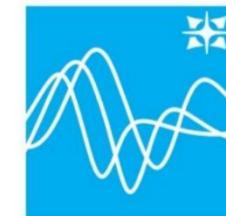
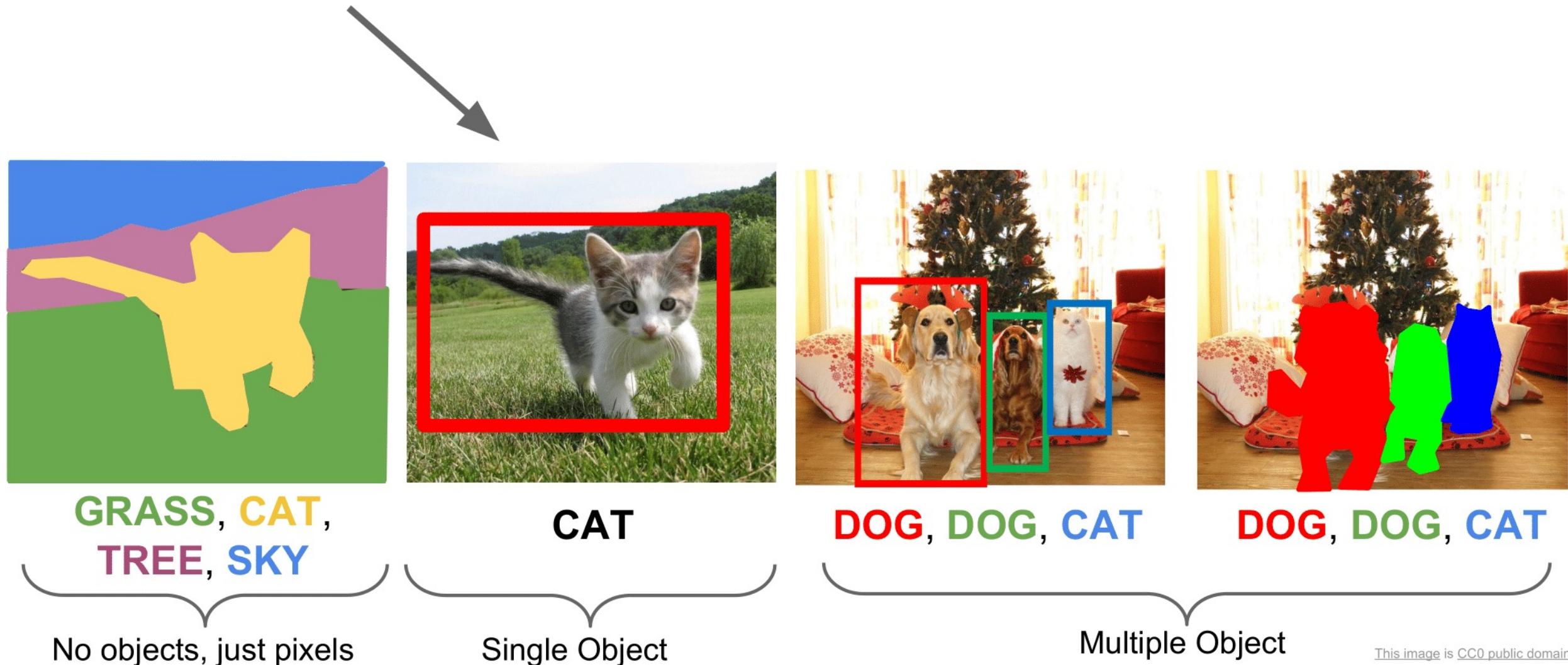


# Сверточные нейронные сети для задачи распознавания объектов



Кафедра  
технологий  
проектирования  
сложных  
технических  
систем

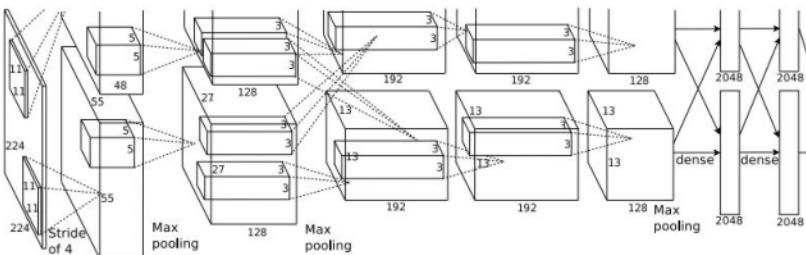
# Classification + Localization



# Classification + Localization



[This image is CC0 public domain](#)



Treat localization as a  
regression problem!

Fully  
Connected:  
4096 to 1000

Vector:  
Fully  
Connected:  
4096 to 4

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

Multitask Loss

Box  
Coordinates → L2 Loss  
( $x, y, w, h$ )

Correct label:  
Cat

Softmax  
Loss

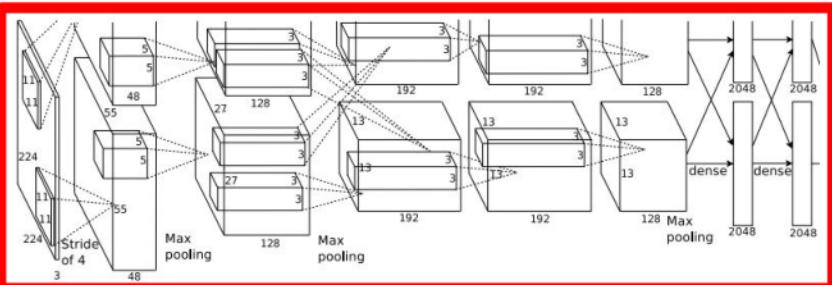
+

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

# Classification + Localization



This image is CC0 public domain



Often pretrained on ImageNet  
(Transfer learning)

Treat localization as a  
regression problem!

Vector: 4096  
Fully Connected: 4096 to 4

Box Coordinates  $\rightarrow$  L2 Loss  
( $x, y, w, h$ )

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

Correct label:  
Cat

Softmax  
Loss

+

L2 Loss

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

# Aside: Human Pose Estimation



Represent pose as a set of 14 joint positions:

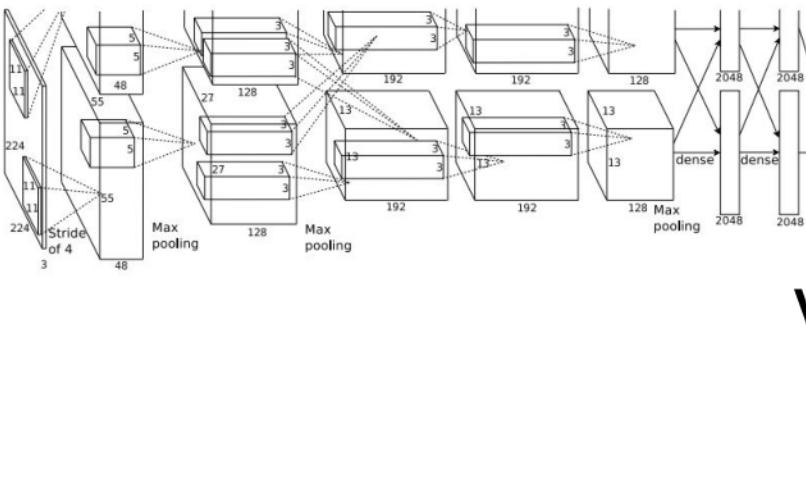
- Left / right foot
- Left / right knee
- Left / right hip
- Left / right shoulder
- Left / right elbow
- Left / right hand
- Neck
- Head top

This image is licensed under [CC-BY 2.0](#).

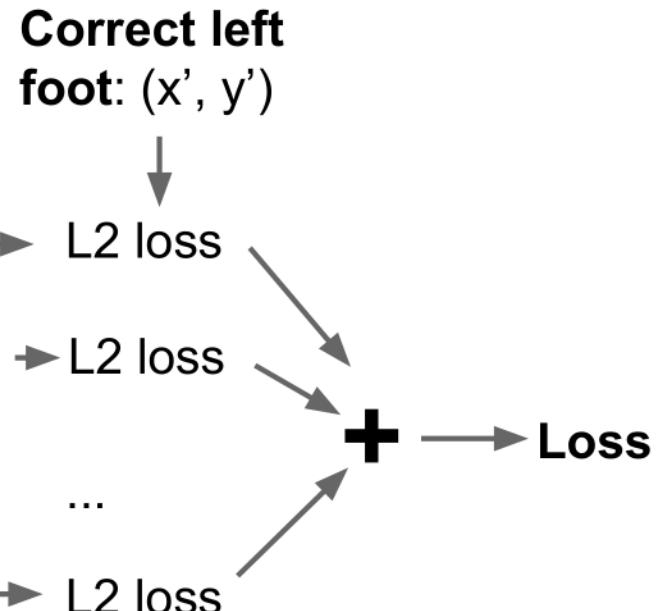
Johnson and Everingham, "Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation", BMVC 2010

*Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung*

# Aside: Human Pose Estimation



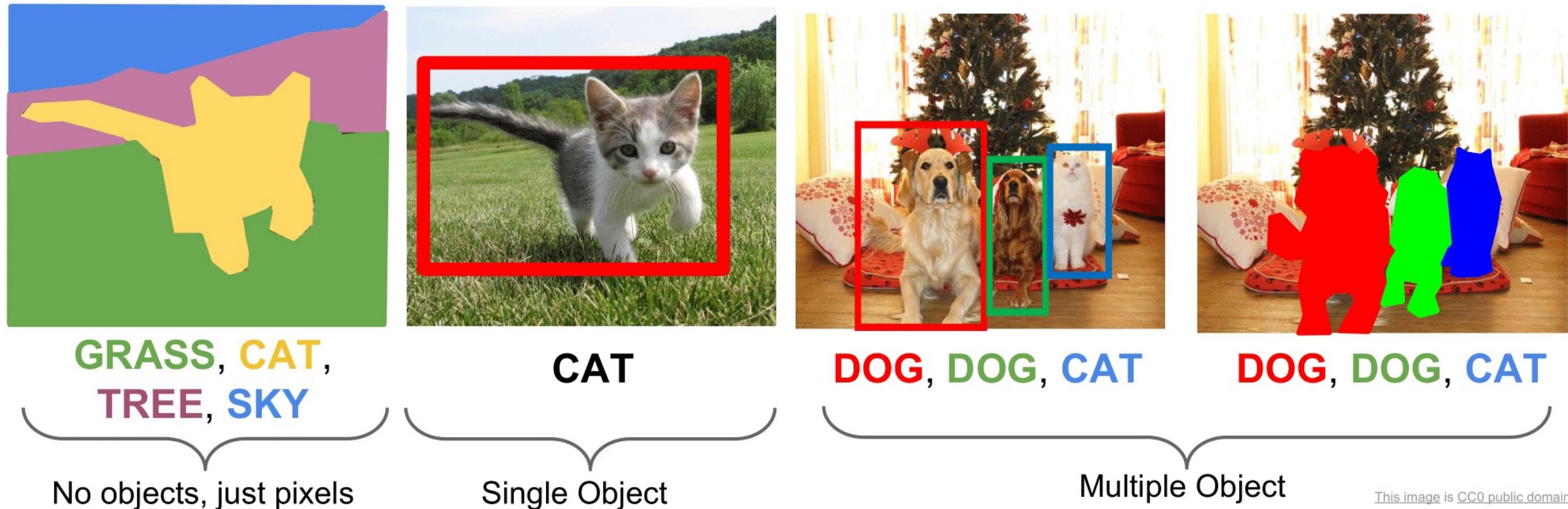
Vector:  
4096



Toshev and Szegedy, “DeepPose: Human Pose Estimation via Deep Neural Networks”, CVPR 2014

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

# Object Detection



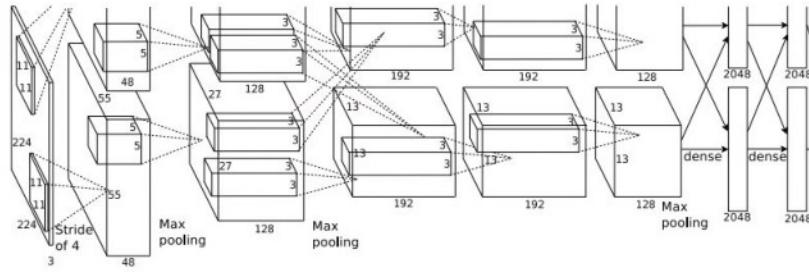
No objects, just pixels

Single Object

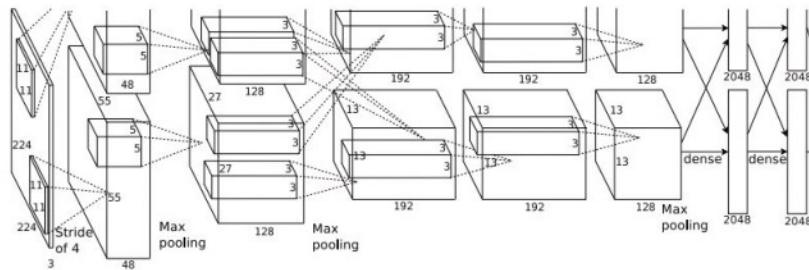
Multiple Object

[This image is CC0 public domain](#)

# Object Detection as Regression?



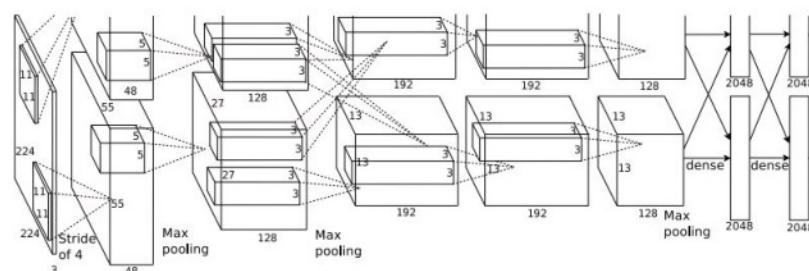
CAT: (x, y, w, h)



DOG: (x, y, w, h)

DOG: (x, y, w, h)

CAT: (x, y, w, h)



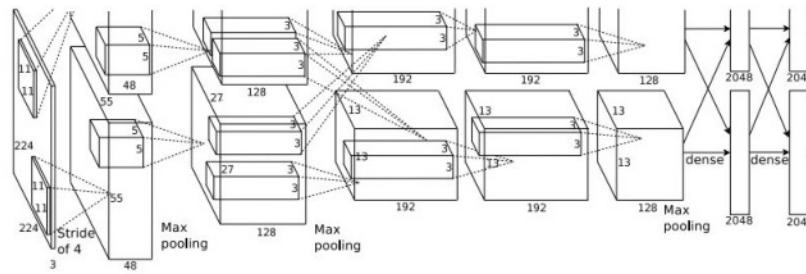
DUCK: (x, y, w, h)

DUCK: (x, y, w, h)

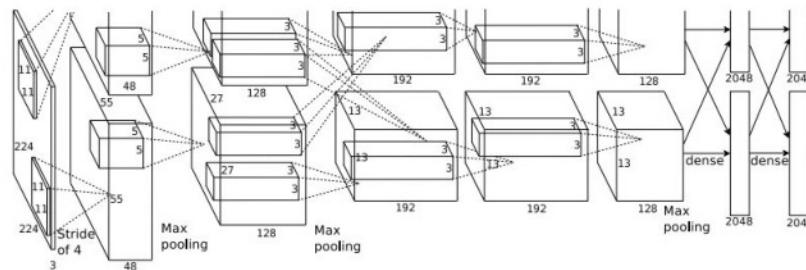
...

