Object Detection

Pablo Mesejo

pmesejo@go.ugr.es

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial

UNIVERSIDAD DE GRANADA

DECSAI

DaSCI

Instituto Andaluz de Investigación en

Data Science and **Computational Intelligence**

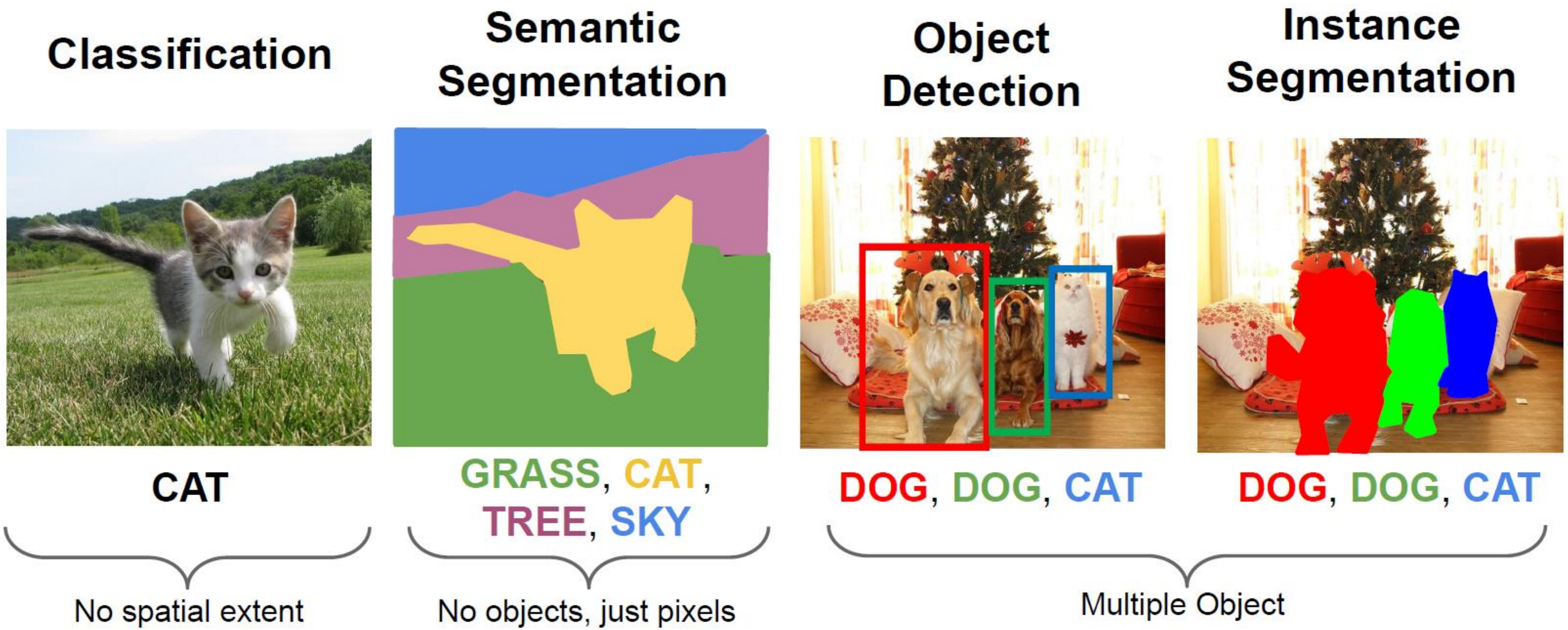
Readings

- Szeliski (2022), Chapter 6.3.
- Zhang, Lipton, Li and Smola (2023), *Dive into Deep Learning*, Chapter 14.3-14.8.
- Stanford University CS231n (2023): Deep Learning for Computer Vision. Lecture 11.
- Liu et al. (2020). Deep learning for generic object detection: A survey. International journal of computer vision, 128, 261-318.

More classical approaches:

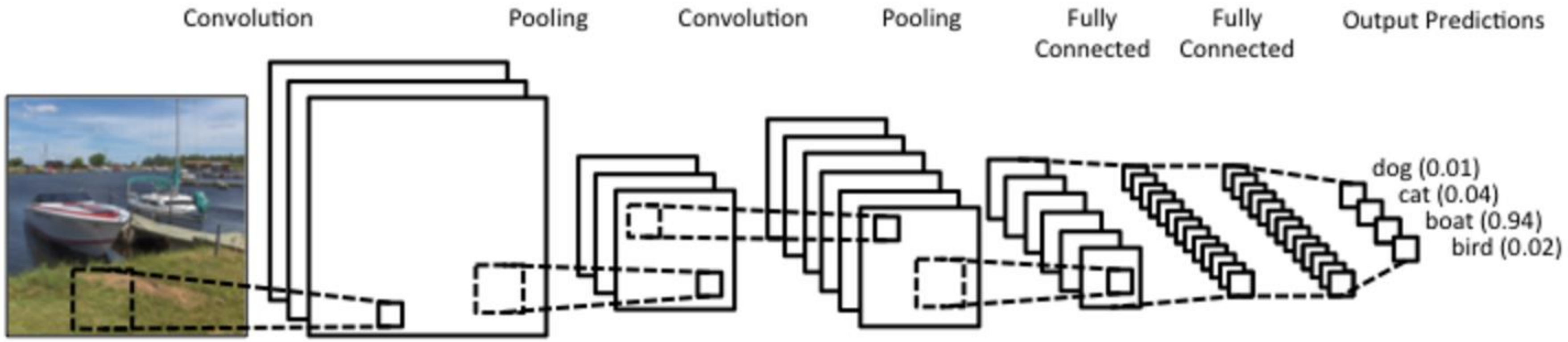
- Forsyth & Ponce (2012). Chapter 17 and 18.

