/1703.06870.pdf

Mask R-CNN

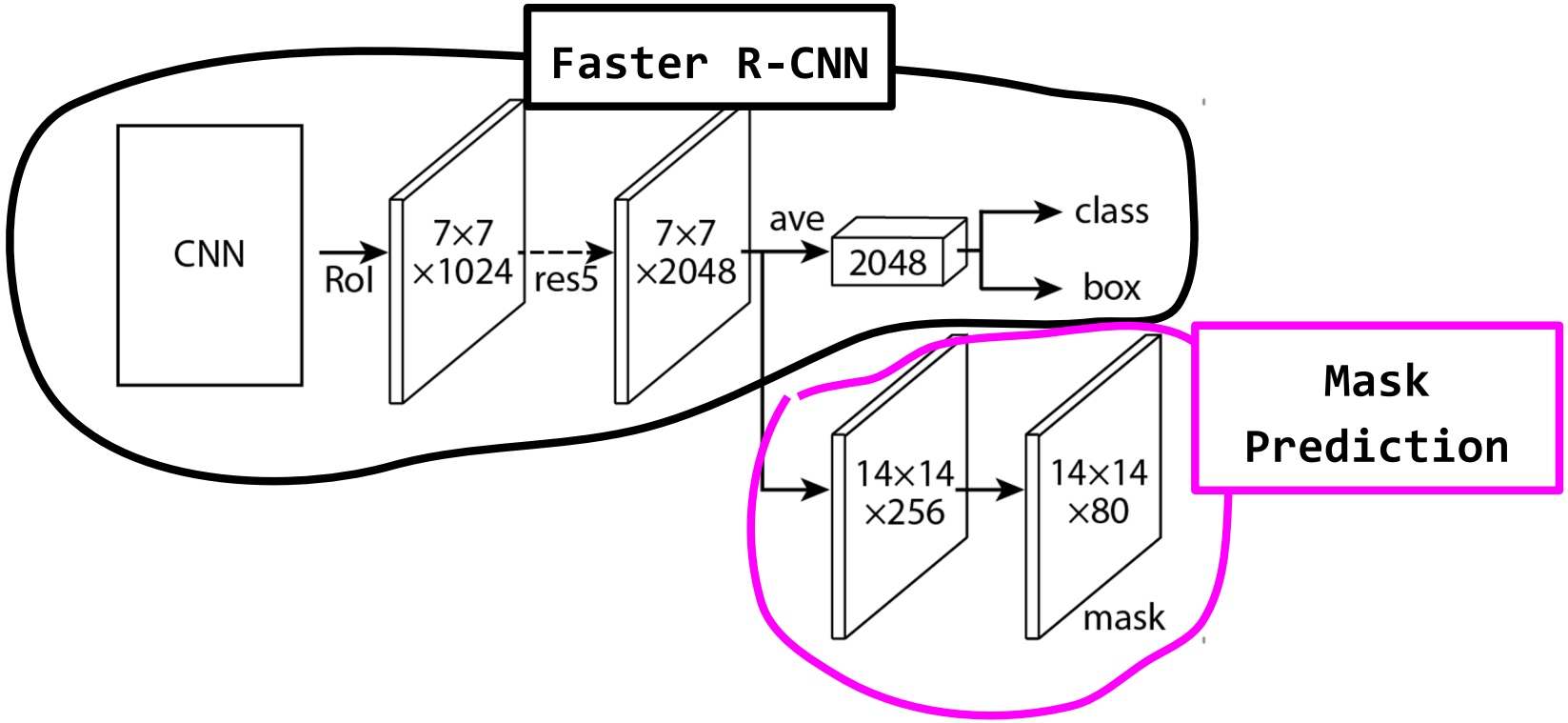
Similar to R-CNN but predicts mask *as well as* bounding box



