

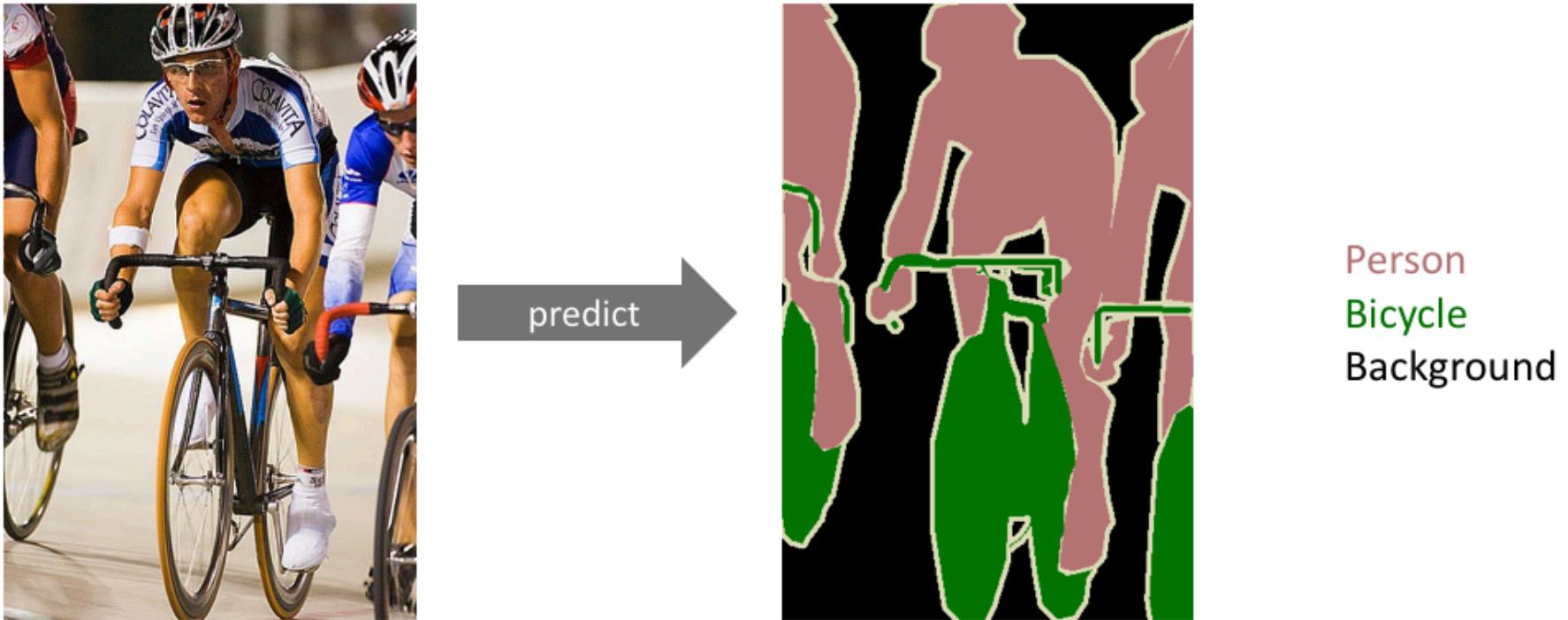
Deep Learning Basics

Lecture 6: Computer Vision Applications (Semantic Segmentation and Detection)

최성준 (고려대학교 인공지능학과)