Next slides are mainly based on those from [cs231n]


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

From image classification to object detection

Image classification



Object detection

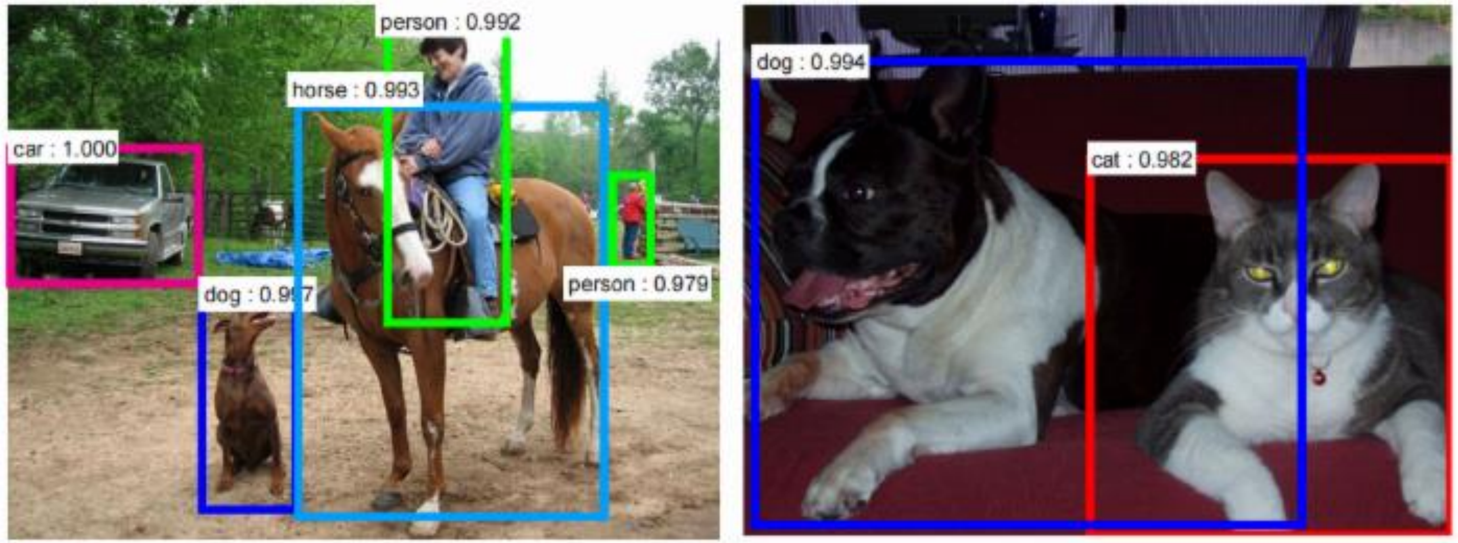
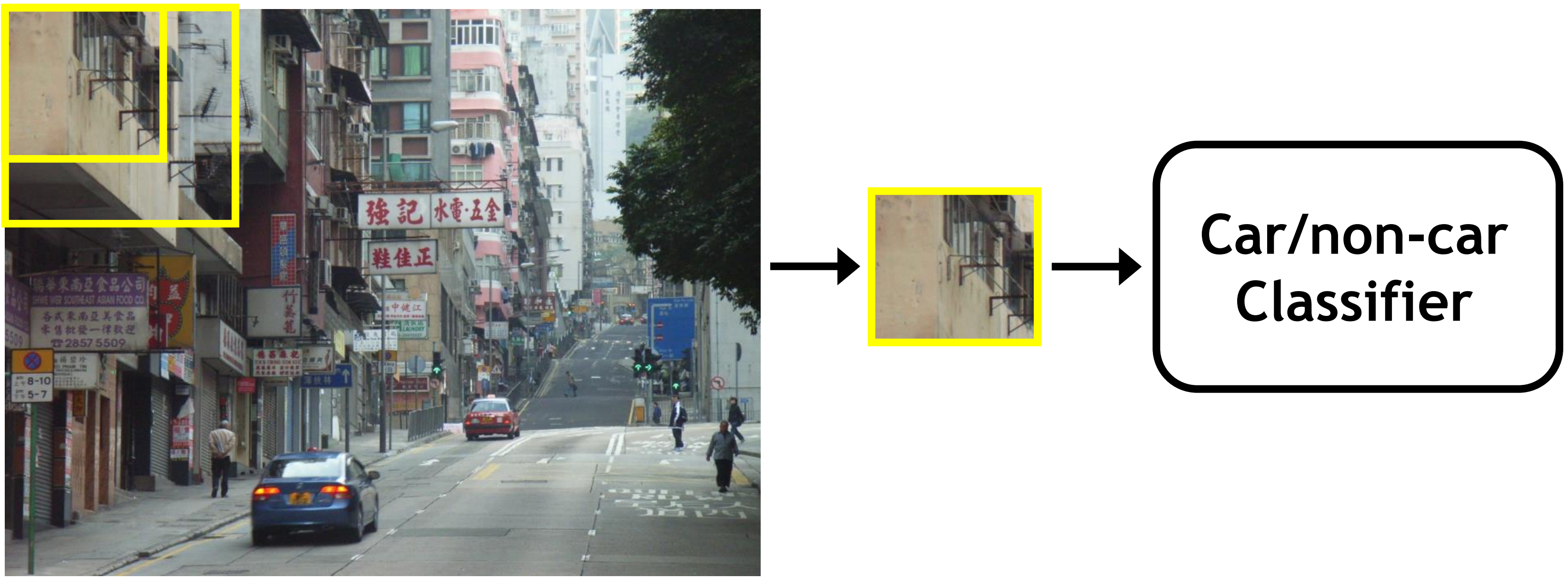