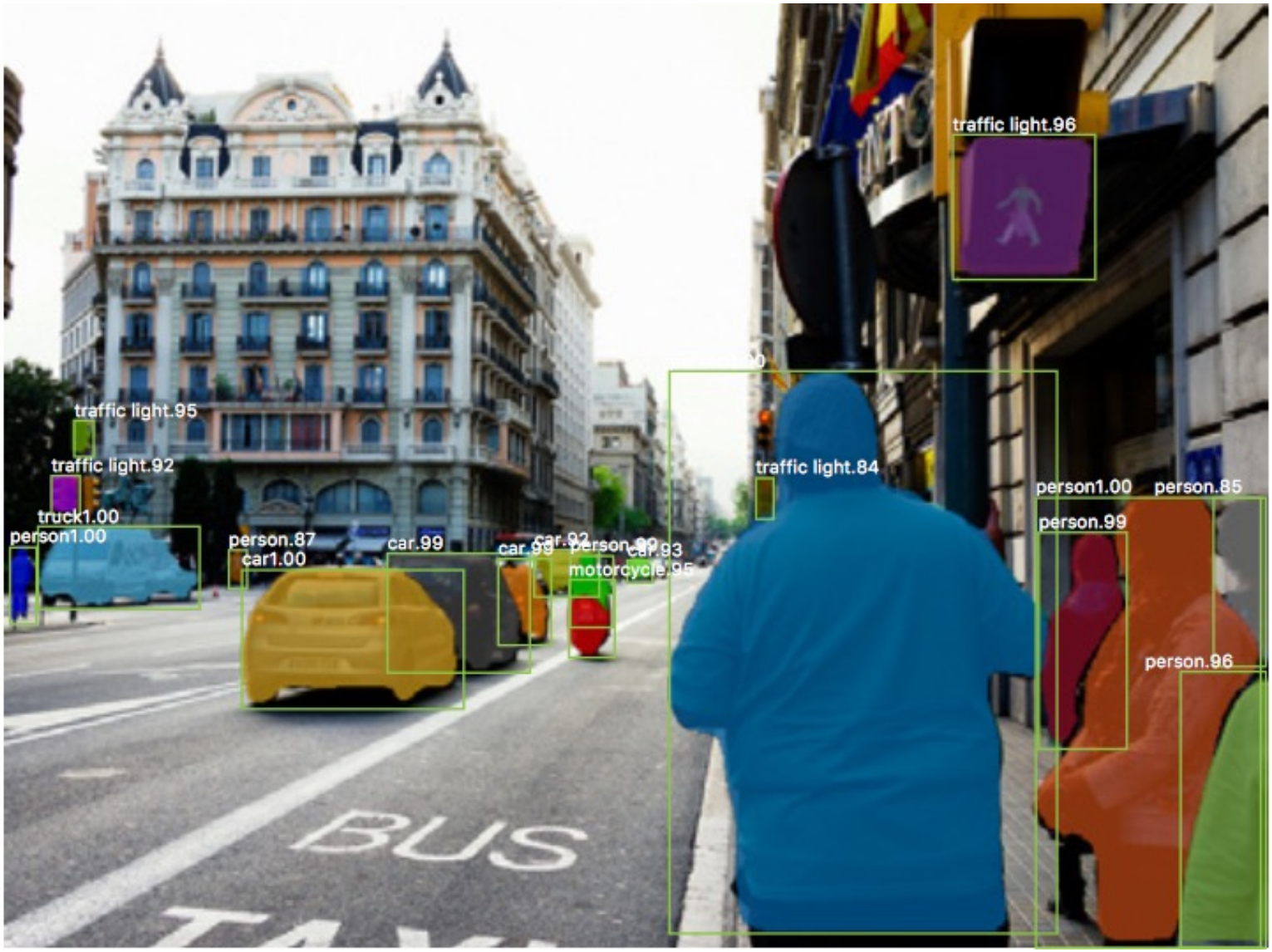
https://arxiv.org/pdf/1703.06870.pdf

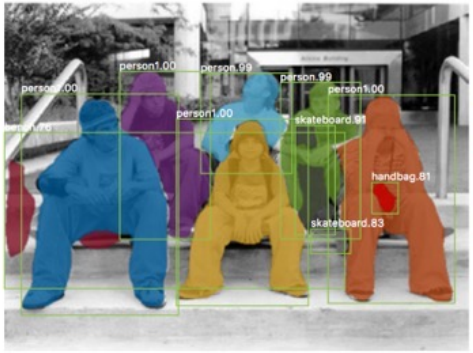
Mask R-CNN

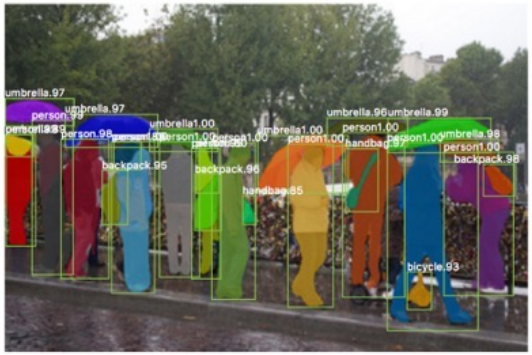
Similar to R-CNN but predicts mask *as well as* bounding box