Semantic Segmentation

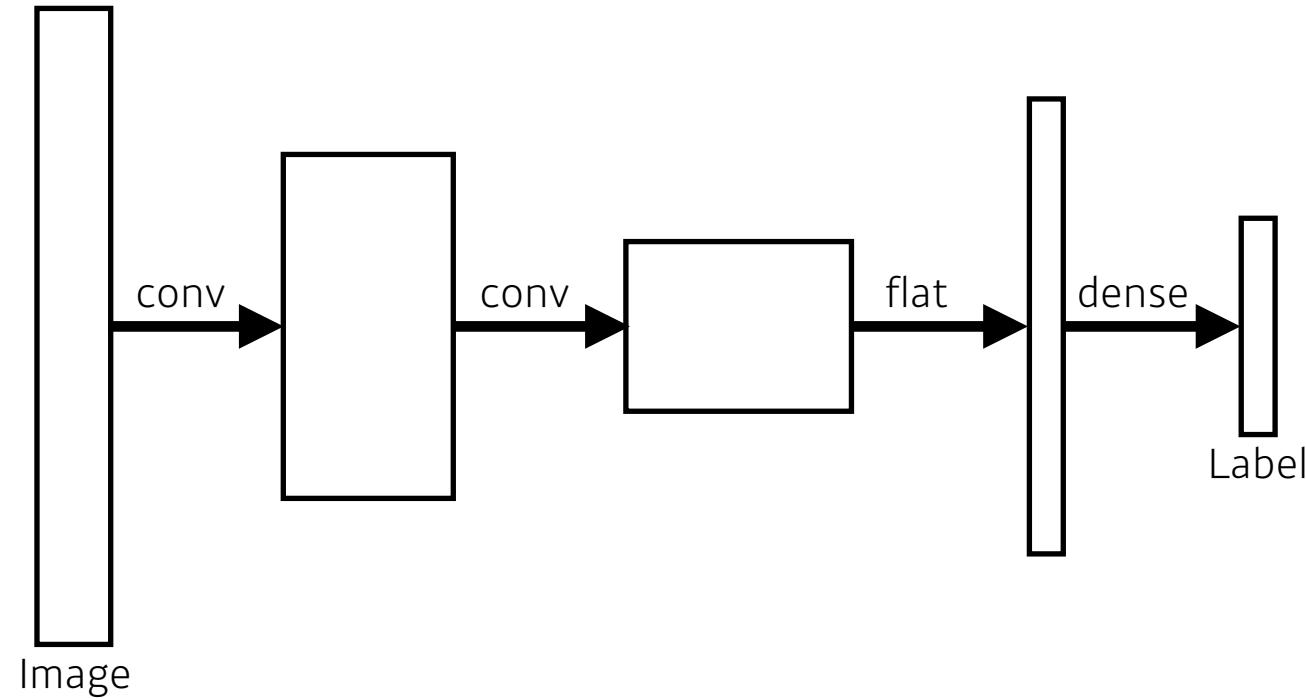
Semantic Segmentation



Semantic Segmentation

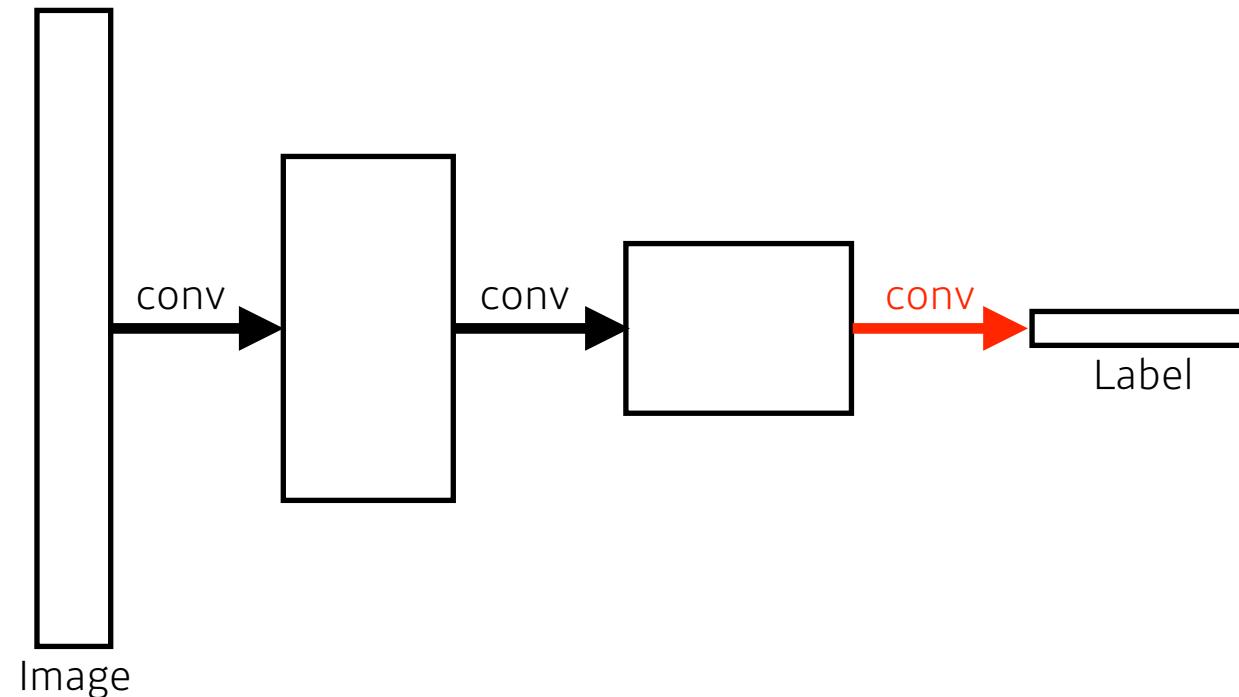


Fully Convolutional Network



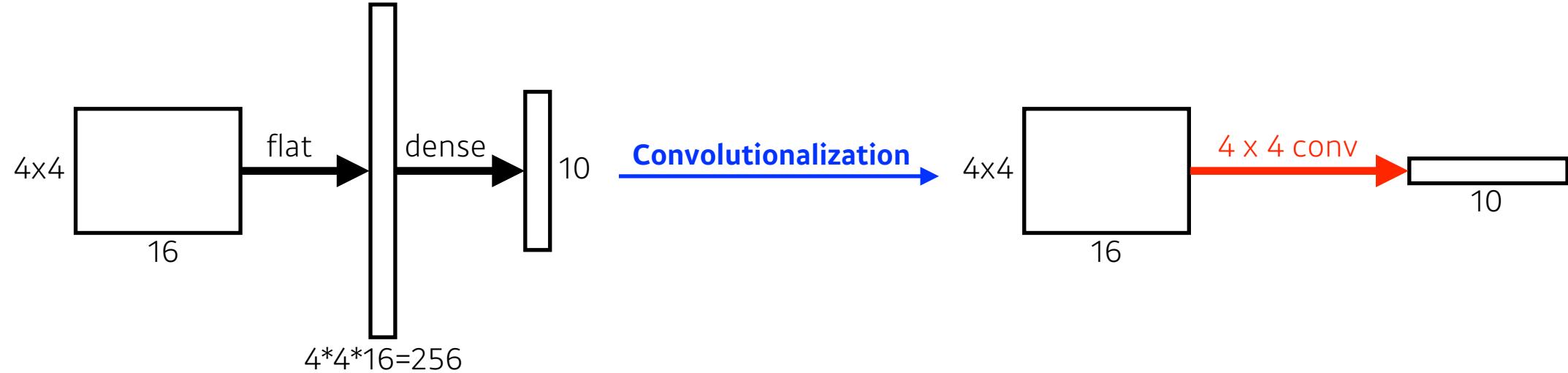
- This is how an ordinary CNN looks like.

Fully Convolutional Network



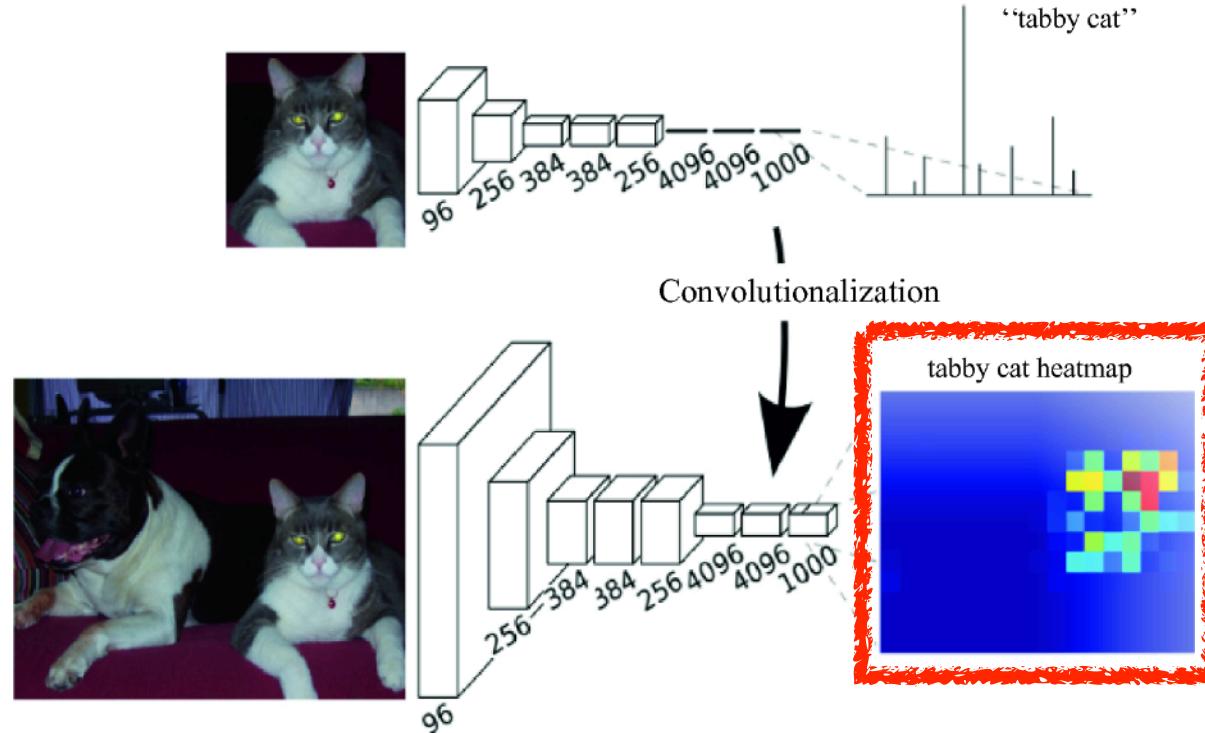
- This is a **fully convolutional** network.

Fully Convolutional Network



- ➊ # of parameters
 - ➊ Left: $4 \times 4 \times 16 \times 10 = 2,560$
 - ➋ Right: $4 \times 4 \times 16 \times 10 = 2,560$
- ➌ This process is called **convolutionalization**.