# Object Detection as Regression?

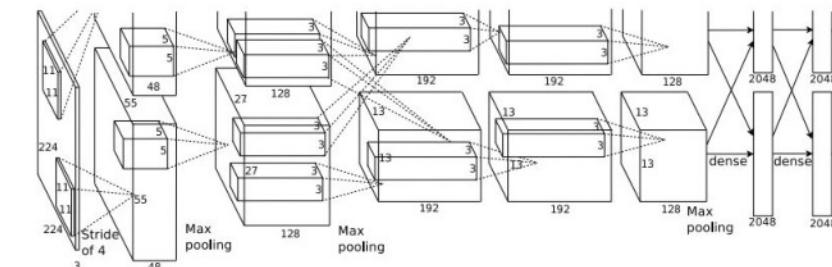
Each image needs a different number of outputs!



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



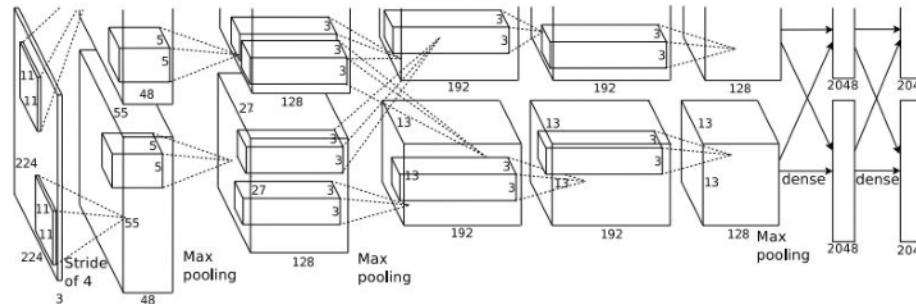
DOG: (x, y, w, h)  
DOG: (x, y, w, h) 16 numbers  
CAT: (x, y, w, h)



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

# Object Detection as Classification: Sliding Window

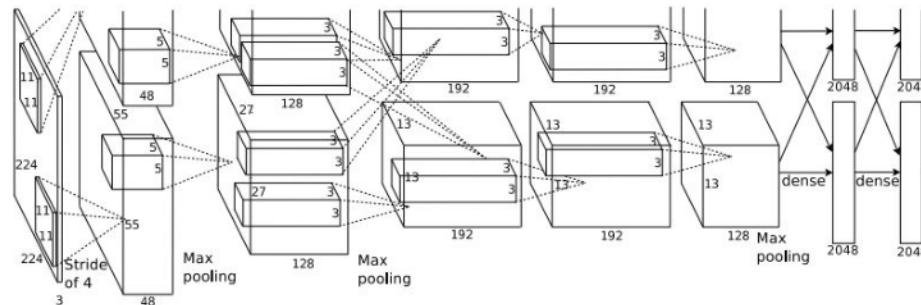
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO  
Cat? NO  
Background? YES

# Object Detection as Classification: Sliding Window

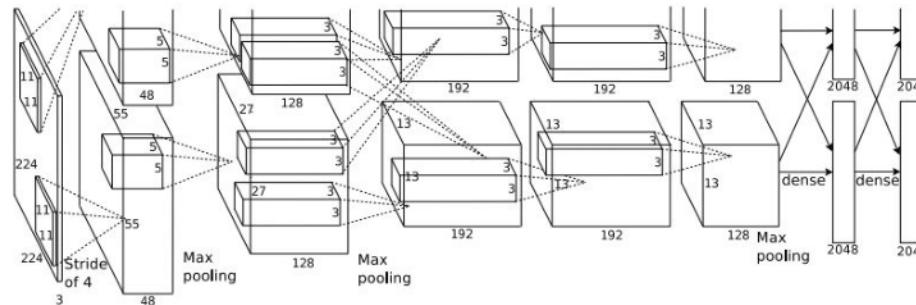
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES  
Cat? NO  
Background? NO

# Object Detection as Classification: Sliding Window

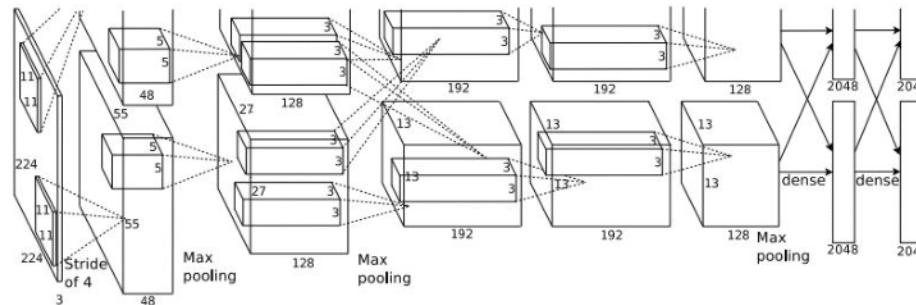
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES  
Cat? NO  
Background? NO

# Object Detection as Classification: Sliding Window

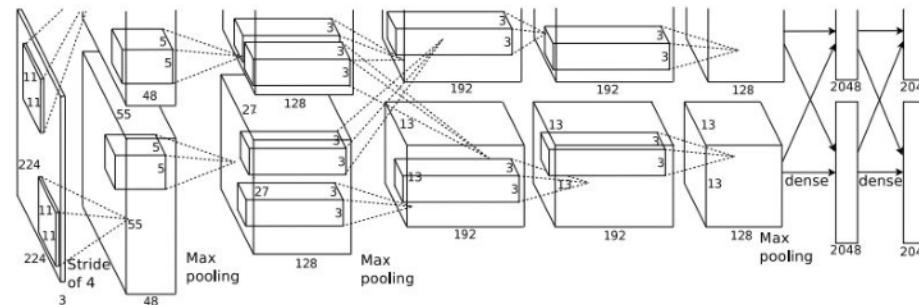
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO  
Cat? YES  
Background? NO

# Object Detection as Classification: Sliding Window

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

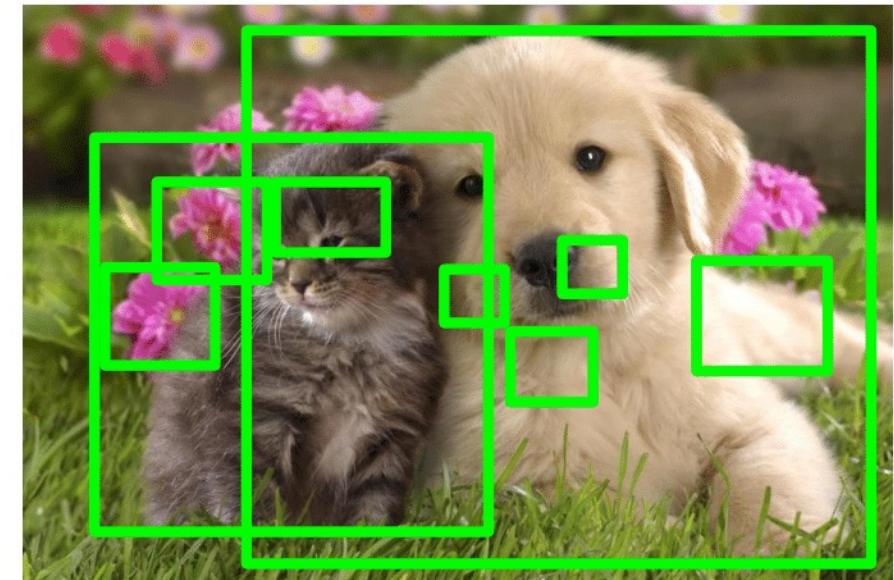


Dog? NO  
Cat? YES  
Background? NO

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

# Region Proposals

- Find “blobby” image regions that are likely to contain objects
- Relatively fast to run; e.g. Selective Search gives 1000 region proposals in a few seconds on CPU



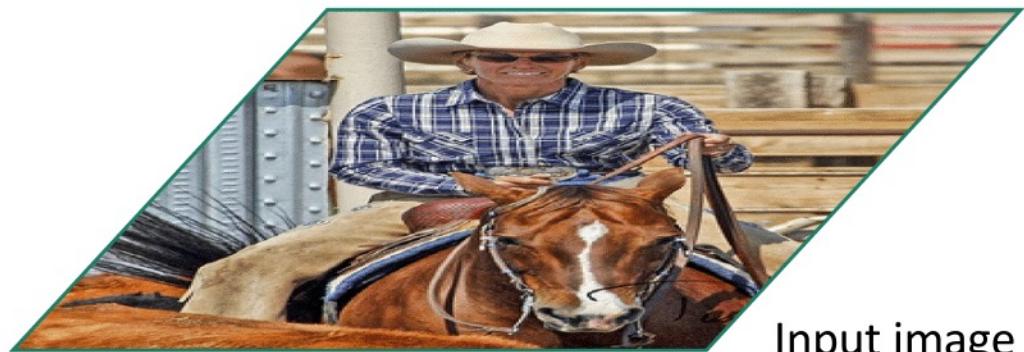
Alexe et al, “Measuring the objectness of image windows”, TPAMI 2012

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



Input 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.

# R-CNN

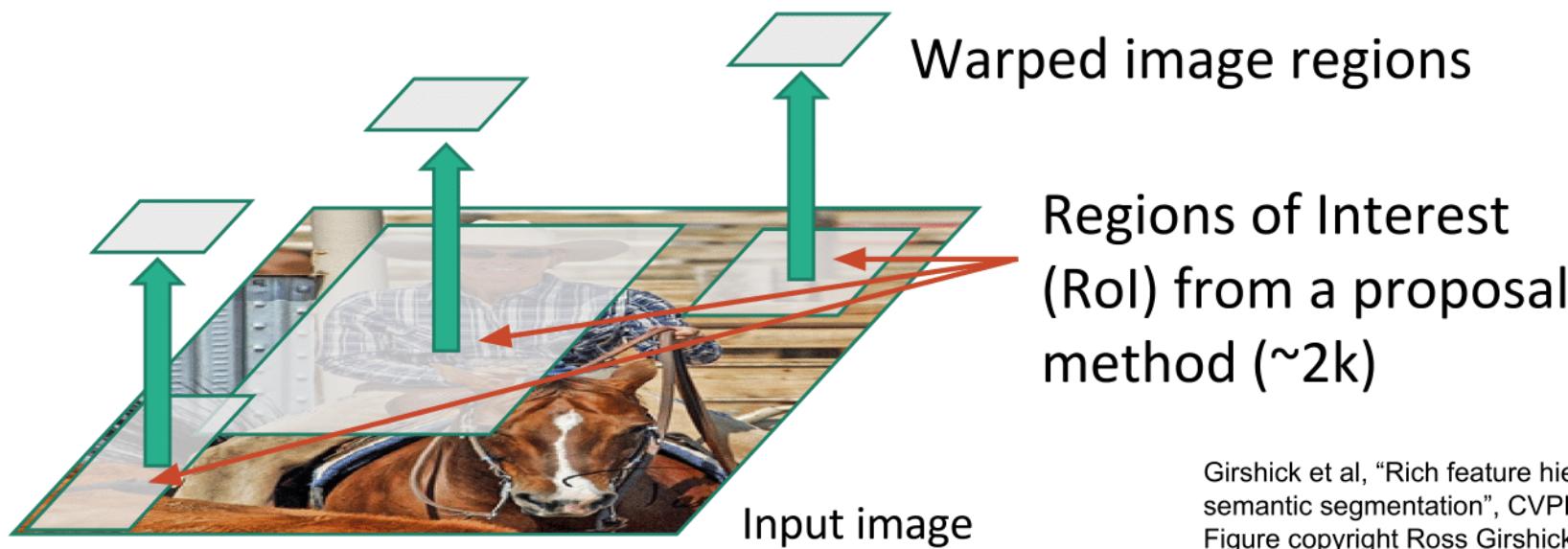


Input image

Regions of Interest  
(RoI) from a proposal  
method (~2k)