Image source

Sliding Window Detection

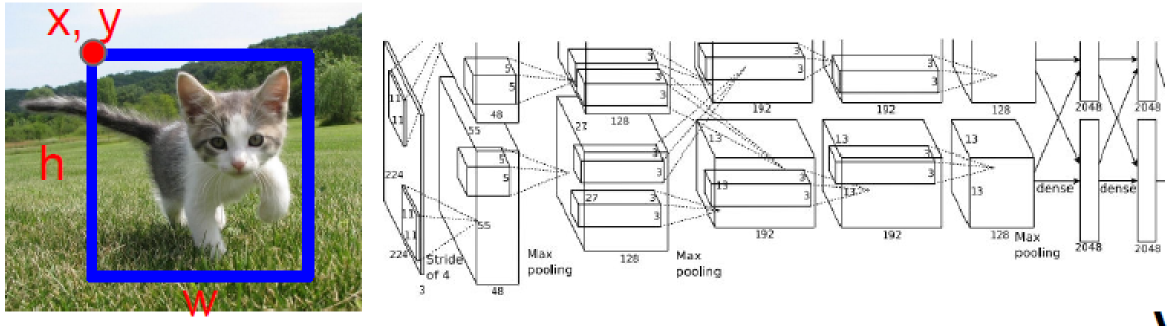


- Slide a window across the image and evaluate a detection model at each location
  - Thousands of windows to evaluate: **efficiency** and **low false positive rates** are essential
  - **Difficult to extend to a large range of scales, aspect ratios**

Kristen Grauman

Object Detection: Single Object

(Classification + Localization)



Vector: 4096

Fully Connected: 4096 to 1000

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

Correct label: Cat

Softmax Loss

Fully Connected: 4096 to 4

Box Coordinates (x, y, w, h)

L2 Loss

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

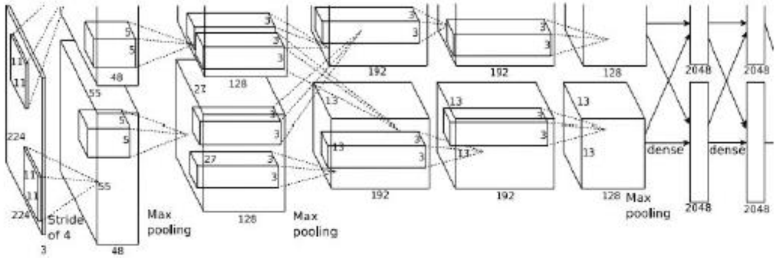
Multitask Loss

\+ → **Loss**

Treat localization as a regression problem!

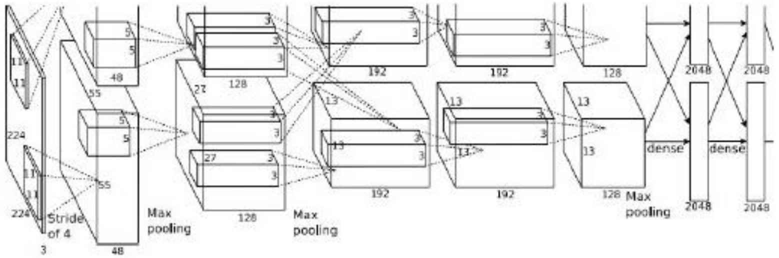
Object Detection: Multiple Objects

Each image needs a different number of outputs!