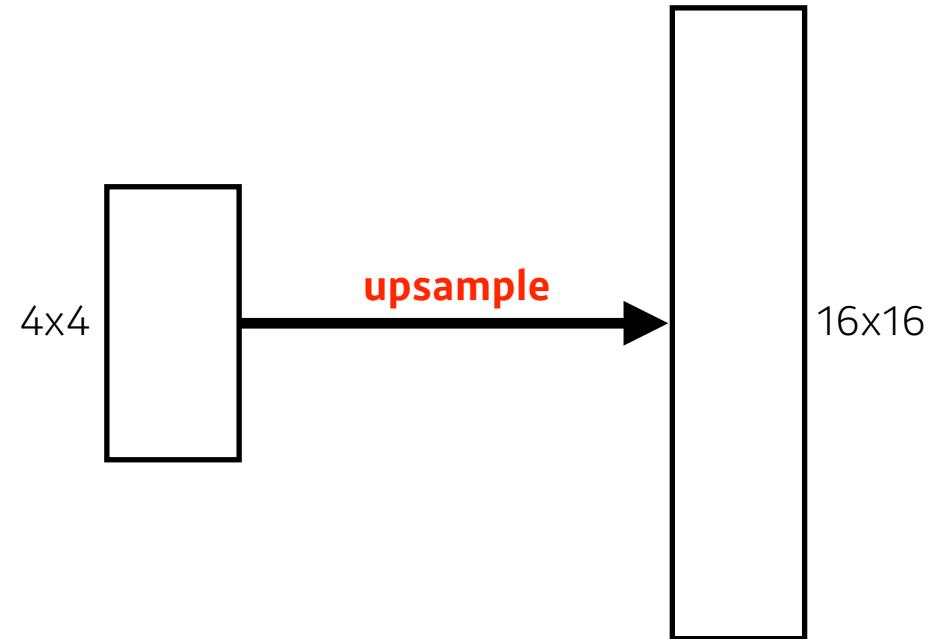
Fully Convolutional Network



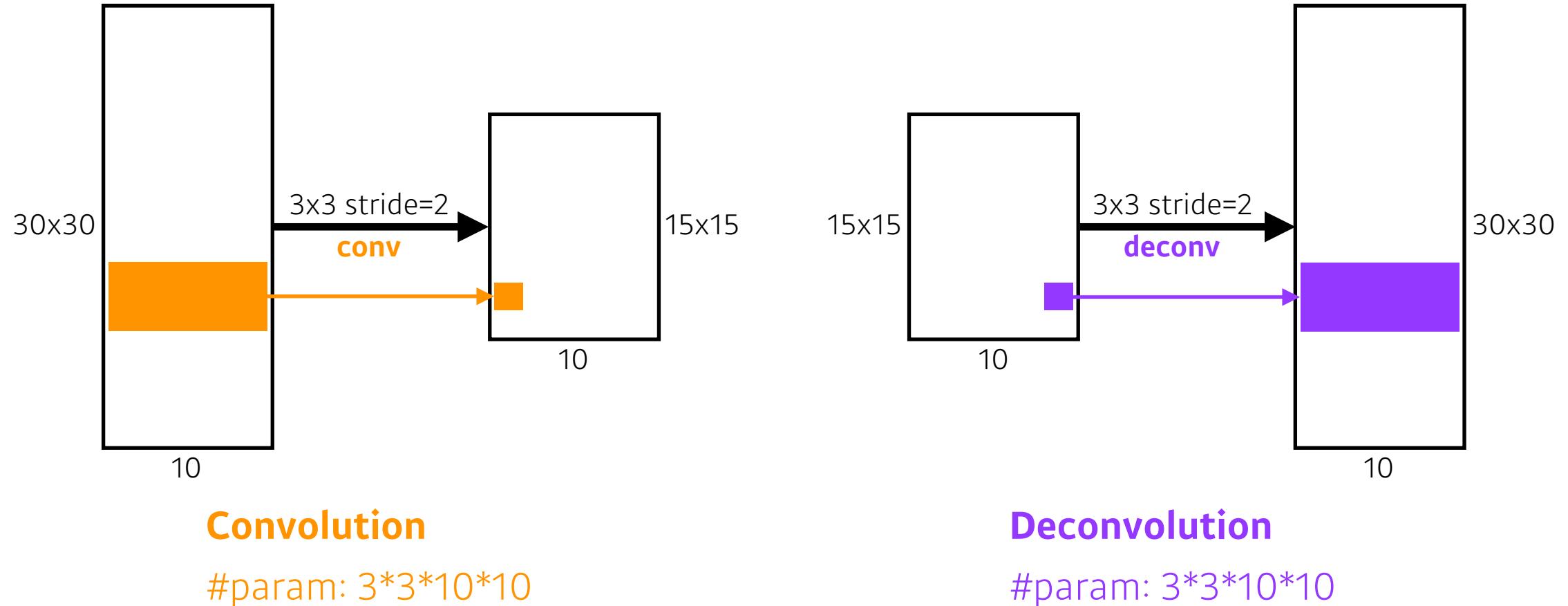
- Transforming fully connected layers into convolution layers enables a classification net to output a heatmap.

Fully Convolutional Network

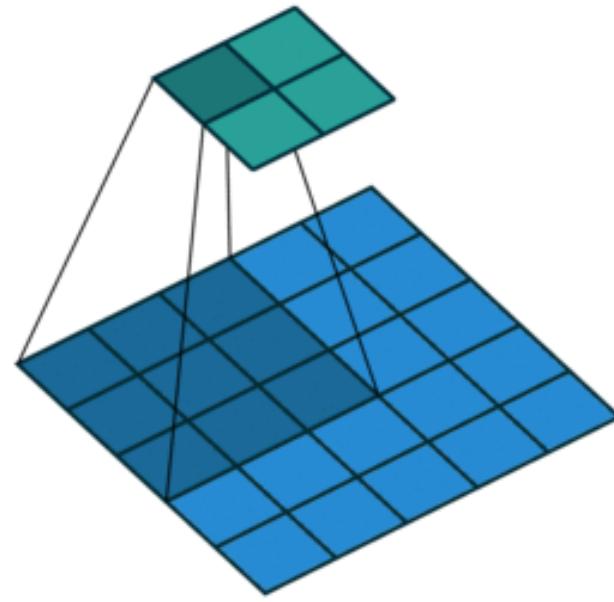
- While FCN can run with inputs of any size, the output dimensions are typically reduced by subsampling.
- So we need a way to connect the coarse output to the dense pixels.



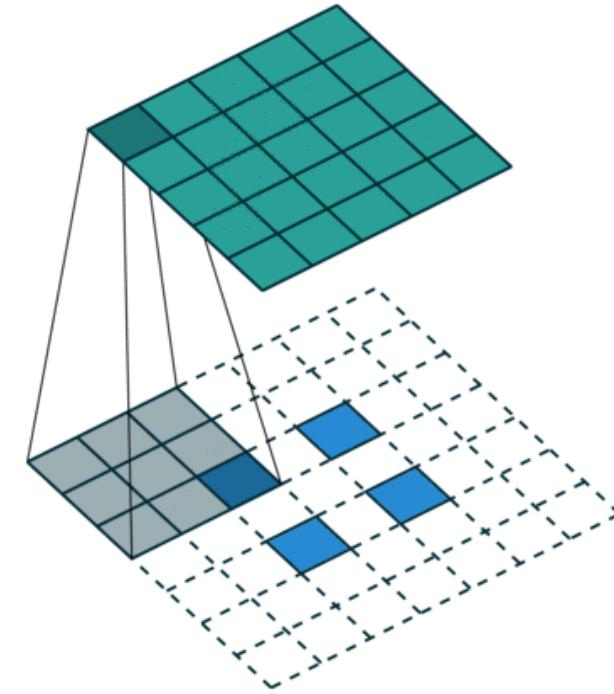
Deconvolution (conv transpose)



Deconvolution (conv transpose)