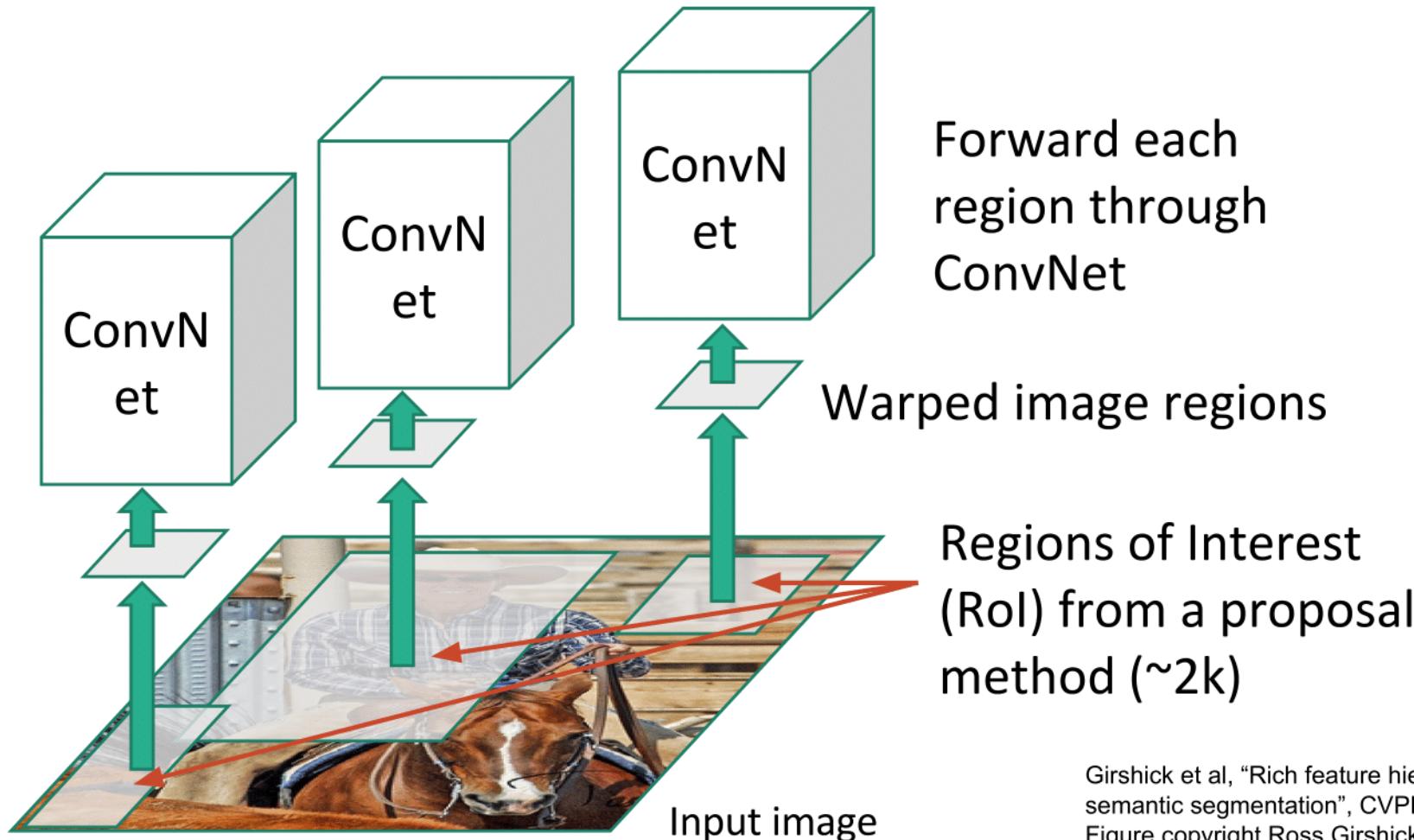
Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.  
Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

# R-CNN



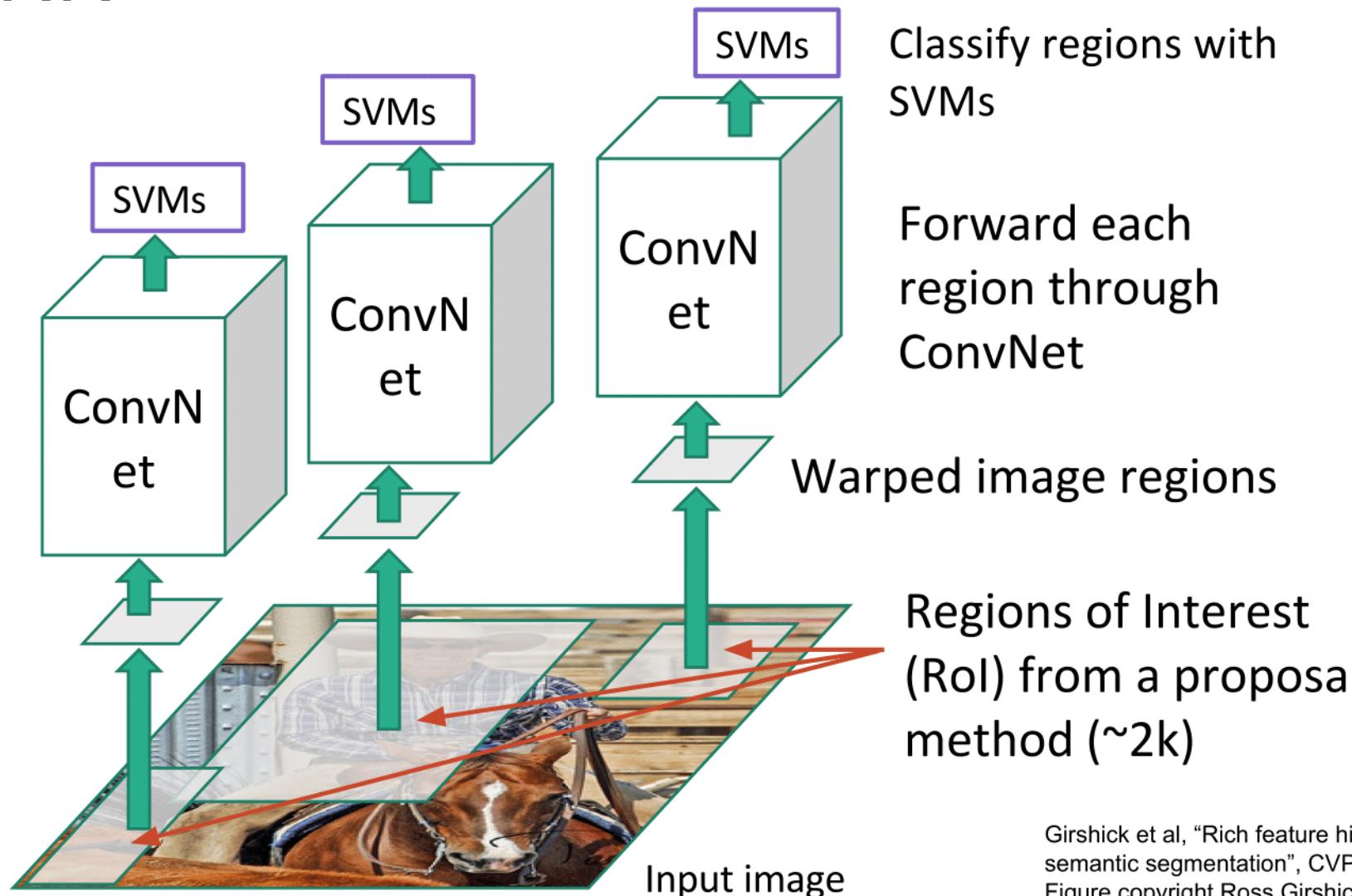
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.  
Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

# R-CNN



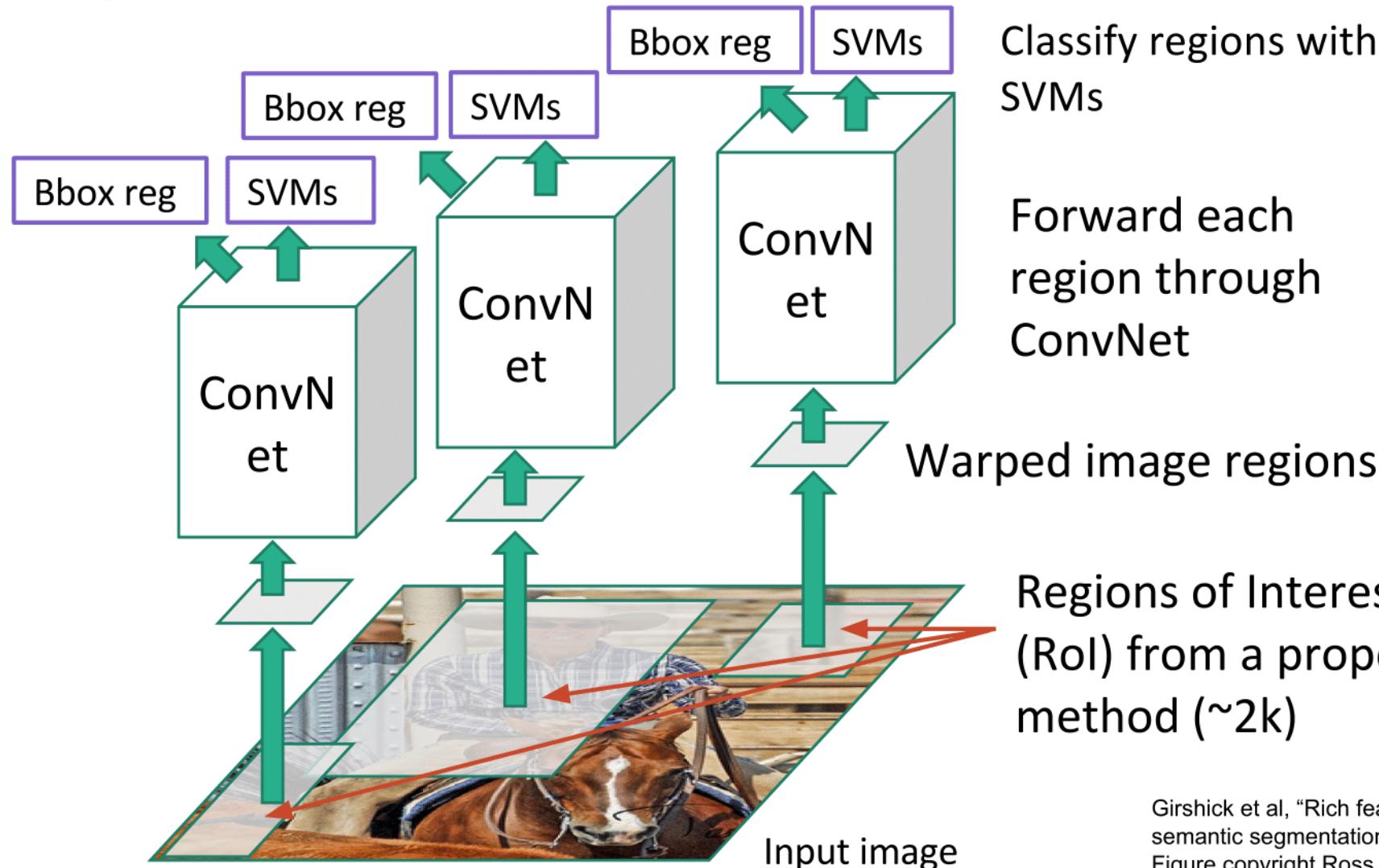
Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.  
Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

# R-CNN



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

# R-CNN



Linear Regression for bounding box offsets

Classify regions with SVMs

Forward each region through ConvNet

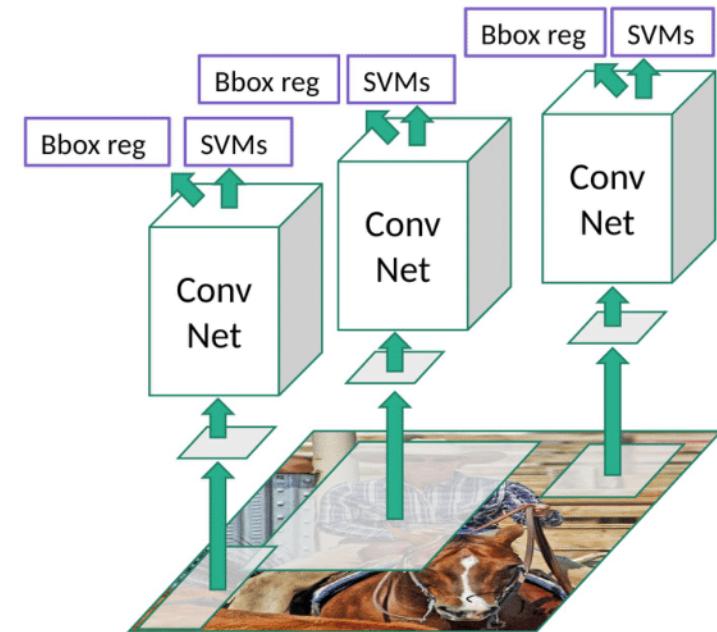
Warped image regions

Regions of Interest (RoI) from a proposal method (~2k)

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

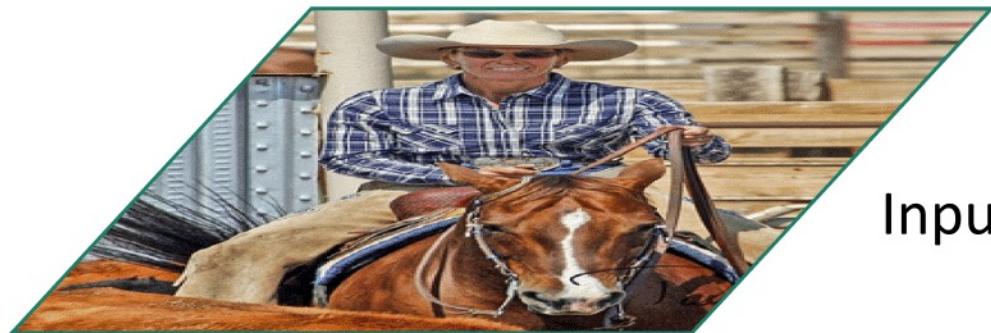
# R-CNN: Problems

- Ad hoc training objectives
  - Fine-tune network with softmax classifier (log loss)
  - Train post-hoc linear SVMs (hinge loss)
  - Train post-hoc bounding-box regressions (least squares)
- Training is slow (84h), takes a lot of disk space
- Inference (detection) is slow
  - 47s / image with VGG16 [Simonyan & Zisserman. ICLR15]
  - Fixed by SPP-net [He et al. ECCV14]