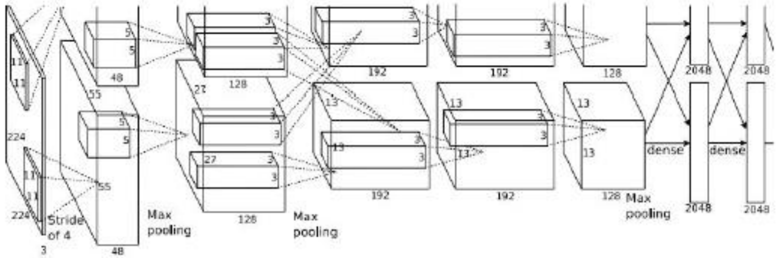
CAT: (x, y, w, h)

4 numbers



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

12 numbers

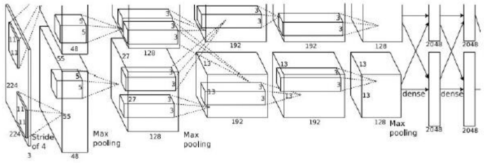


DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
….

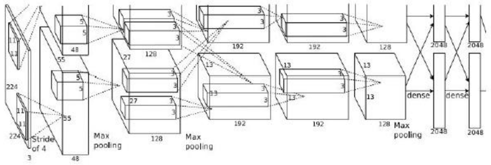
Many numbers!

Object Detection: Multiple Objects

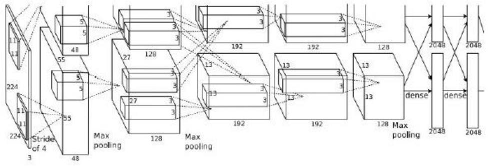
- Possible solution: Apply a ConvNet to many different image crops → ConvNet classifies each crop as object or background



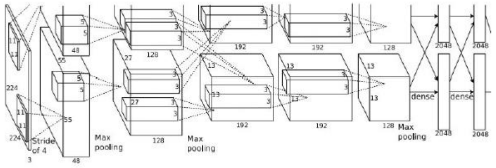
Dog? NO
Cat? NO
Background? YES



Dog? YES
Cat? NO
Background? NO



Dog? YES
Cat? NO
Background? NO



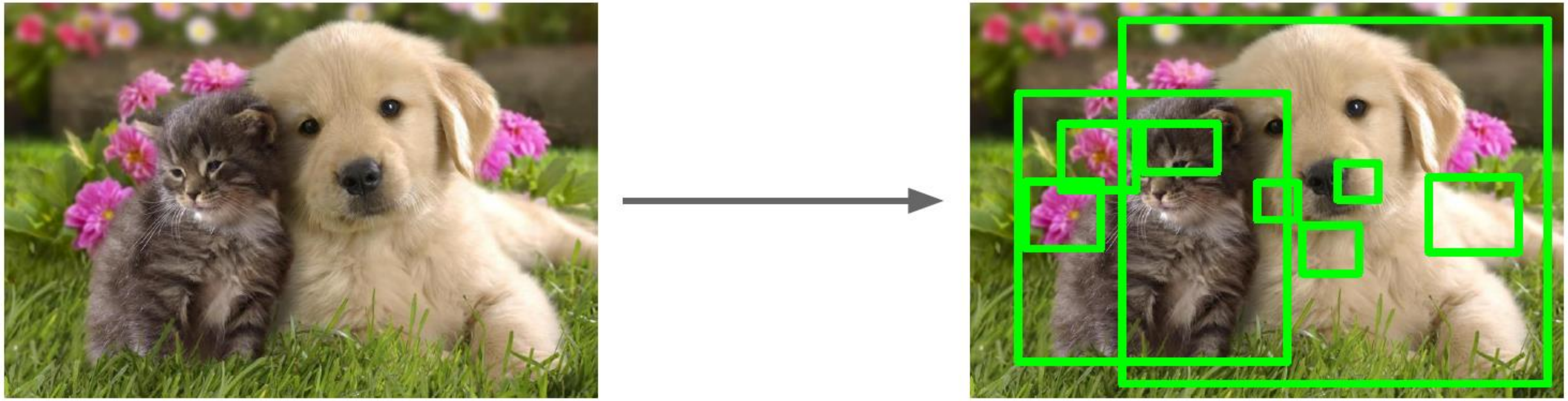
Dog? NO
Cat? YES
Background? NO

What's the problem with this approach?