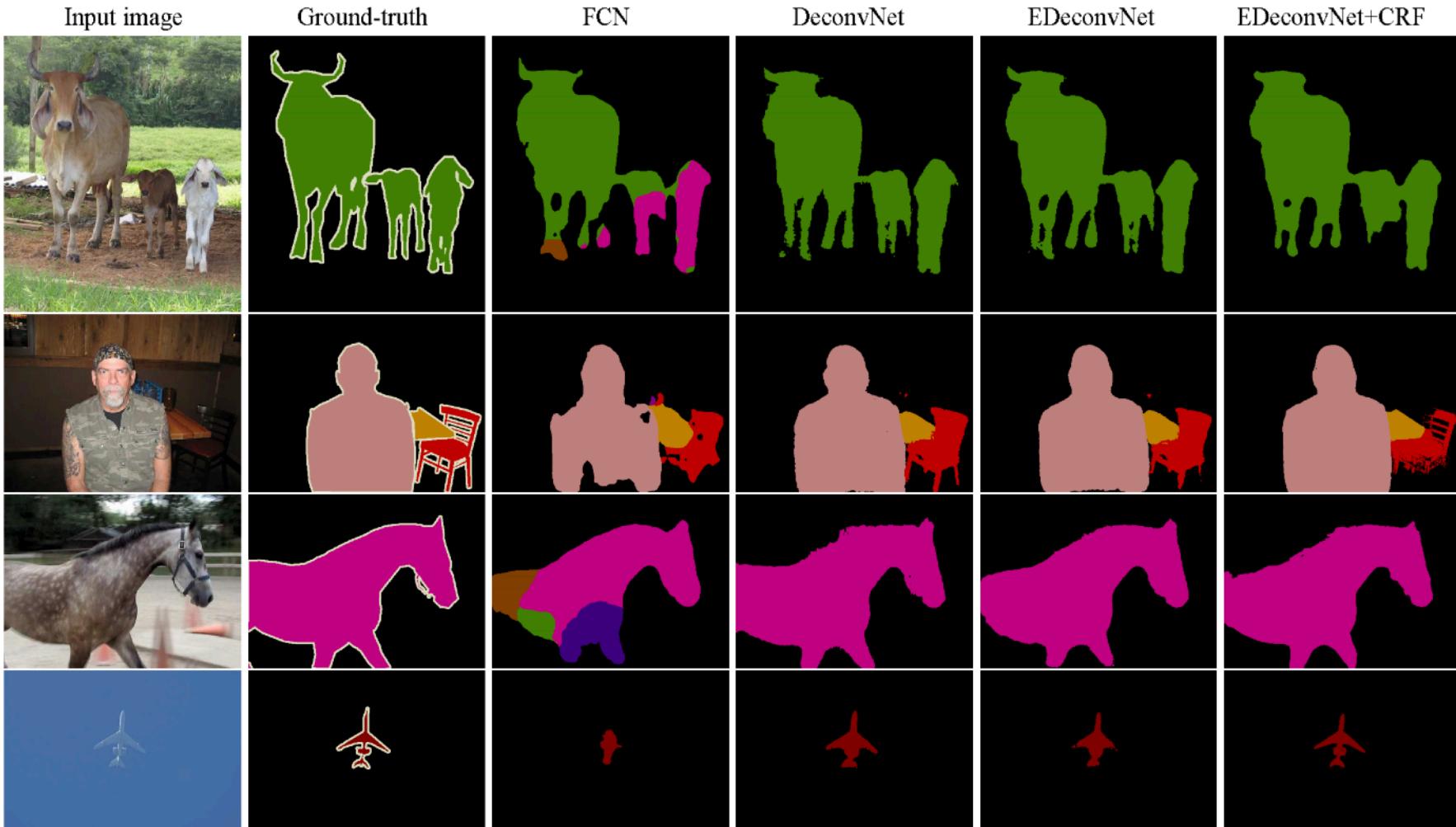


Convolution



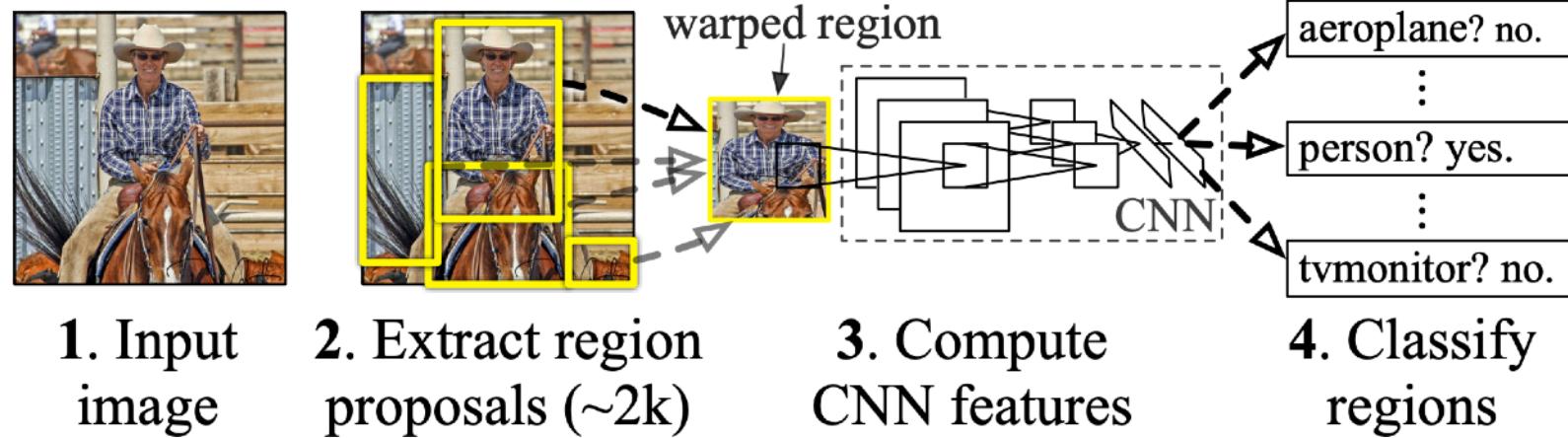
Deconvolution

Results



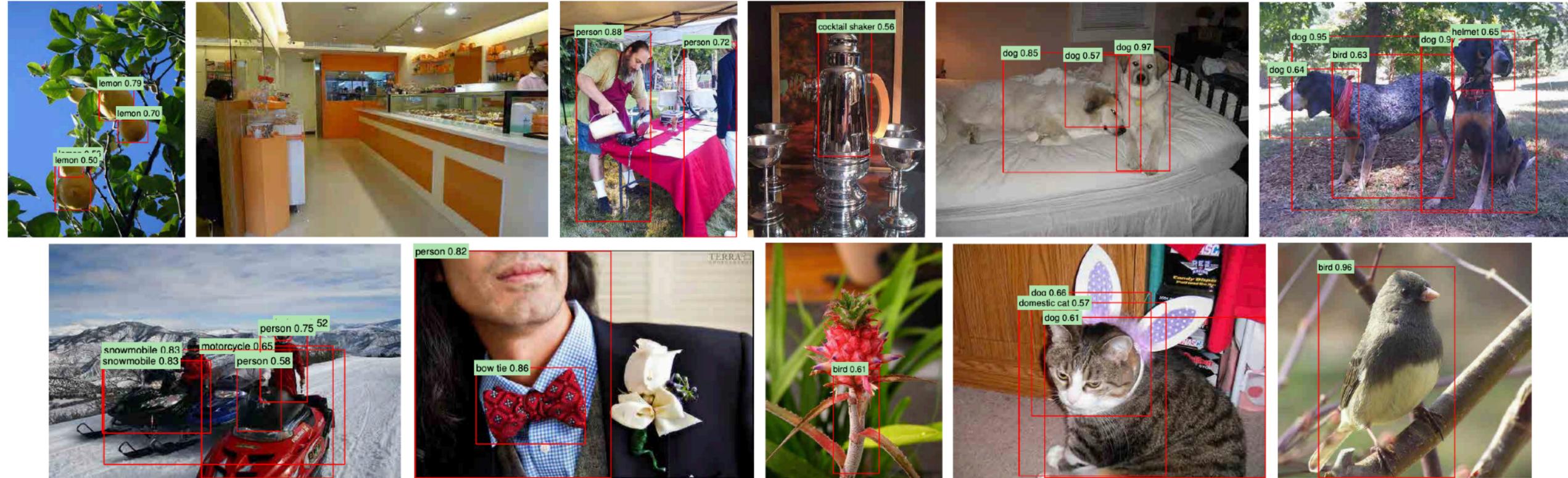
Detection

R-CNN



- R-CNN (1) takes an input image, (2) extracts around 2,000 region proposals (using Selective search), (3) compute features for each proposal (using AlexNet), and then (4) classifies with linear SVMs.