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

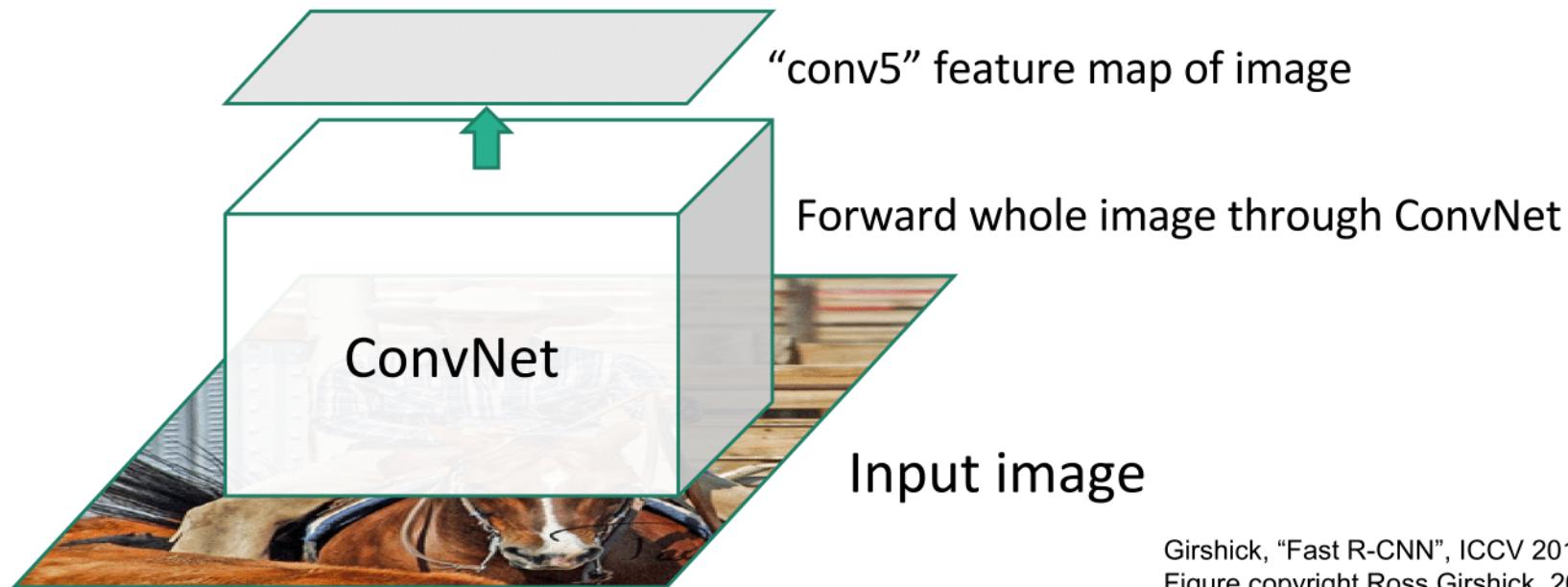
# Fast R-CNN



Input image

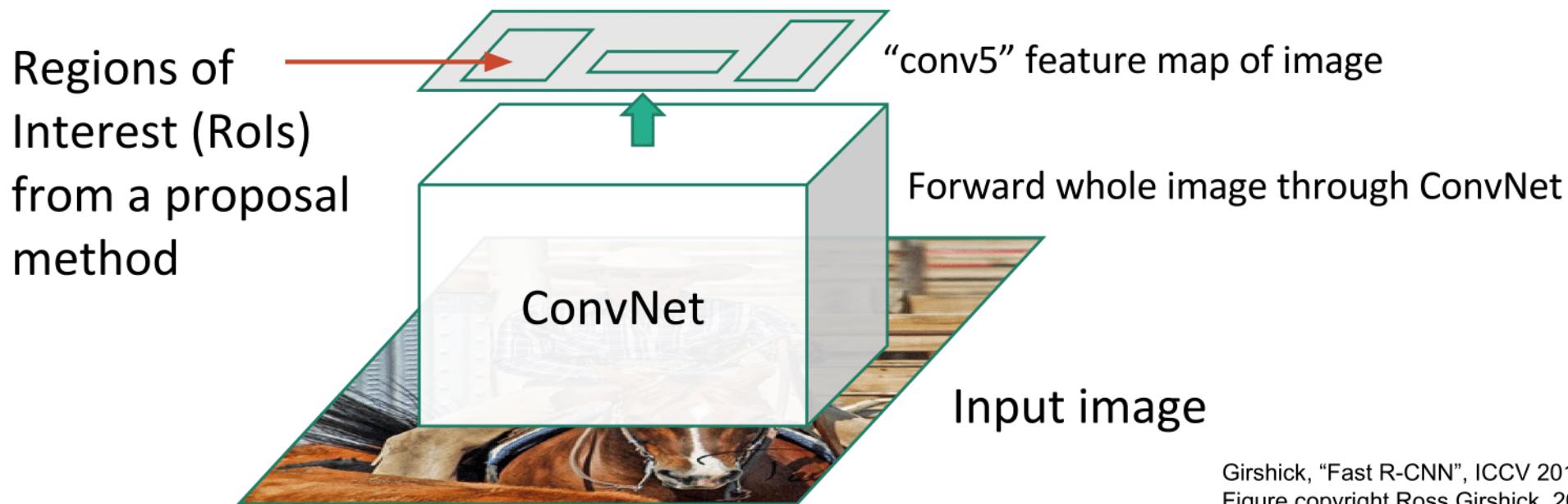
Girshick, "Fast R-CNN", ICCV 2015.  
Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

# Fast R-CNN



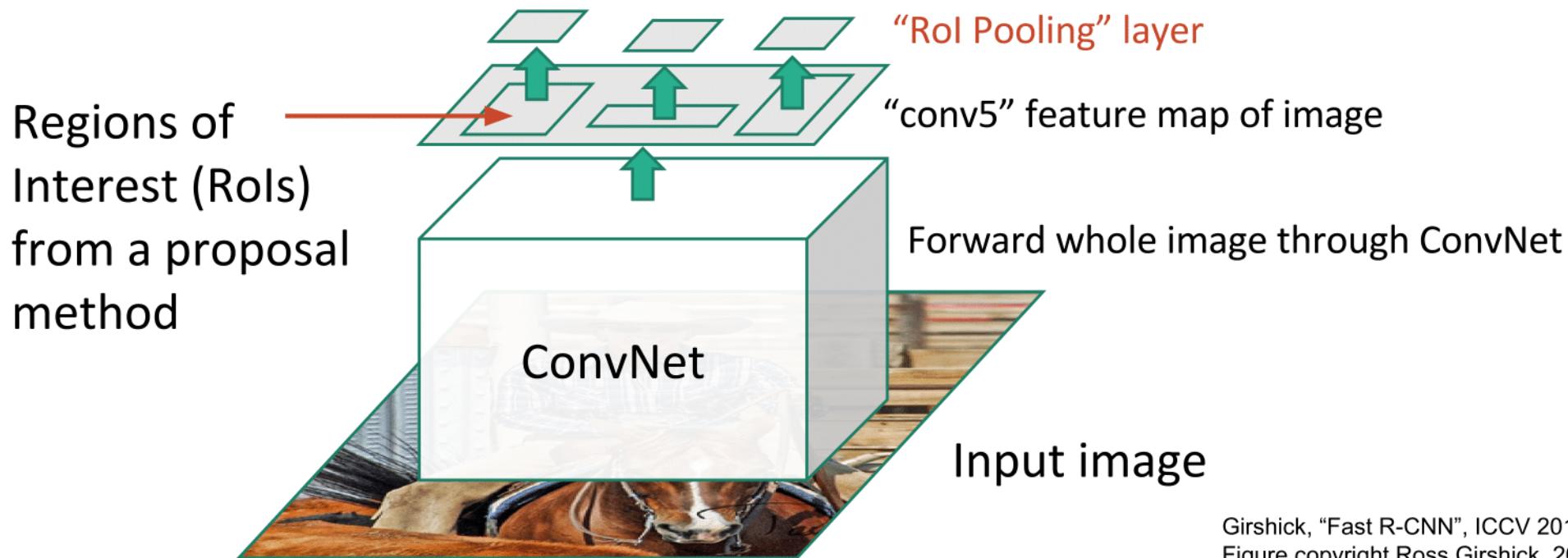
Girshick, "Fast R-CNN", ICCV 2015.  
Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

# Fast R-CNN



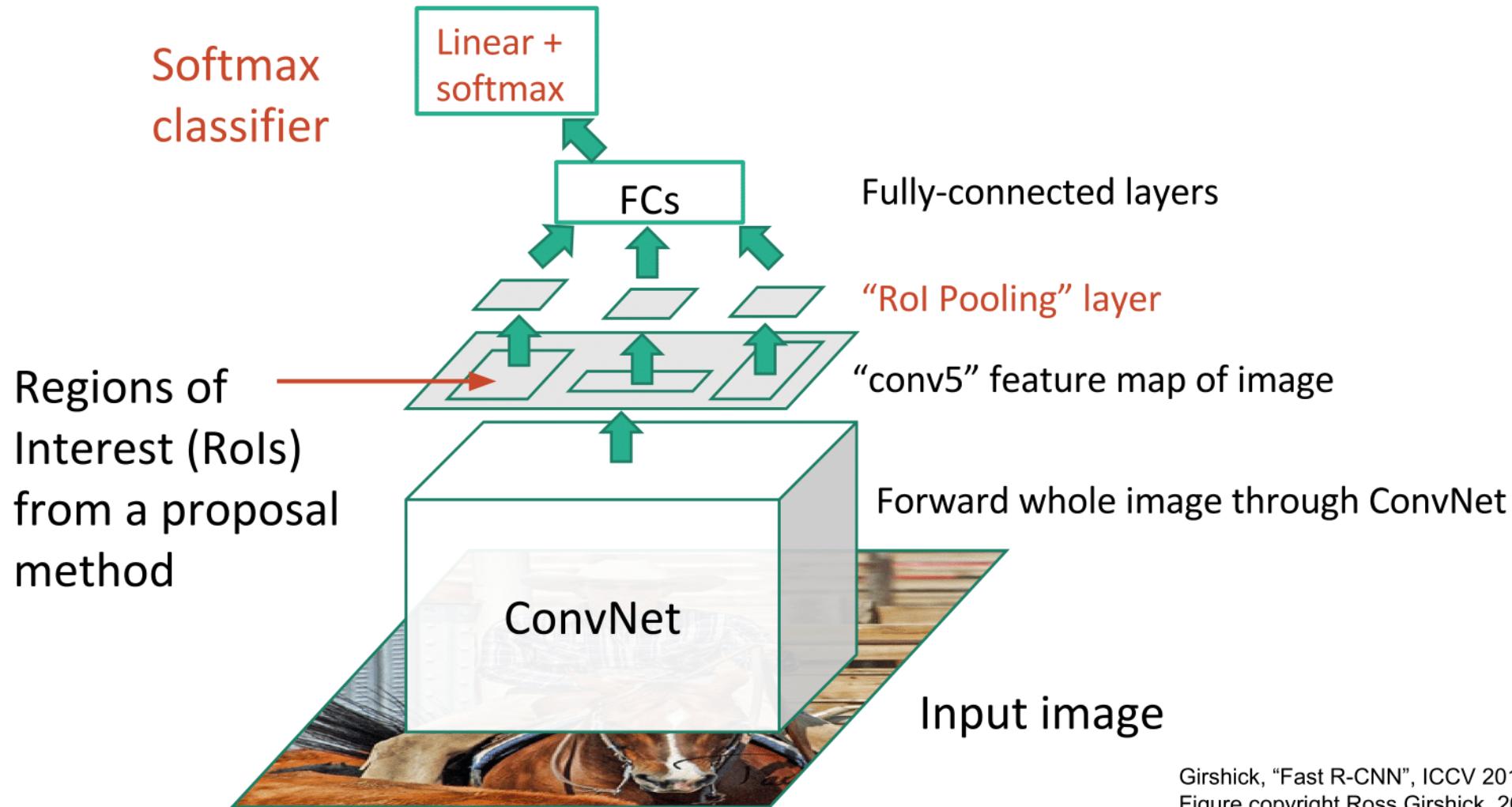
Girshick, "Fast R-CNN", ICCV 2015.  
Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

# Fast R-CNN



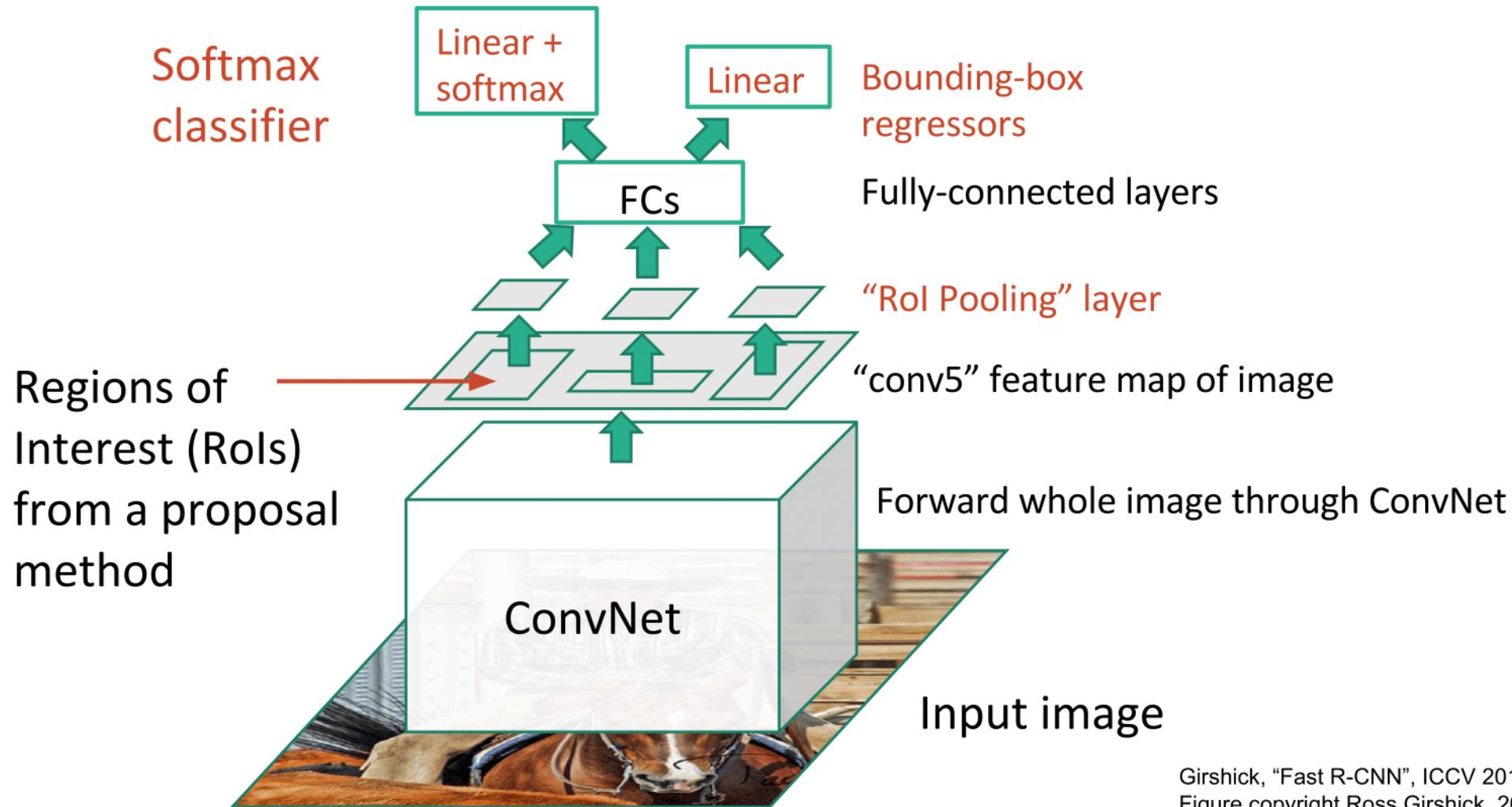
Girshick, "Fast R-CNN", ICCV 2015.  
Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