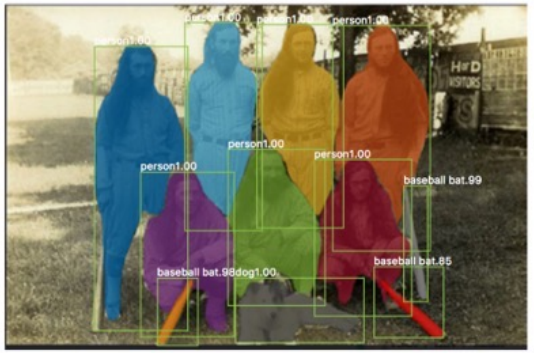


https://arxiv.org/pdf/1703.06870.pdf


traffic light.96
traffic light.95
traffic light.92
truck1.00
person1.00
person.87
car1.00
car.99
car.99
car.92
person.99
car.93
motorcycle.95
traffic light.84
person1.00
person.85
person.99
person.96
BUS

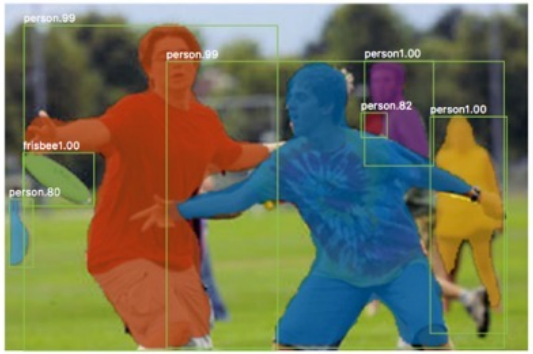

person1.00
person.99
person.99
person1.00
person1.00
skateboard.91
handbag.81
skateboard.83

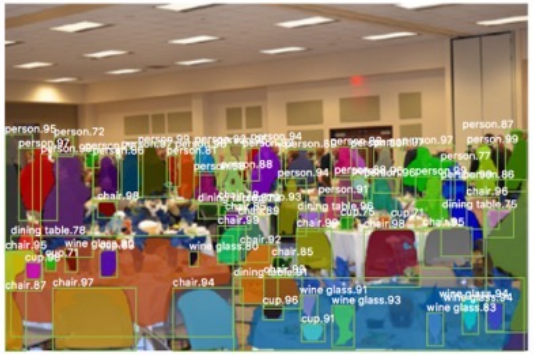

umbrella.97
umbrella.97
umbrella1.00
umbrella.96
umbrella.99
person1.00
backpack.95
backpack.96
handbag.85
person1.00
umbrella.98
backpack.98
bicycle.93


person1.00
person1.00
person1.00
baseball bat.99
baseball bat.98
dog1.00
baseball bat.85


person1.00
person1.00
surfboard1.00
surfboard1.00
surfboard.98
surfboard1.00


zebra.99
zebra.99
zebra.88
zebra1.00
zebra1.00


person.99
person.99
person1.00
person.82
person1.00
frisbee1.00
person.80


person.95
person.72
person.87
person.99
person.77
chair.95
dining table.78
chair.92
chair.85
chair.87
chair.97
chair.94
wine glass.91
wine glass.93
cup.96
cup.91


kite.82
kite.97
kite.99
kite.84
kite.93
person.80
person1.00
handbag.80


elephant1.00
elephant1.00
elephant.99


traffic light.99
traffic light.71
person.99
tie.88
handbag.88
handbag.88
handbag.99
handbag.88


person1.00
person.88
tv.98
person1.00
bottle.97
wine glass.99
dining table.95
wine glass1.00
wine glass1.00


potted plant.92
person1.00
bird.93
handbag.73
bench.97
cell phone.97

Mask R-CNN
Also does pose

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

Common Objects in COntext (COCO)

Dataset

80 objects
117,261 train/val images
902,435 object instances

New detection metric, mAP averaged over IOU [.5 - .95]

Segmentation masks for each instance



http://cocodataset.org/