R-CNN



SPPNet



crop

warp

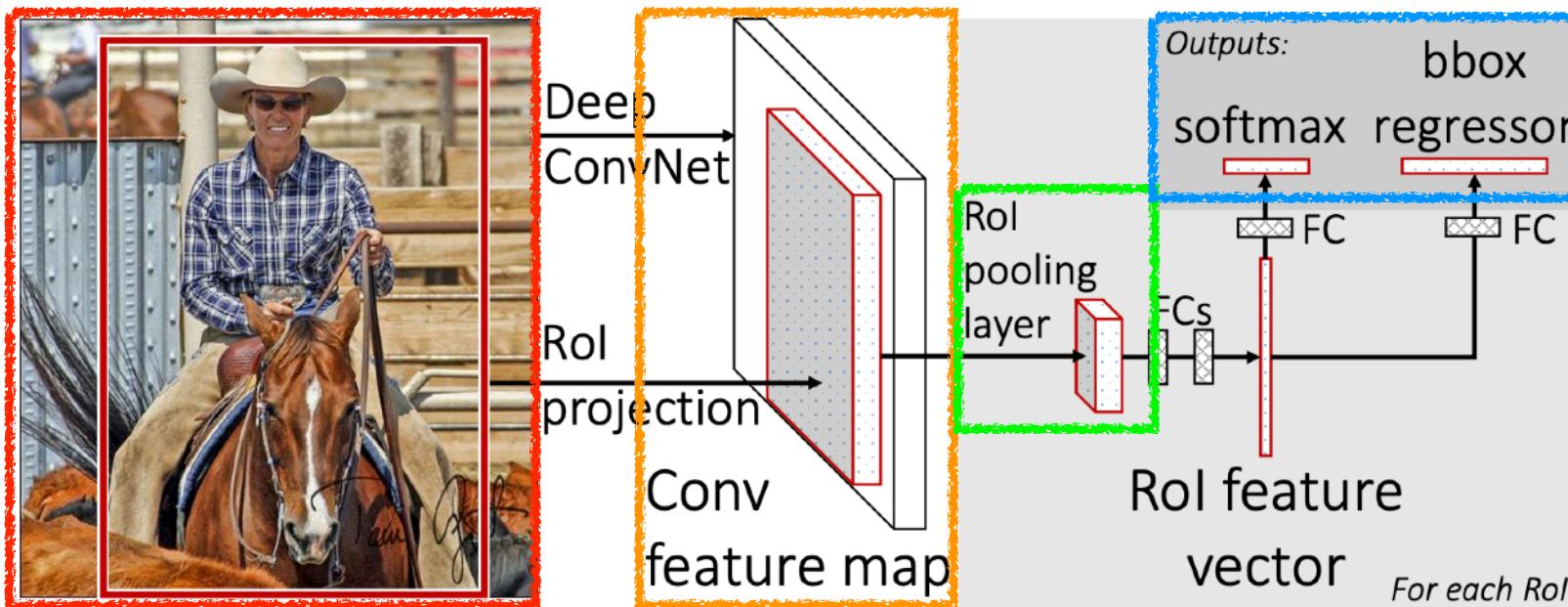


- In R-CNN, the number of crop/warp is usually over 2,000 meaning that CNN must run more than 2,000 times (59s/image on CPU).
- However, in SPPNet, CNN runs once.



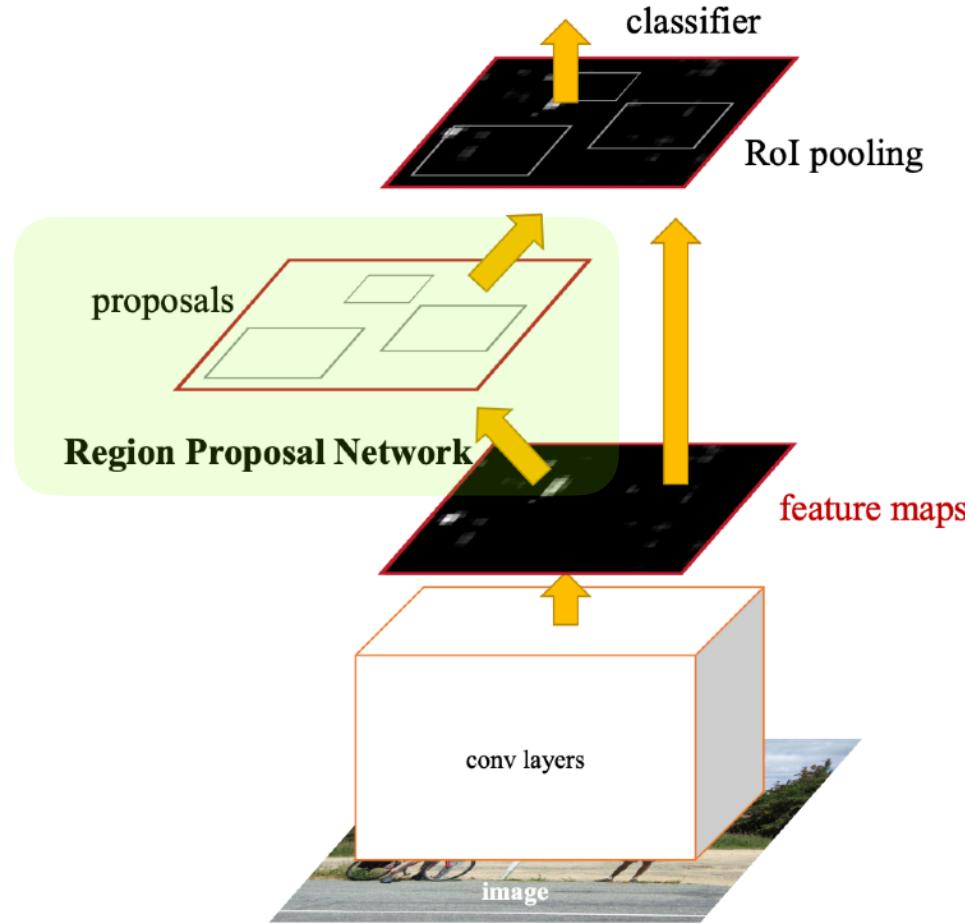
Fast R-CNN

1. Takes an input and a set of bounding boxes.
2. Generated convolutional feature map
3. For each region, get a fixed length feature from ROI pooling
4. Two outputs: class and bounding-box regressor.