# Fast R-CNN



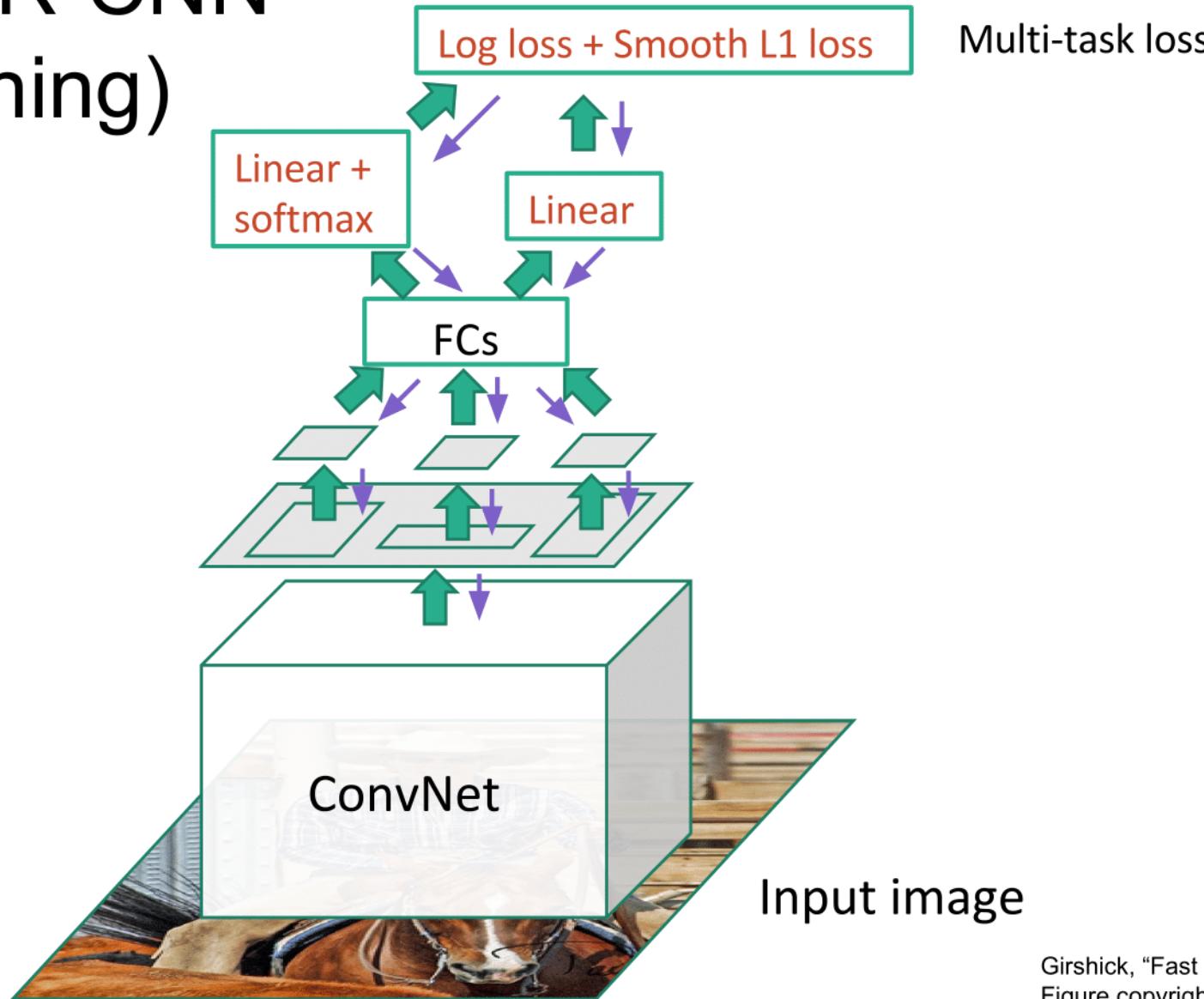
Girshick, "Fast R-CNN", ICCV 2015.  
Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

# Fast R-CNN



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

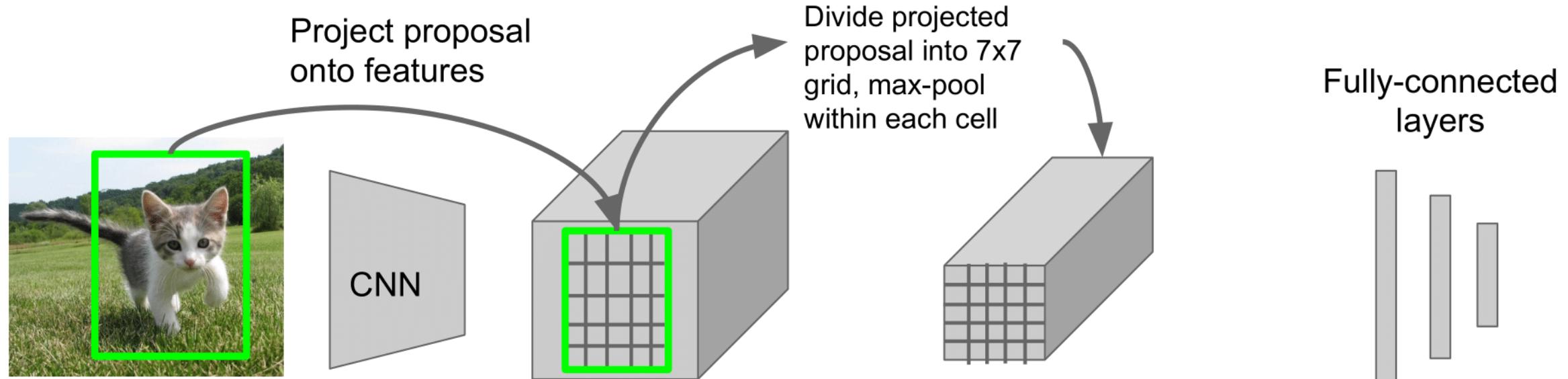
# Fast R-CNN (Training)



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

Figure copyright Ross Girshick, 2015; [source](#). Reproduced with permission.

# Faster R-CNN: RoI Pooling



Hi-res input image:  
3 x 640 x 480  
with region proposal

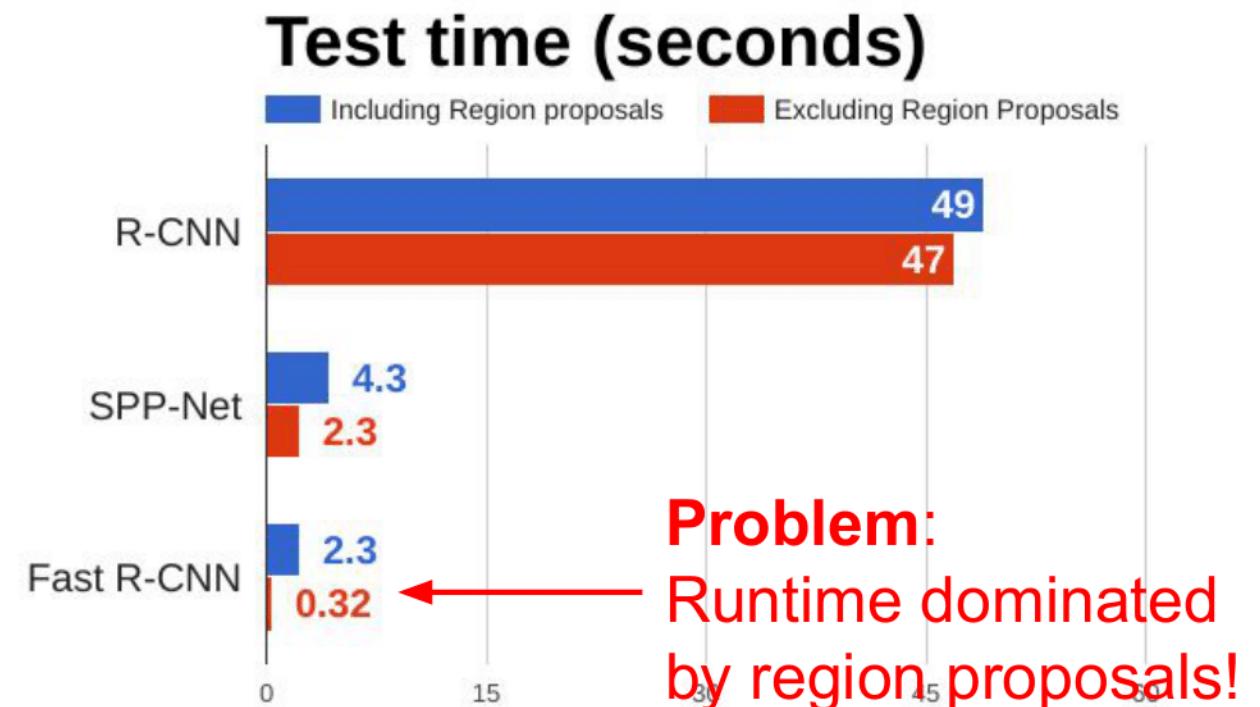
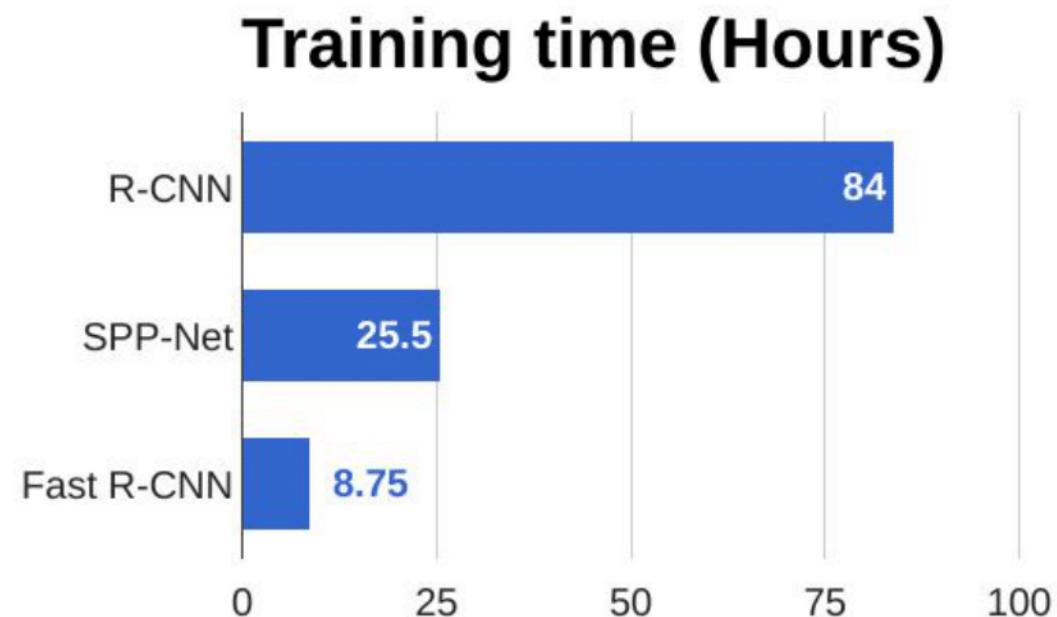
Hi-res conv features:  
512 x 20 x 15;  
  
Projected region proposal is e.g.  
512 x 18 x 8  
(varies per proposal)