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

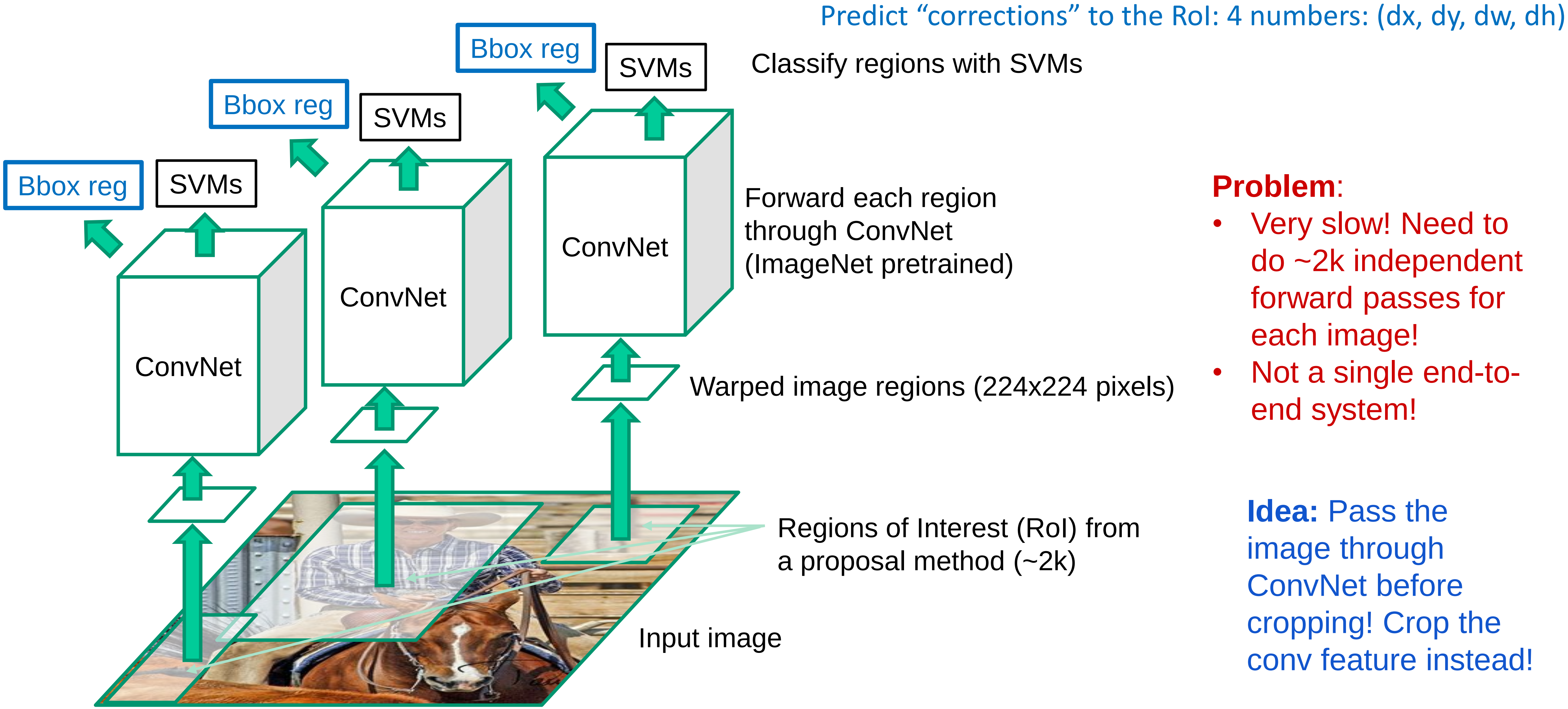
Region Proposals: Selective Search

- 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

R-CNN

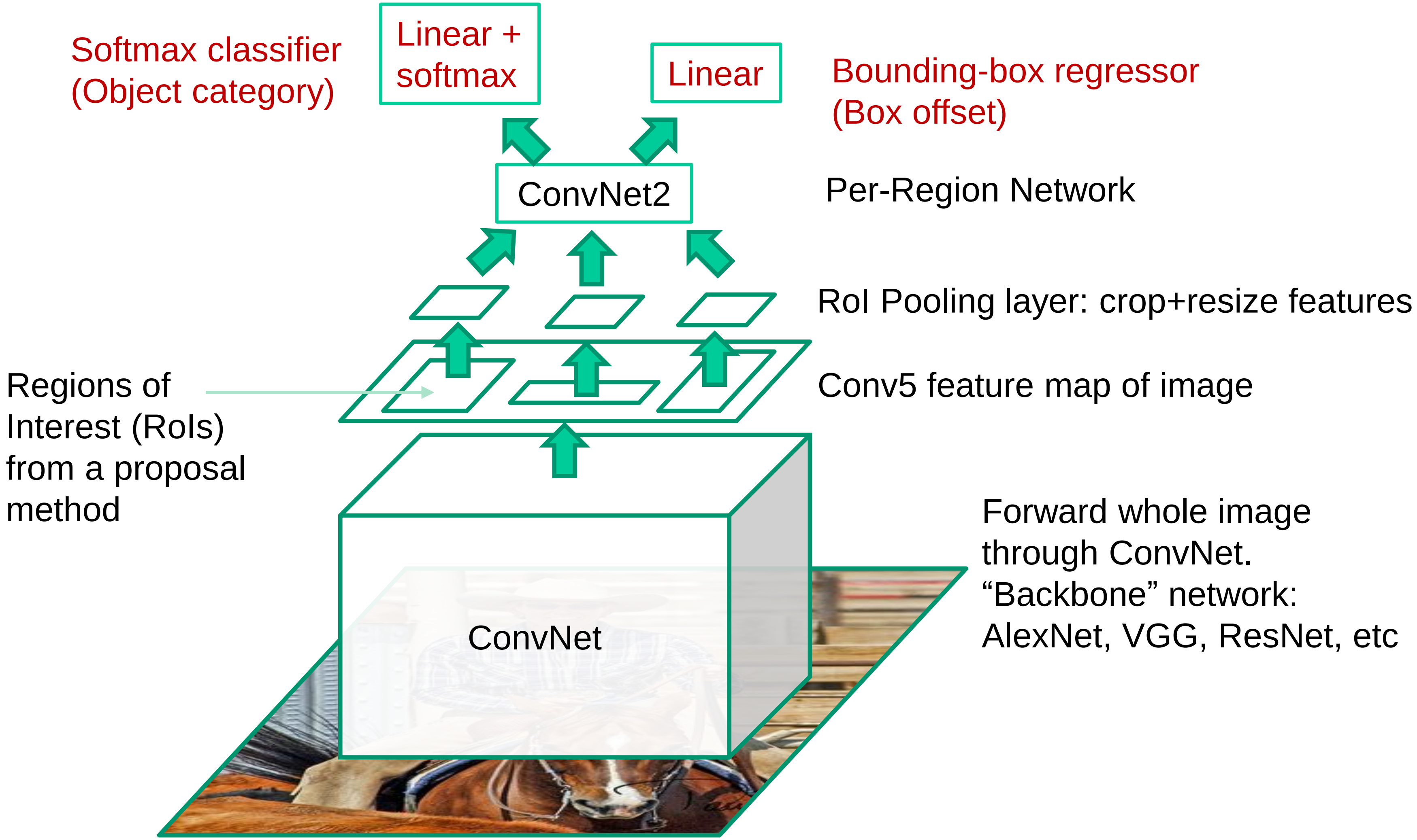


Problem:
- Very slow! Need to do ~2k independent forward passes for each image!
- Not a single end-to-end system!

Idea: Pass the image through ConvNet before cropping! Crop the conv feature instead!

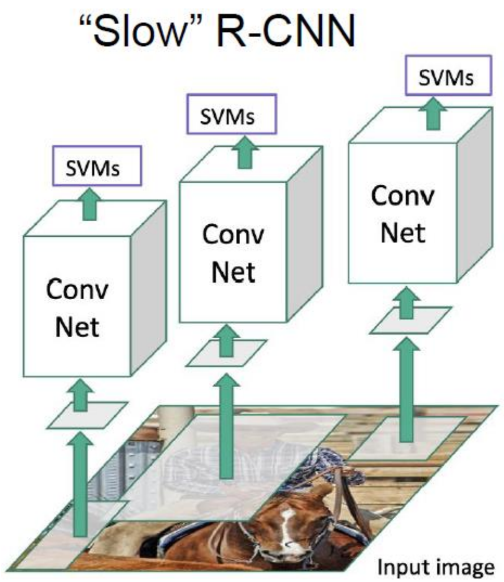
Girshick et al., "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

Fast R-CNN



Problem:
Runtime dominated
by region proposals!

Idea: Make ConvNet
do proposals directly!