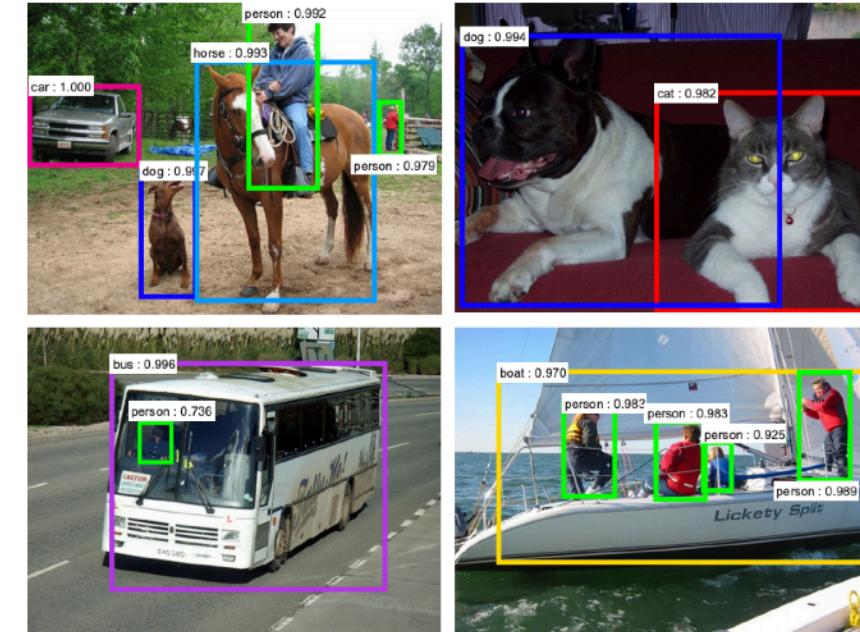
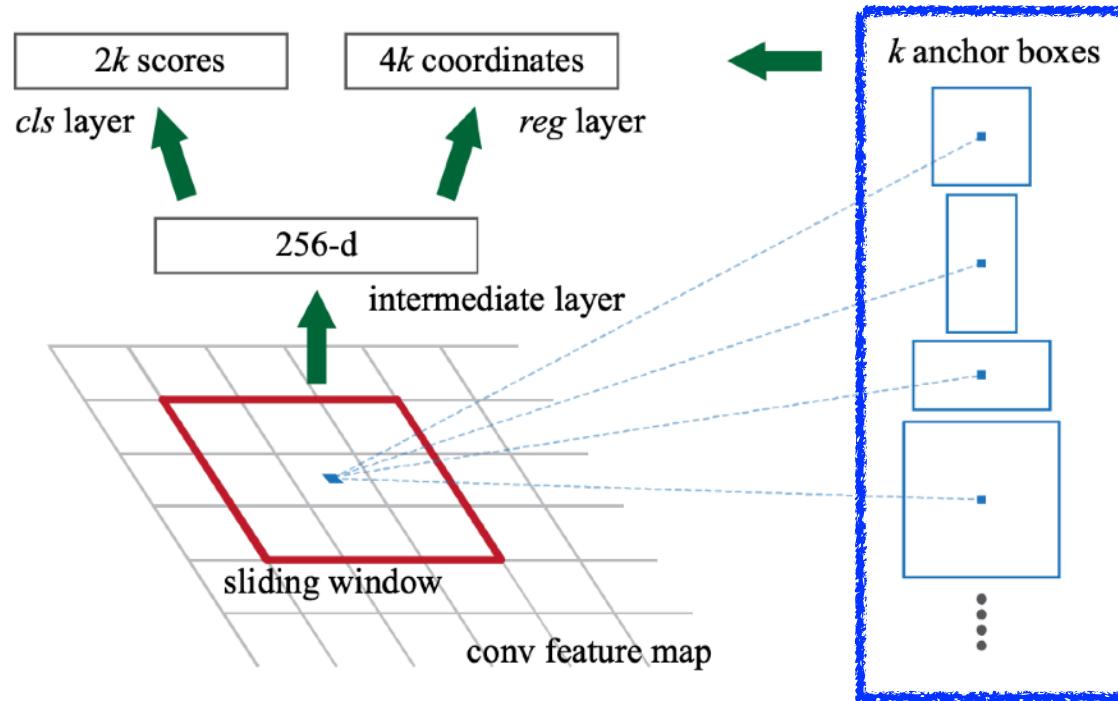
Faster R-CNN

This part is new!



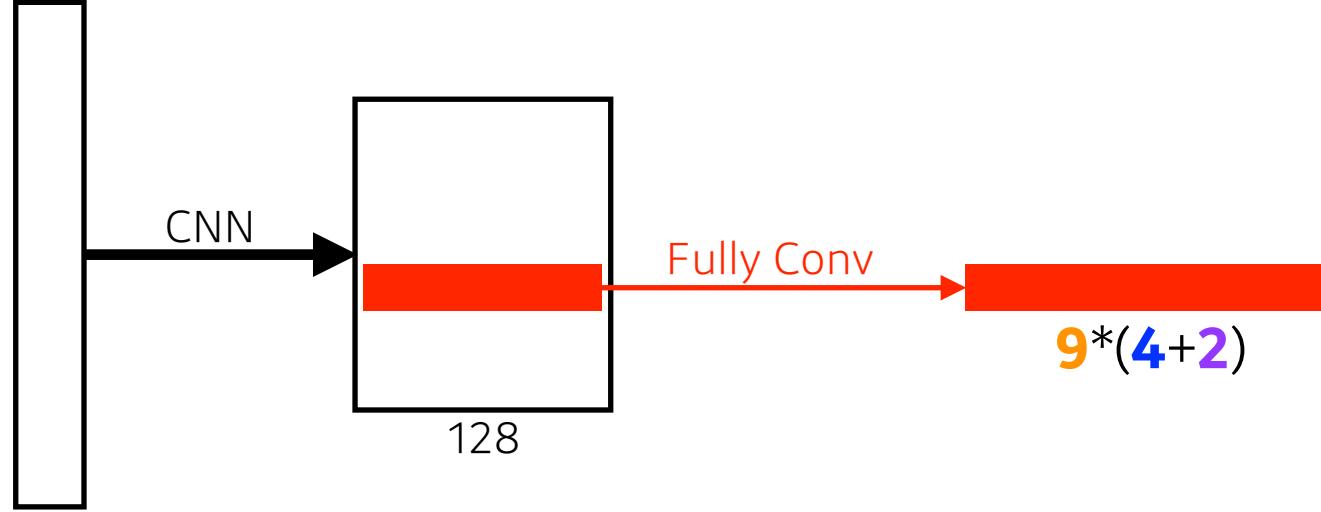
- Faster R-CNN = Region Proposal Network + Fast R-CNN

Region Proposal Network



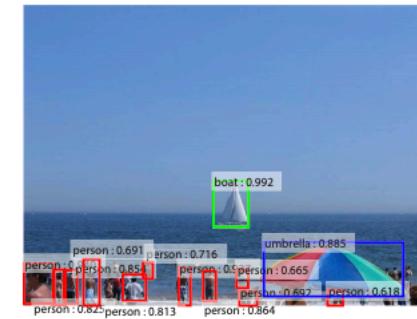
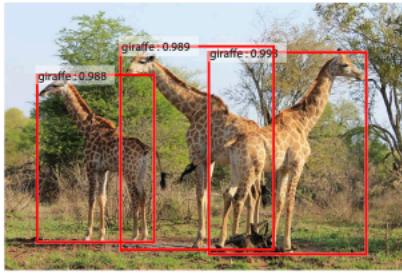
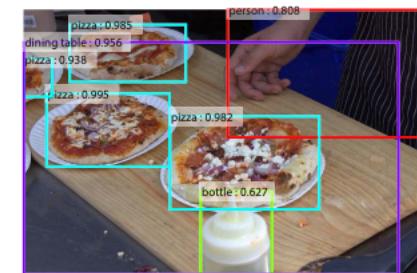
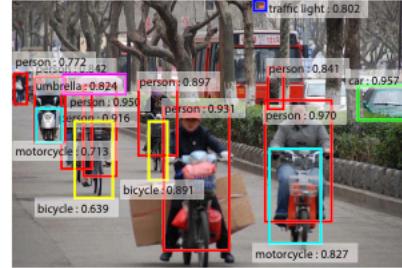
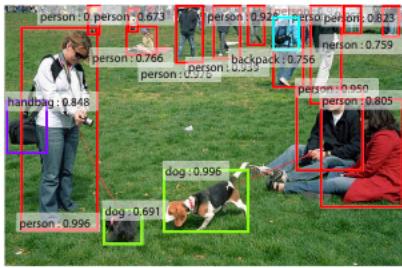
Anchor boxes: detection boxes with predefined sizes.