RoI conv features:  
512 x 7 x 7  
for region proposal

Fully-connected layers expect  
low-res conv features:  
512 x 7 x 7

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

# R-CNN vs SPP vs Fast R-CNN



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

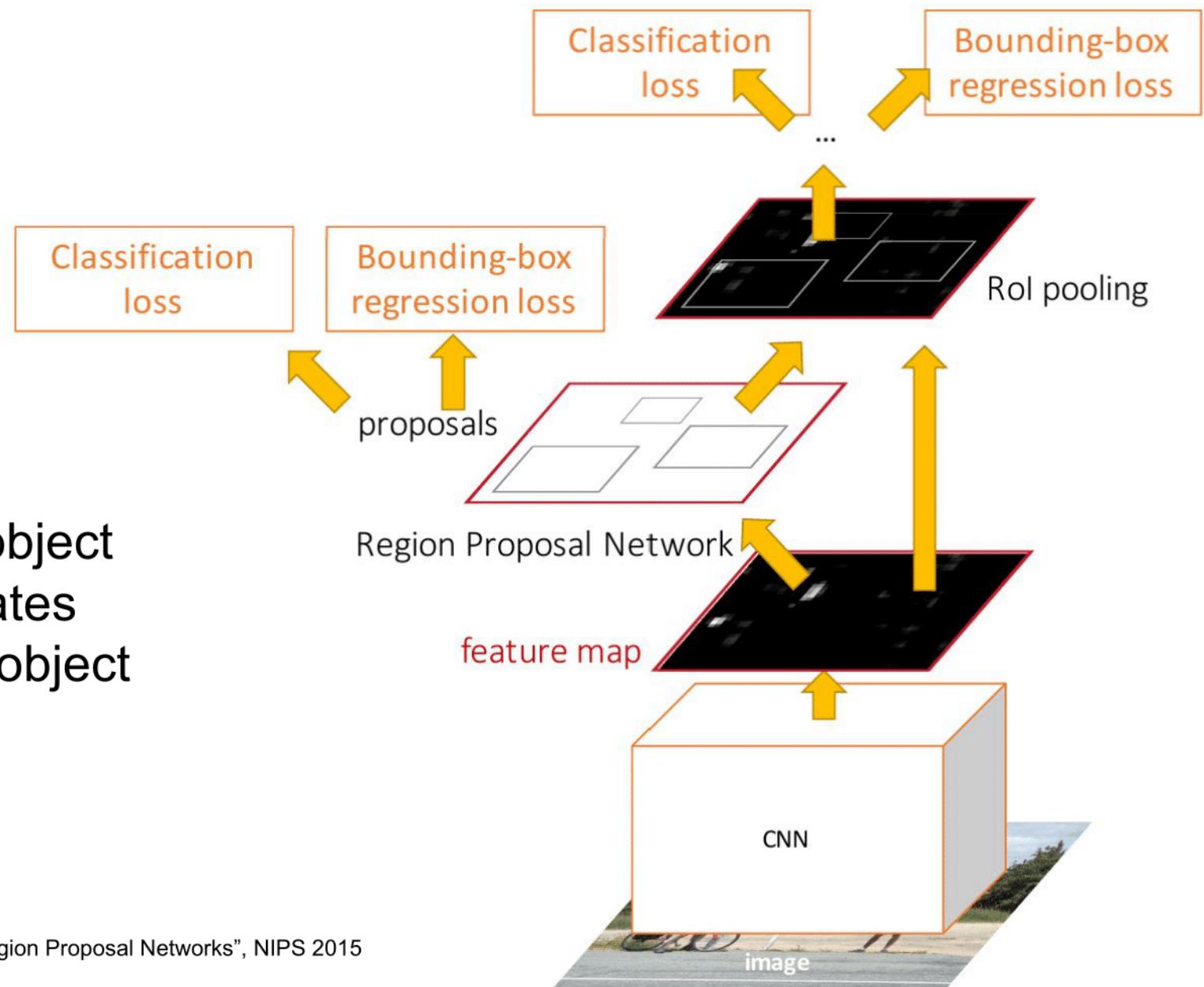
# Faster R-CNN:

Make CNN do proposals!

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

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

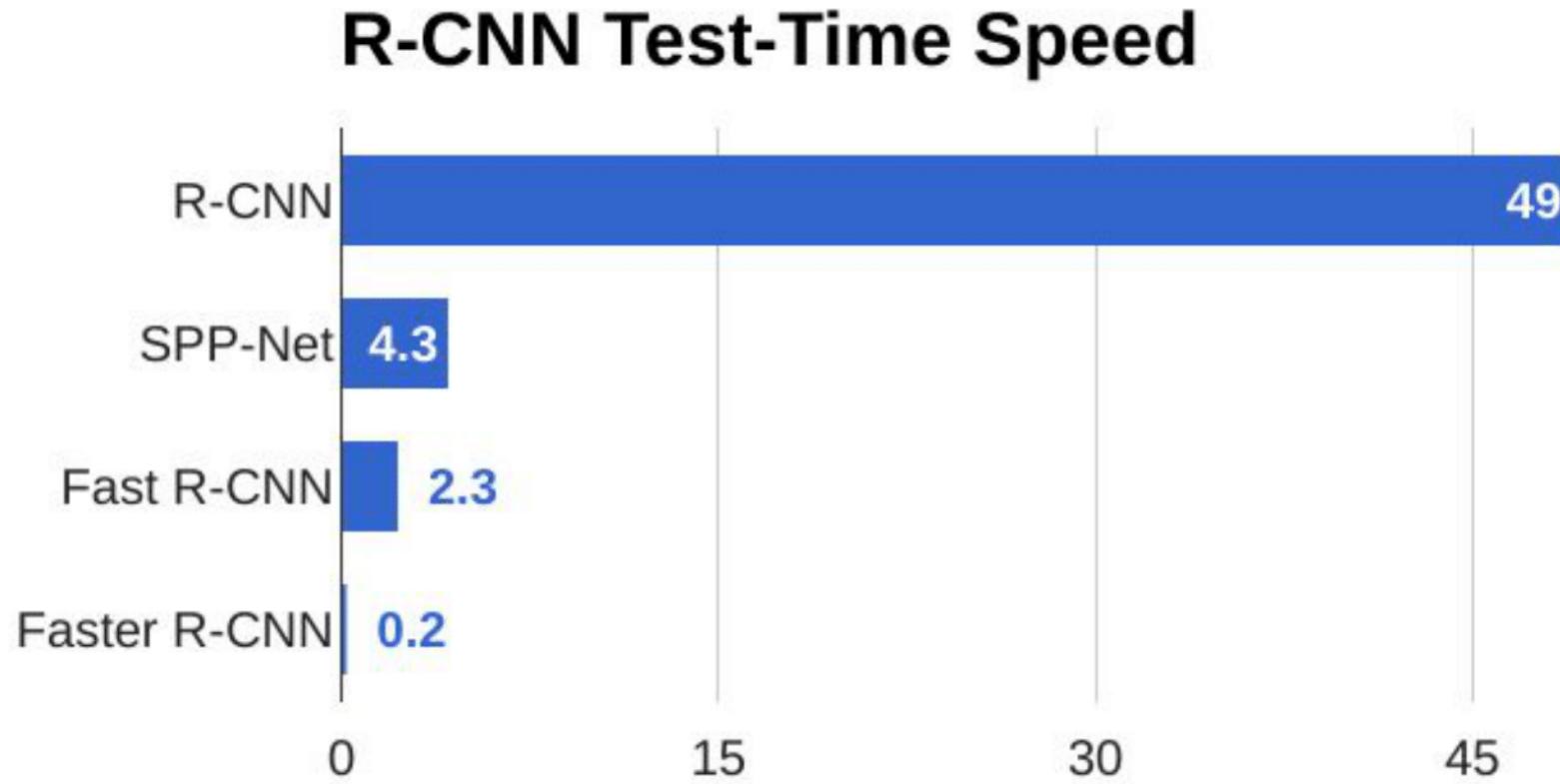


Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015  
Figure copyright 2015, Ross Girshick; reproduced with permission

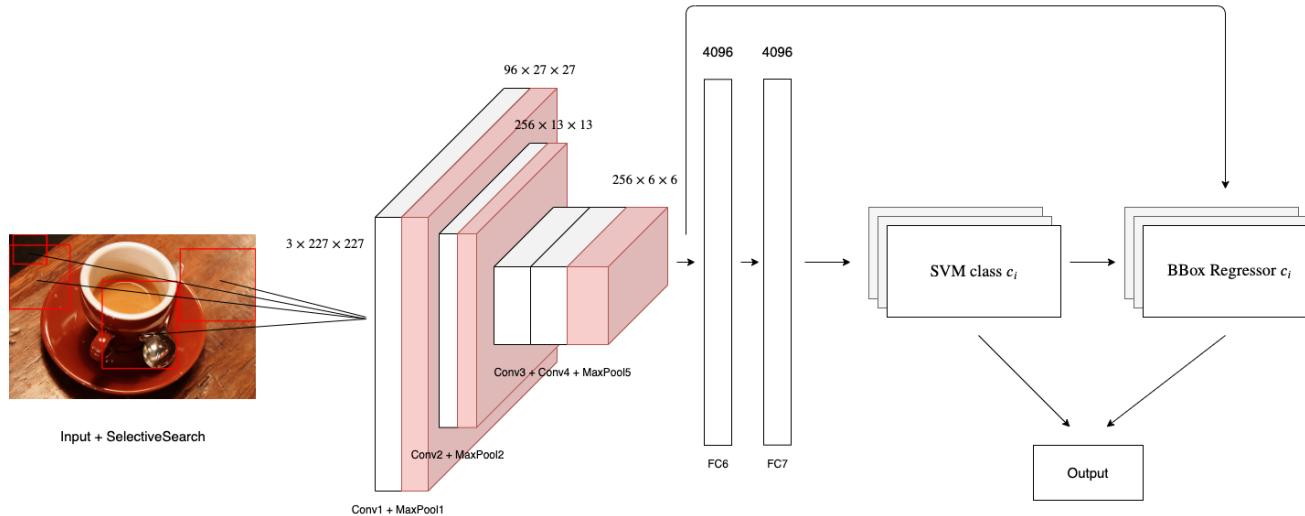
Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

# Faster R-CNN:

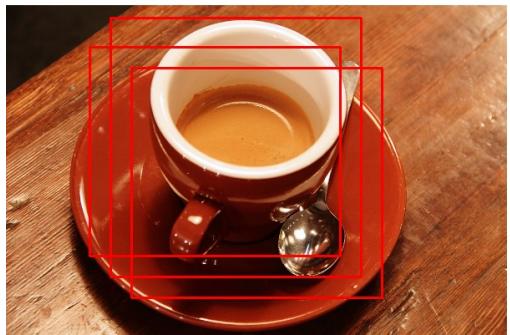
Make CNN do proposals!



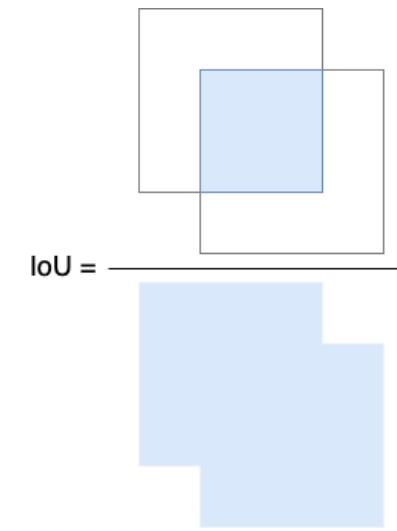
# Results



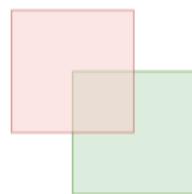
Before non-maximum suppression



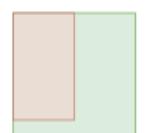
After non-maximum suppression



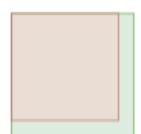
Bad



IoU = 0.44



IoU = 0.77



IoU = 1

Good

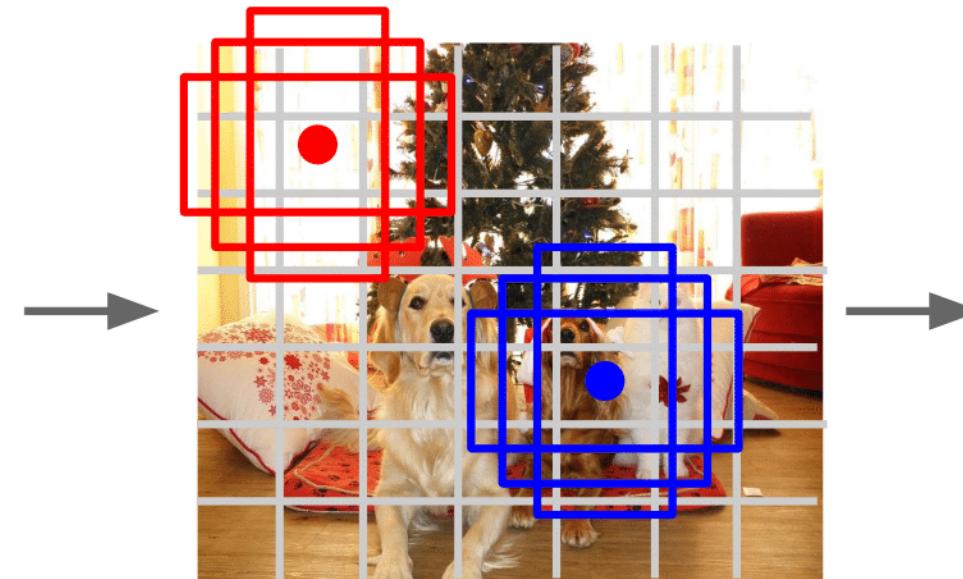


# Detection without Proposals: YOLO / SSD

Go from input image to tensor of scores with one big convolutional network!



Input image  
 $3 \times H \times W$



Divide image into grid  
 $7 \times 7$

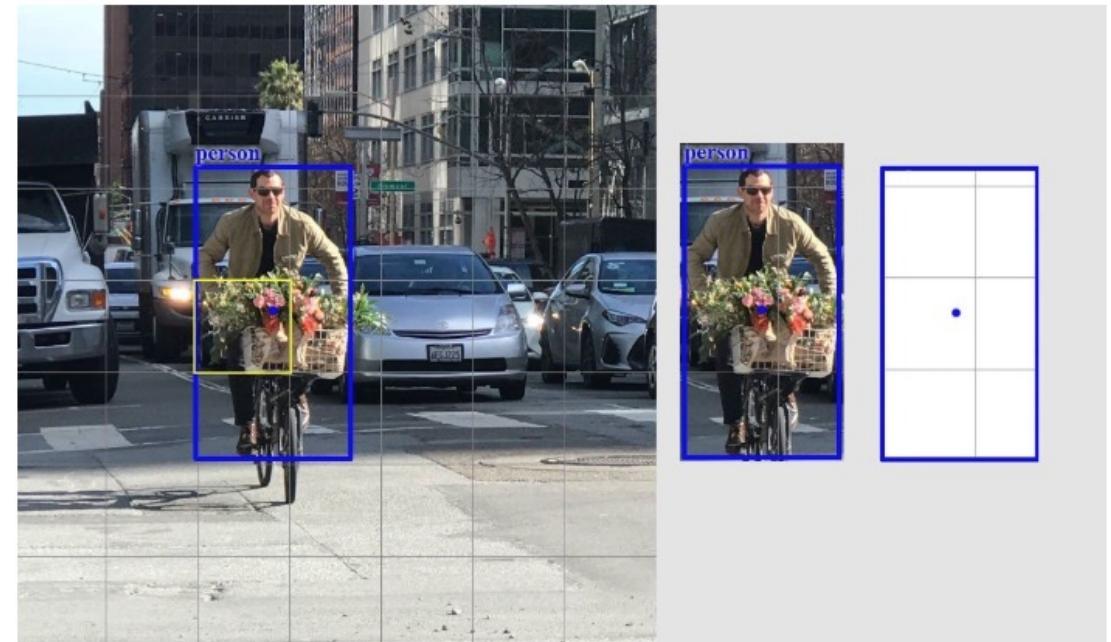
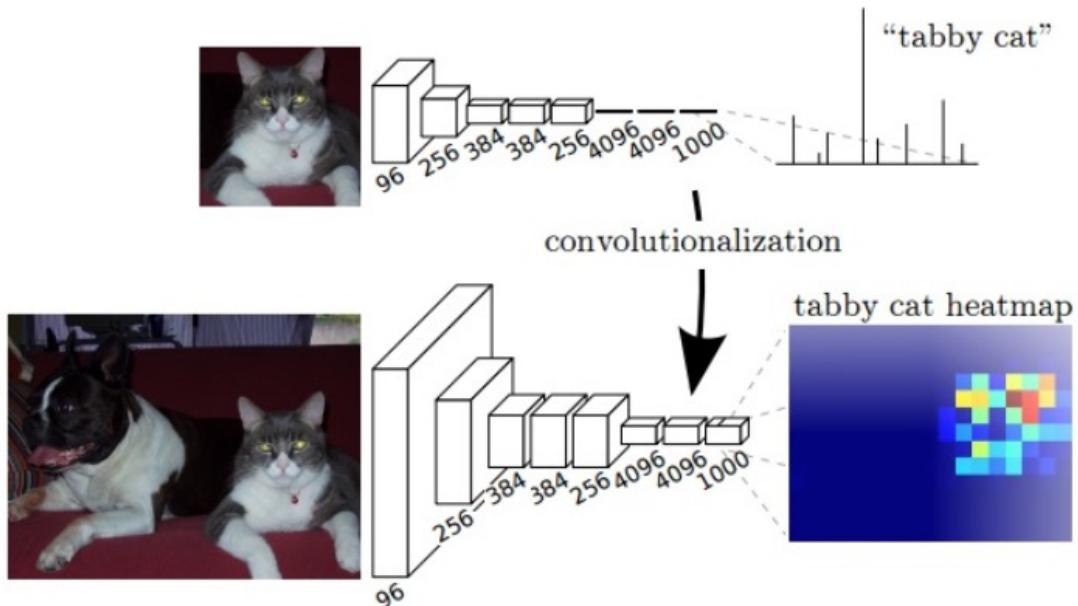
Image a set of **base boxes**  
centered at each grid cell  
Here  $B = 3$

- 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)

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

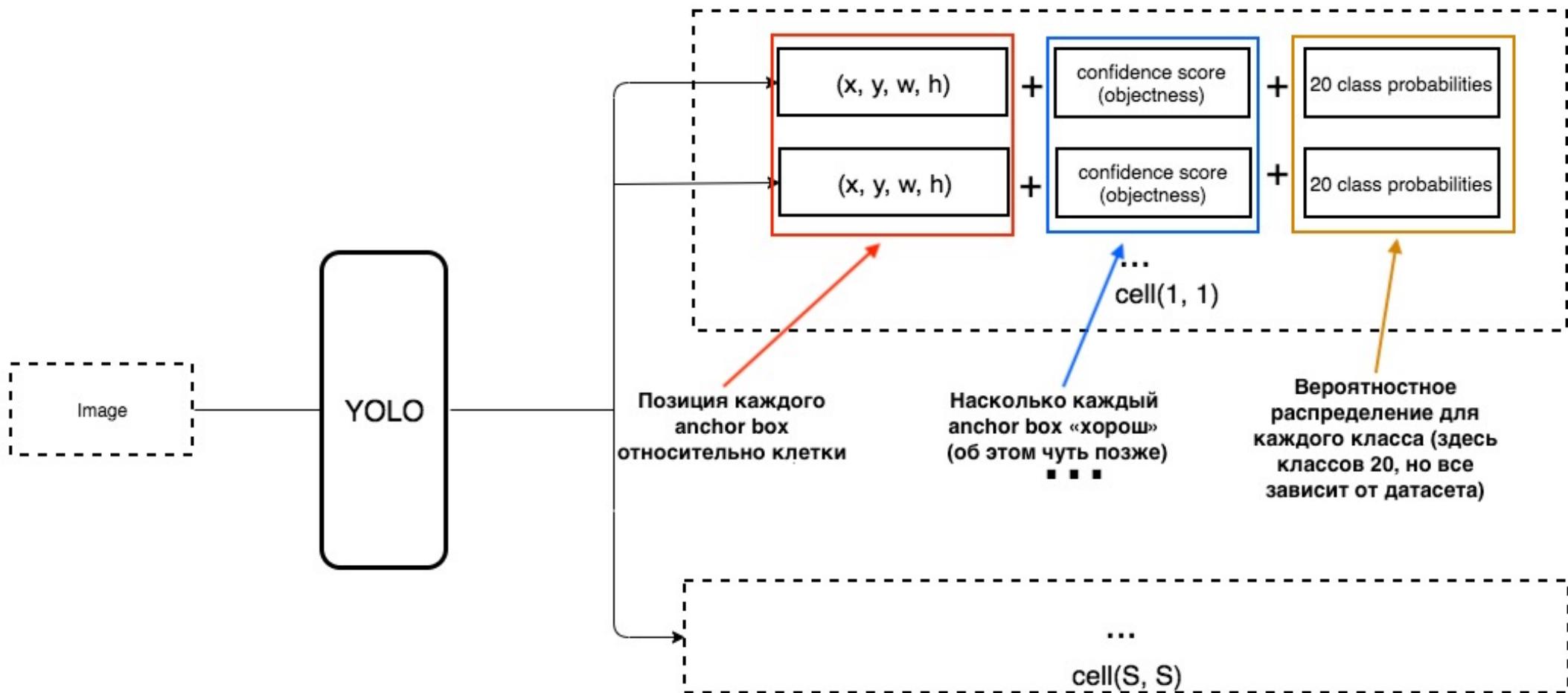
Redmon et al, "You Only Look Once:  
Unified, Real-Time Object Detection", CVPR 2016  
Liu et al, "SSD: Single-Shot MultiBox Detector", ECCV 2016

# One Shot Detection

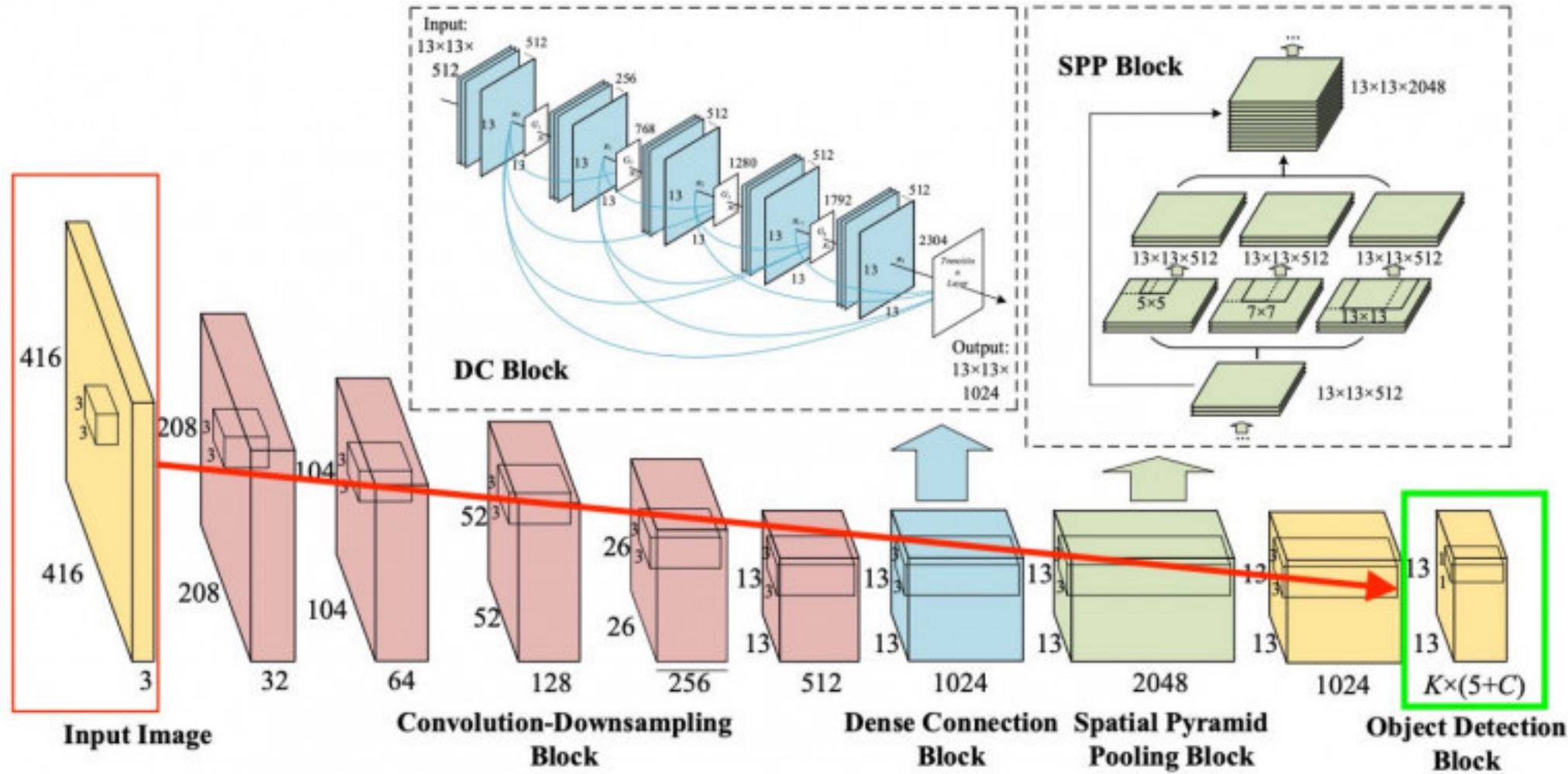


Для каждой ячейки в последнем сопу слое предказываем координаты бокса и класс объекта с центром в ячейке.