Region Proposal Network

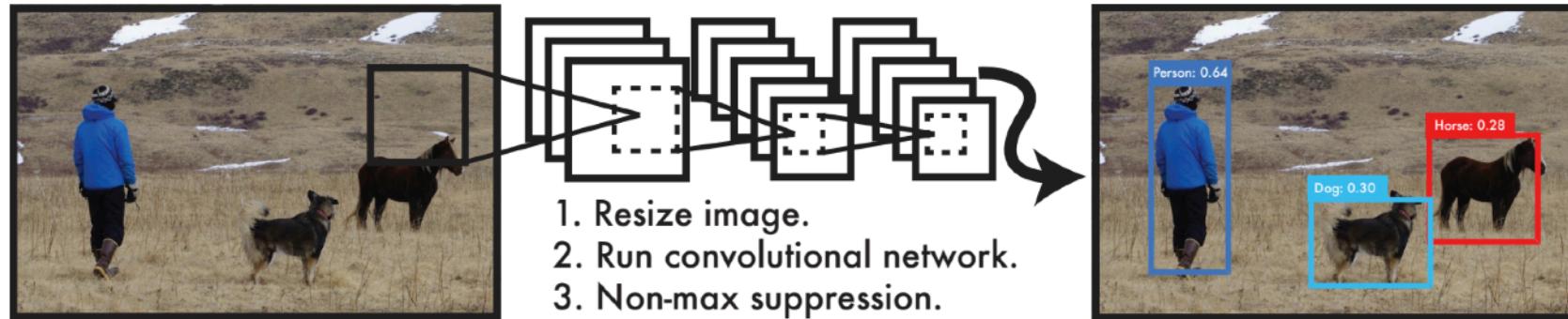


- ➊ **9**: Three different region sizes (128, 256, 512) with three different ratios (1:1, 1:2, 2:1)
- ➋ **4**: four bounding box regression parameters
- ➌ **2**: box classification (whether to use it or not)

Faster R-CNN

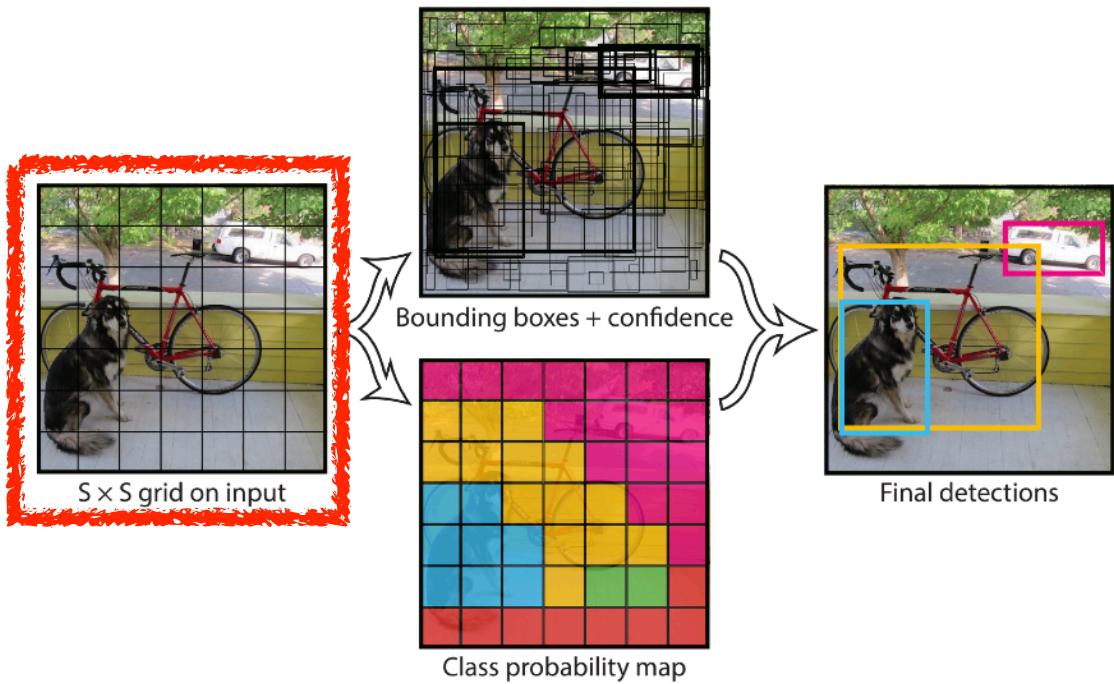


YOLO



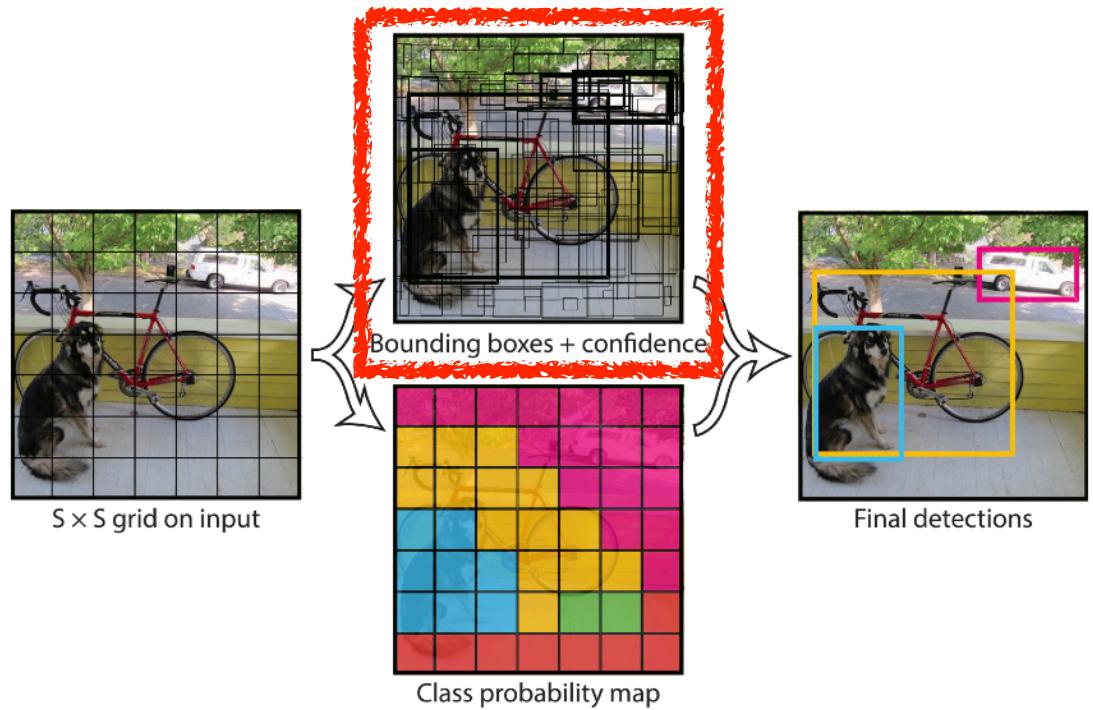
- YOLO (v1) is an extremely fast object detection algorithm.
 - baseline: 45fps / smaller version: 155fps
- It **simultaneously** predicts multiple bounding boxes and class probabilities.
 - No explicit bounding box sampling (compared with Faster R-CNN)

YOLO



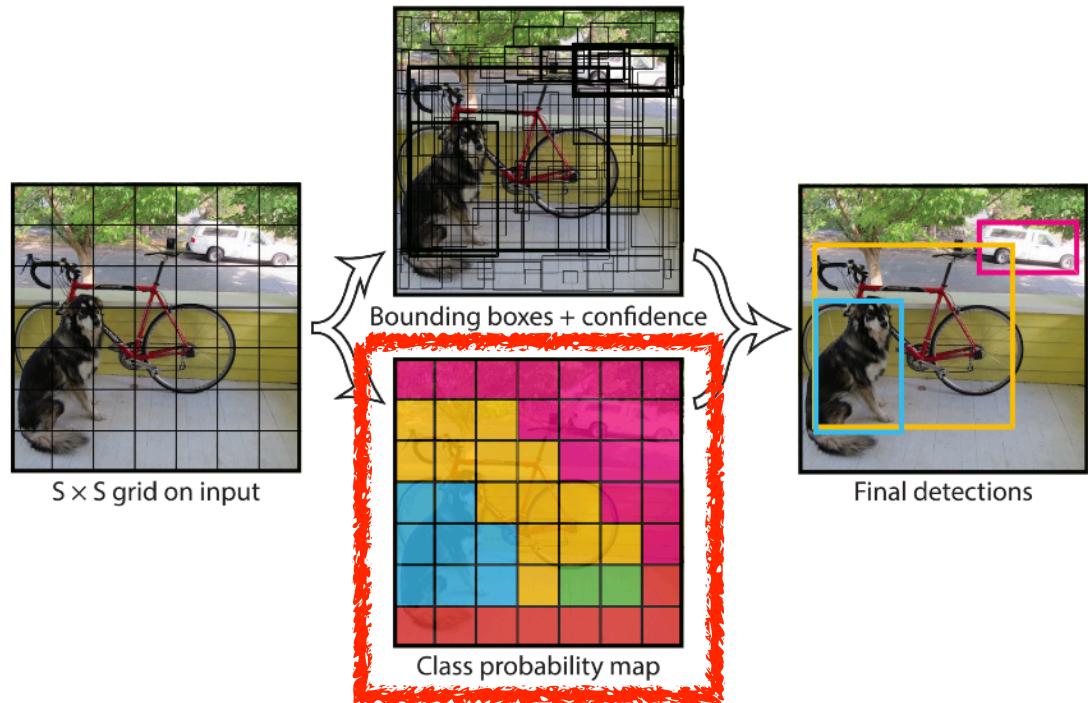
- Given an image, YOLO divides it into $S \times S$ grid.
- If the center of an object falls into the grid cell, that grid cell is responsible for detection.

YOLO



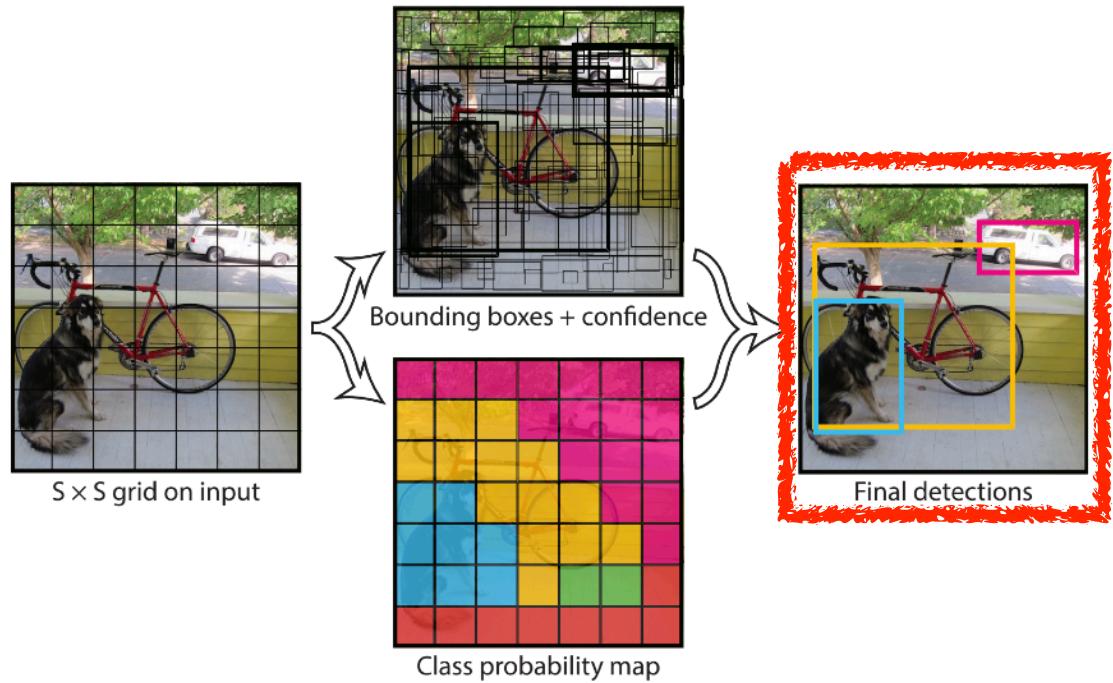
- Each cell predicts B bounding boxes ($B=5$).
 - Each bounding box predicts
 - box refinement ($x / y / w / h$)
 - confidence (of objectness)

YOLO



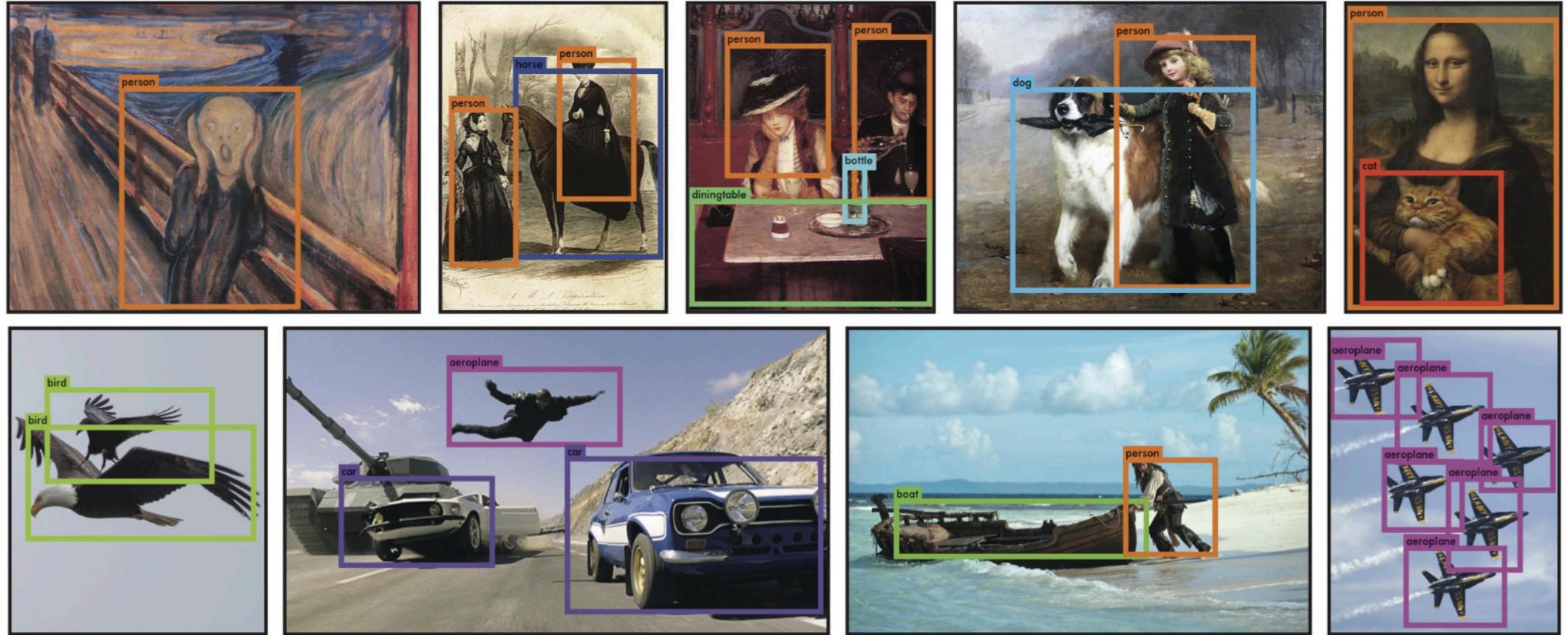
- Each cell predicts B bounding boxes ($B=5$).
 - Each bounding box predicts
 - box refinement ($x / y / w / h$)
 - confidence (of objectness)
- Each cell predicts C class probabilities.

YOLO



- In total, it becomes a tensor with $S \times S \times (B * 5 + C)$ size.
 - $S \times S$: Number of cells of the grid
 - $B * 5$: B bounding boxes with offsets (x, y, w, h) and confidence
 - C : Number of classes

YOLO



Thank you for listening