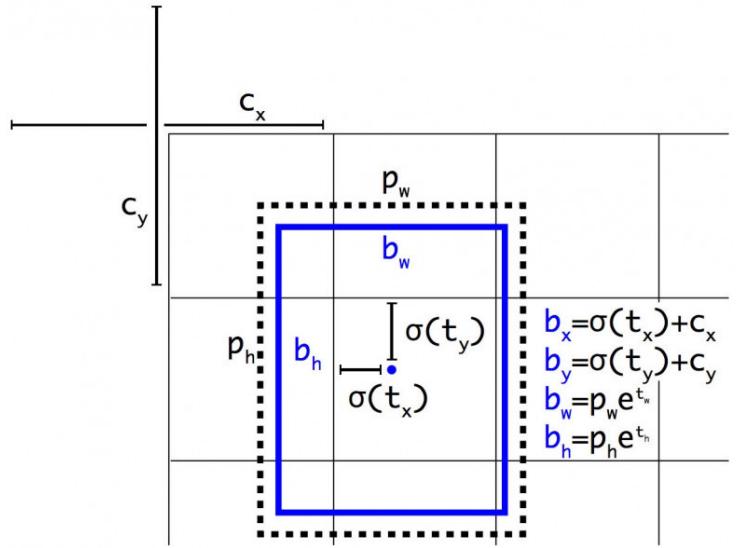
# One Shot Detector: YOLO



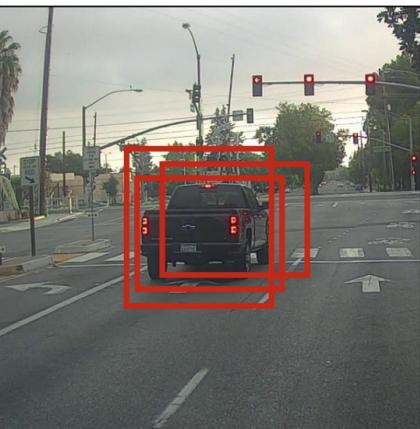
# One Shot Detector: YOLO



# One Shot Detector: YOLO

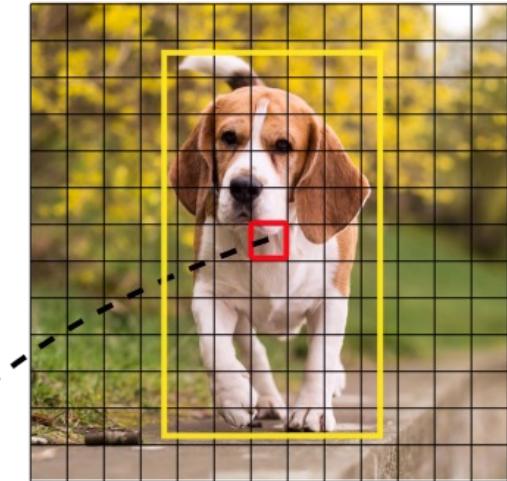


Before non-max suppression

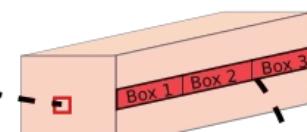


Non-Max  
Suppression  
→

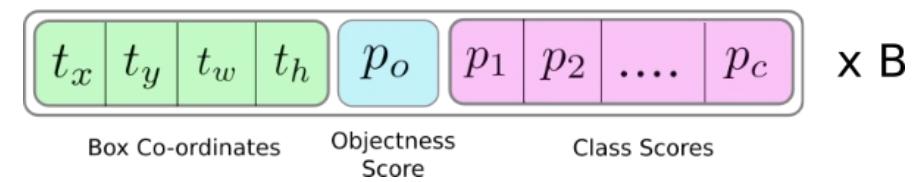
After non-max suppression



Prediction Feature Map

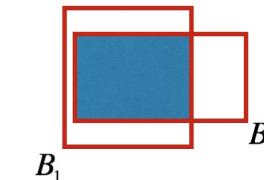


Attributes of a bounding box

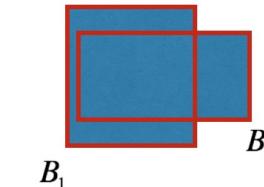


# One Shot Detector: YOLO

Intersection

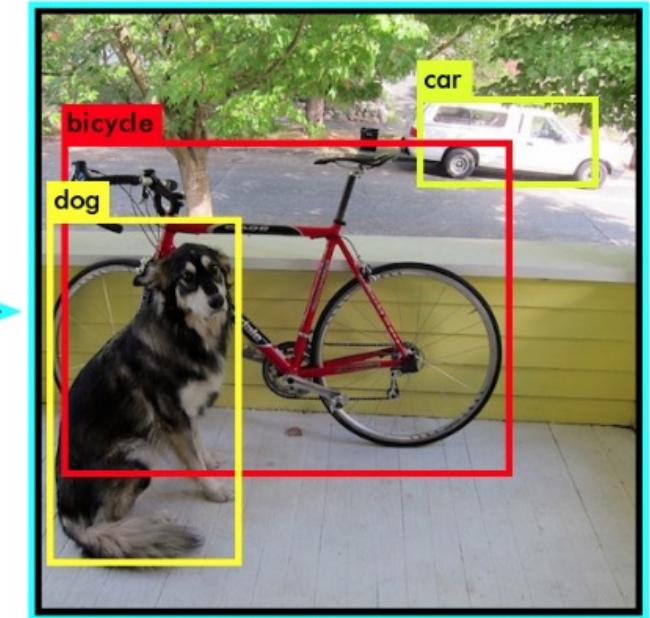
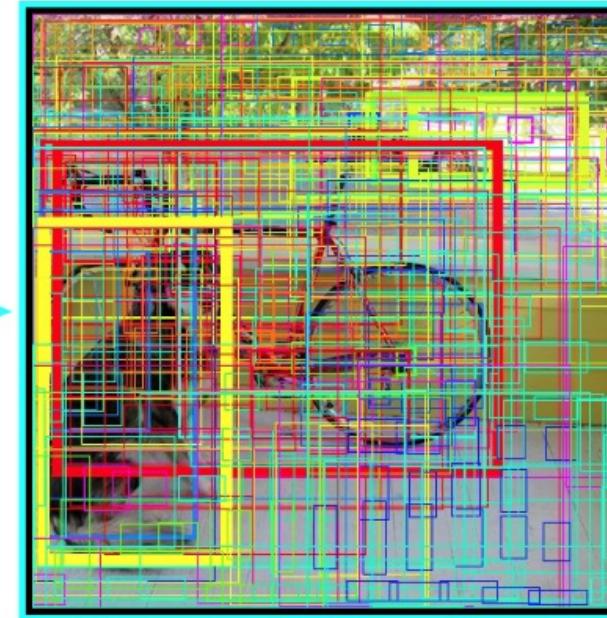
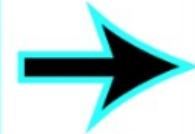
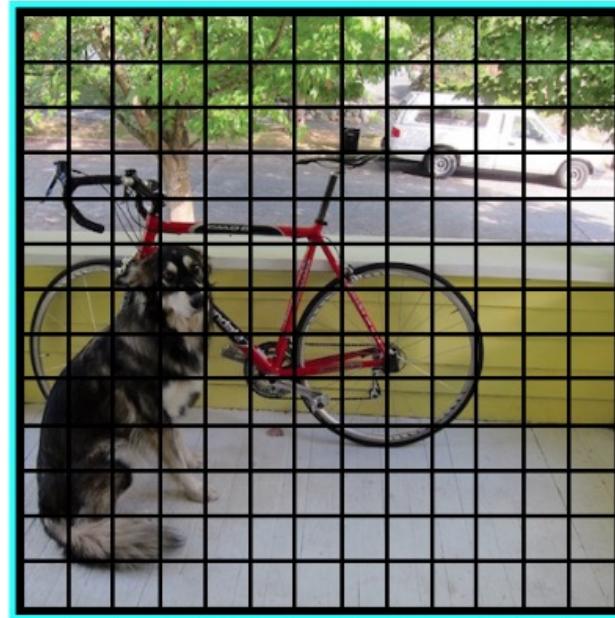


Union



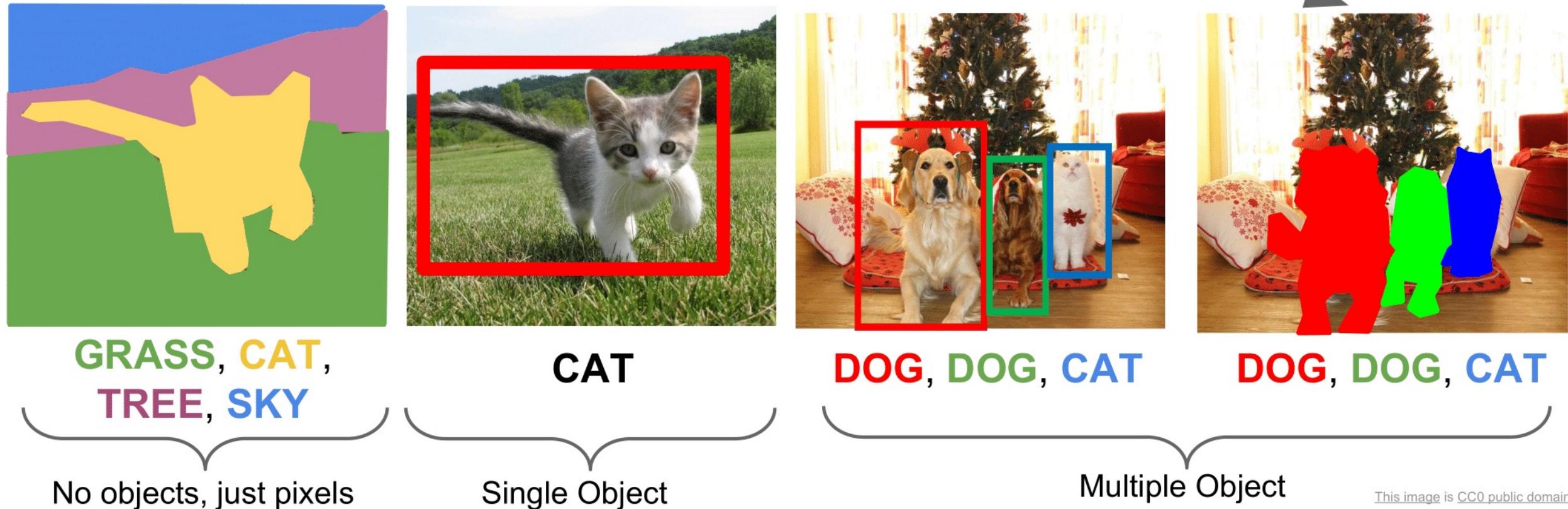
Intersection over Union

$$IoU = \frac{B_1 \cap B_2}{B_1 \cup B_2} = \frac{\text{Area of intersection}}{\text{Area of union}}$$



Для каждой ячейки в последнем сопу слое предказываем координаты бокса и класс объекта с центром в ячейке.

# Instance Segmentation



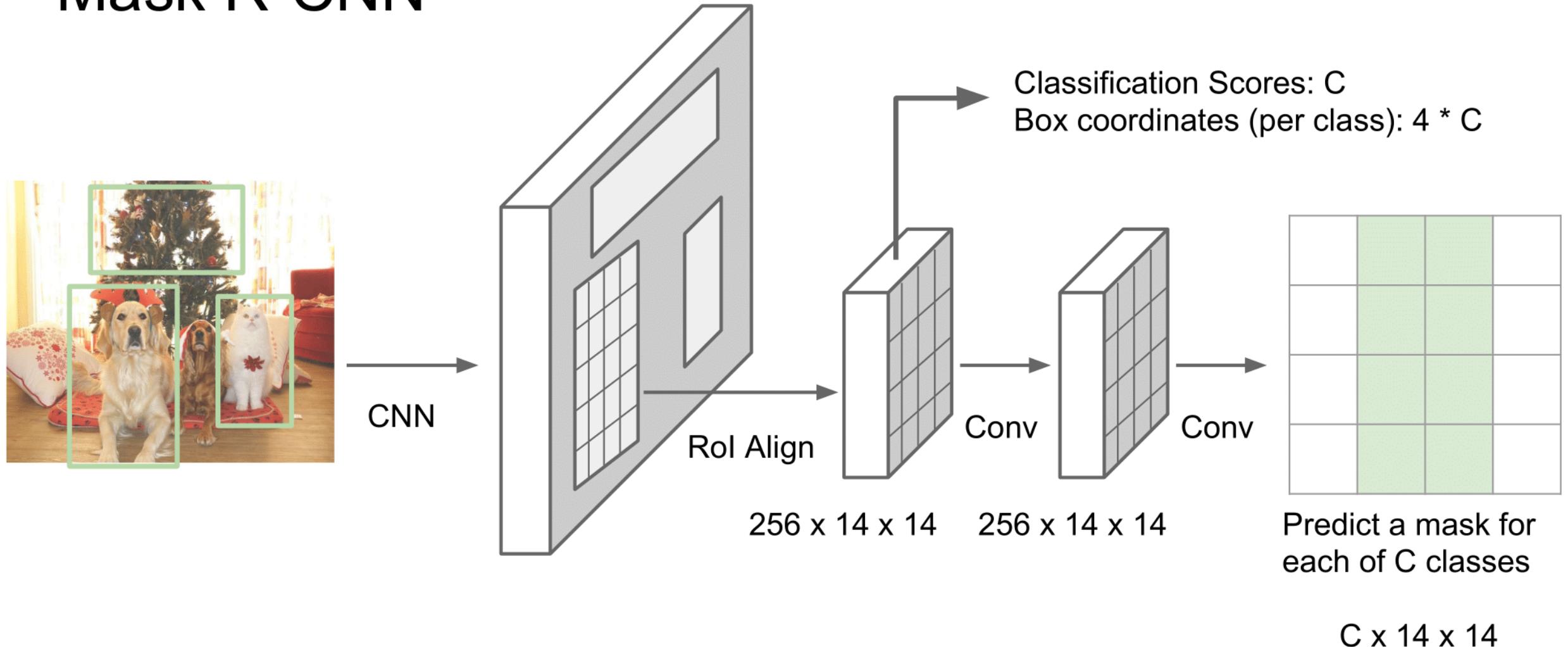
No objects, just pixels

Single Object

Multiple Object

[This image is CC0 public domain](#)

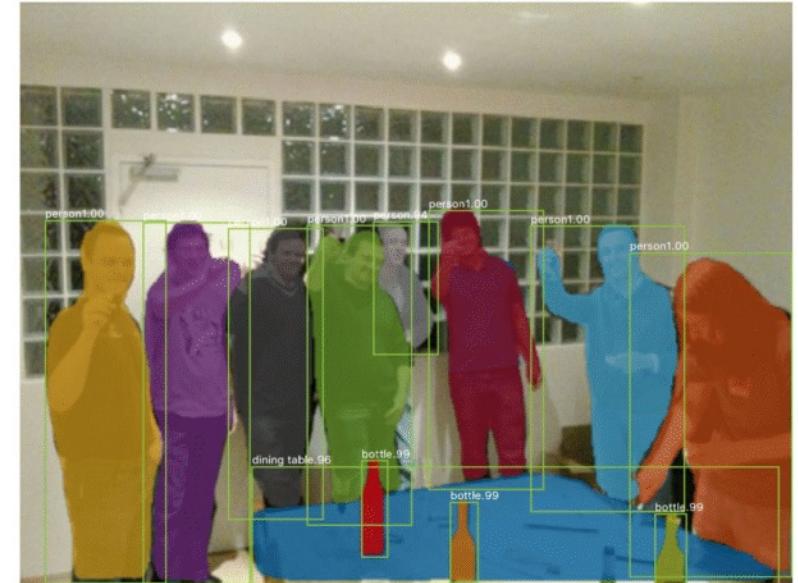
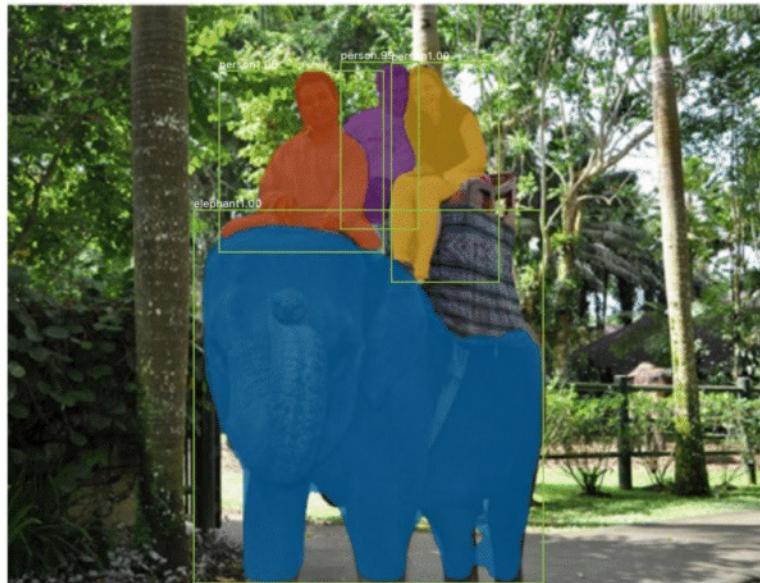
# Mask R-CNN



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

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

# Mask R-CNN: Very Good Results!



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

Figures copyright Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, 2017.

Reproduced with permission.

Source: Stanford CS231n Lecture 11 2017 by Fei-Fei Li & Justin Johnson & Serena Yeung

# Recap:

## Semantic Segmentation



GRASS, CAT,  
TREE, SKY

No objects, just pixels

## Classification + Localization



CAT

Single Object

## Object Detection



DOG, DOG, CAT

Multiple Object

## Instance Segmentation



DOG, DOG, CAT

[This image is CC0 public domain](#)