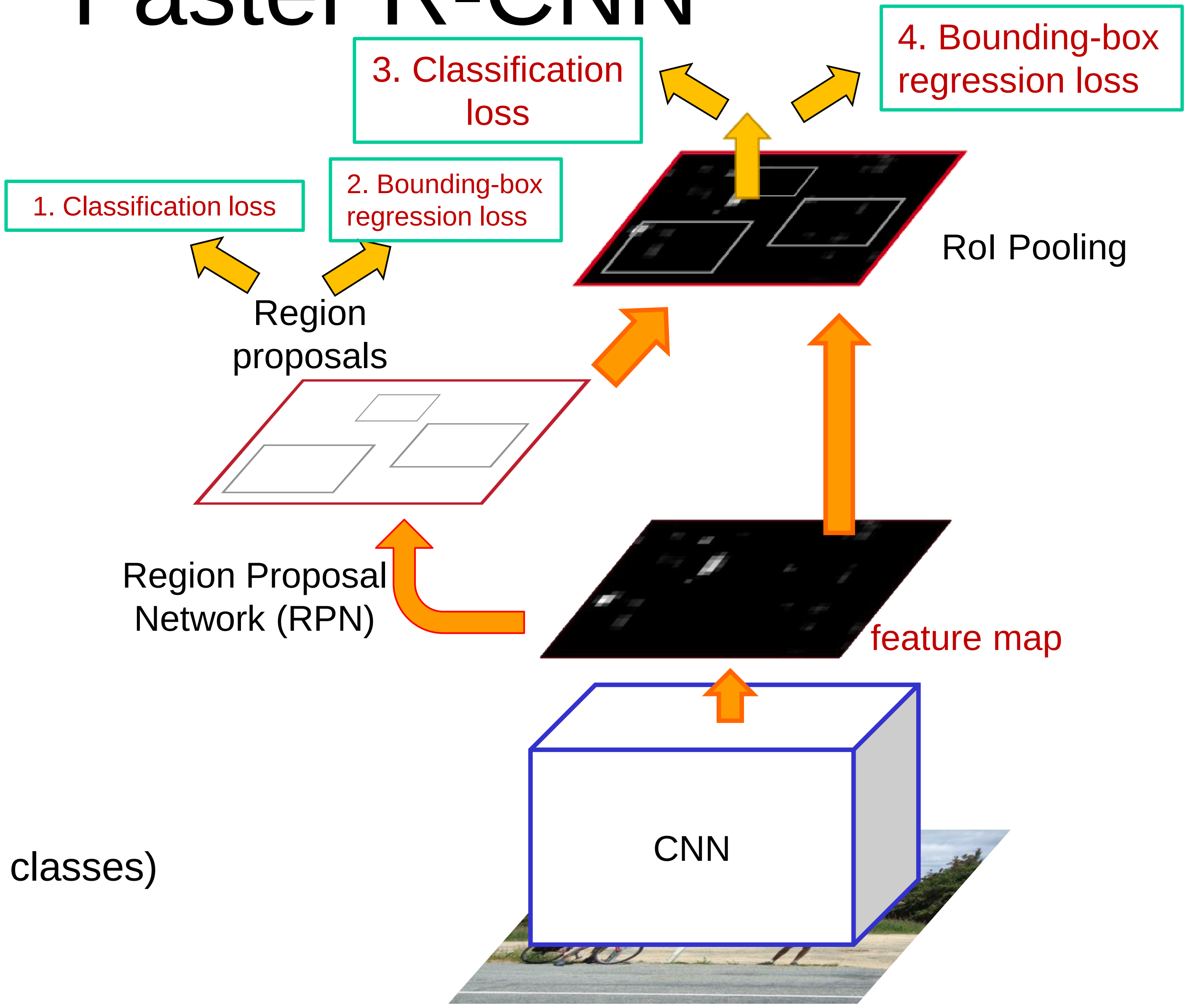


Girshick et al., "Fast R-CNN", ICCV 2015.

Faster R-CNN

Make ConvNet do proposals!

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



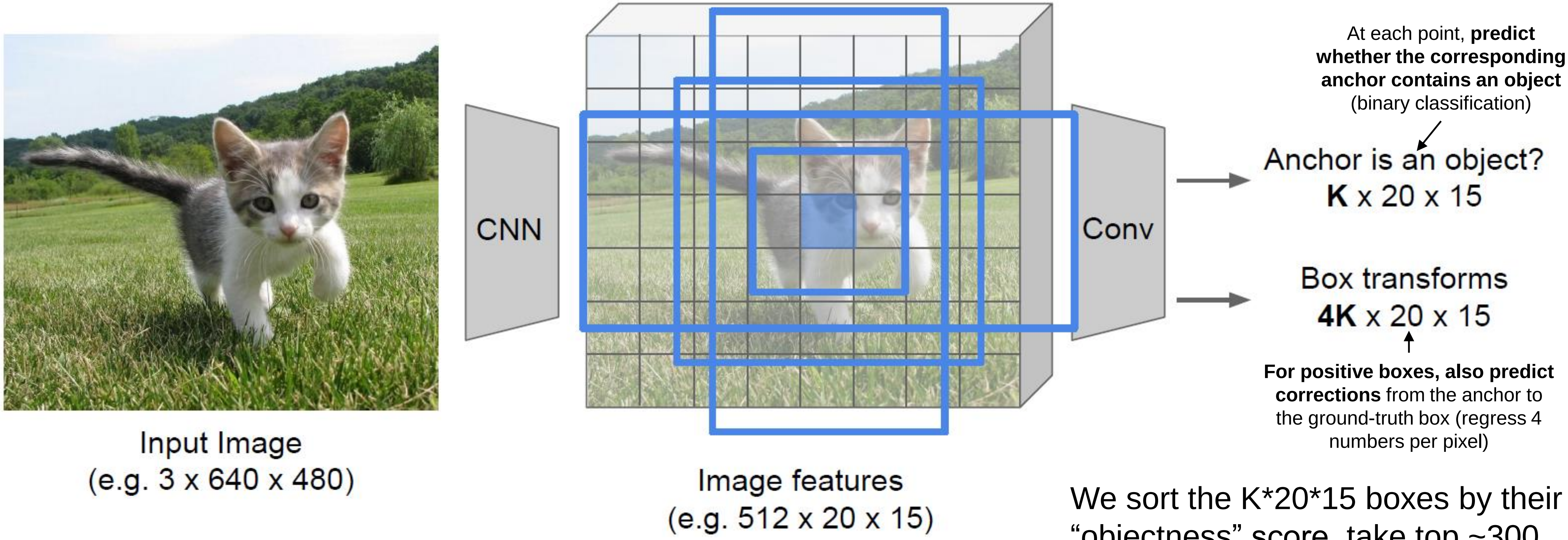
Jointly train with 4 losses:
1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates

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

Region Proposal Network (RPN)

We use an **anchor box** (1x20x15) of fixed size at each point in the feature map

We use **K** different anchor boxes of different size / scale at each point



We sort the K*20*15 boxes by their “objectness” score, take top ~300 as our proposals

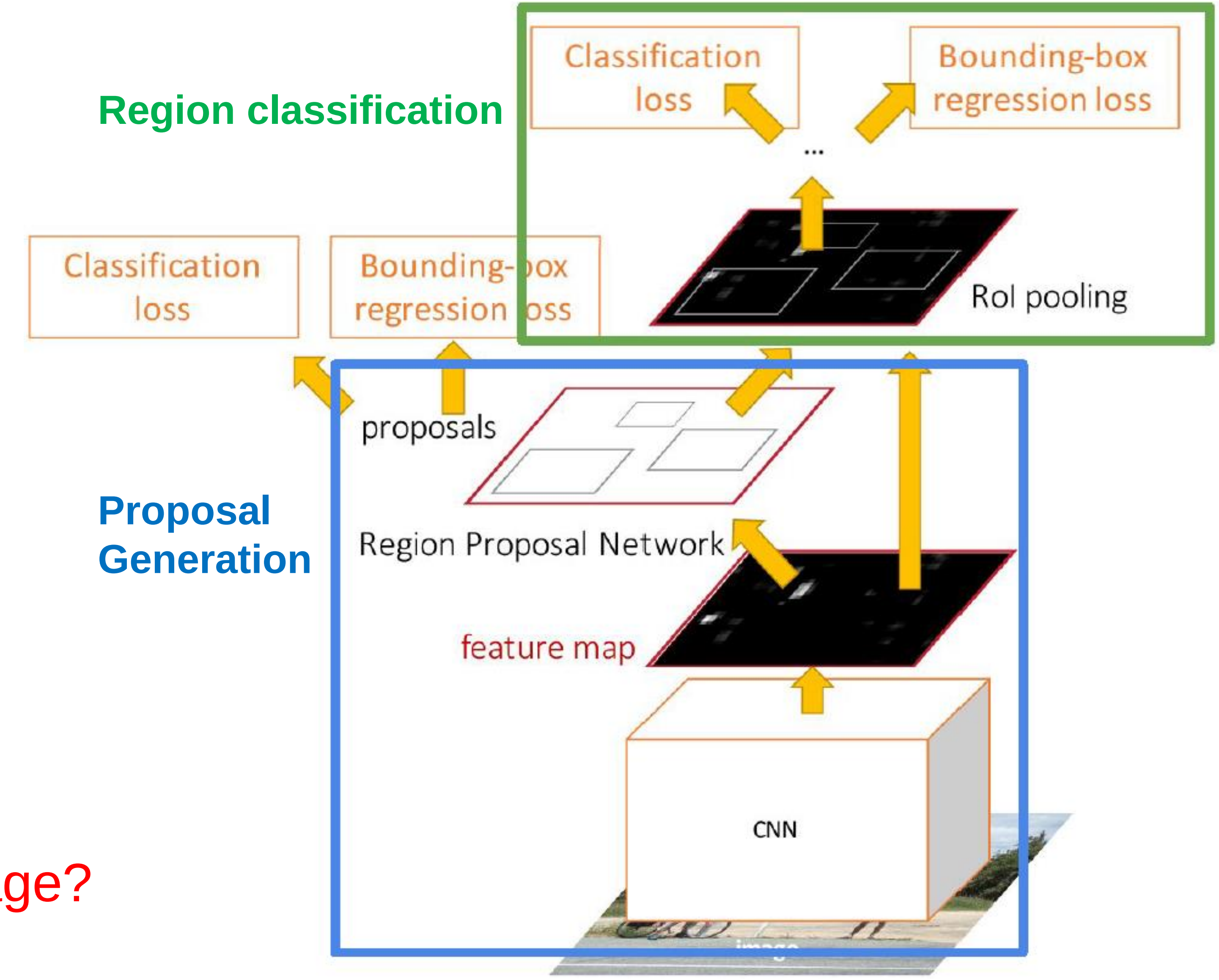
Faster R-CNN

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?



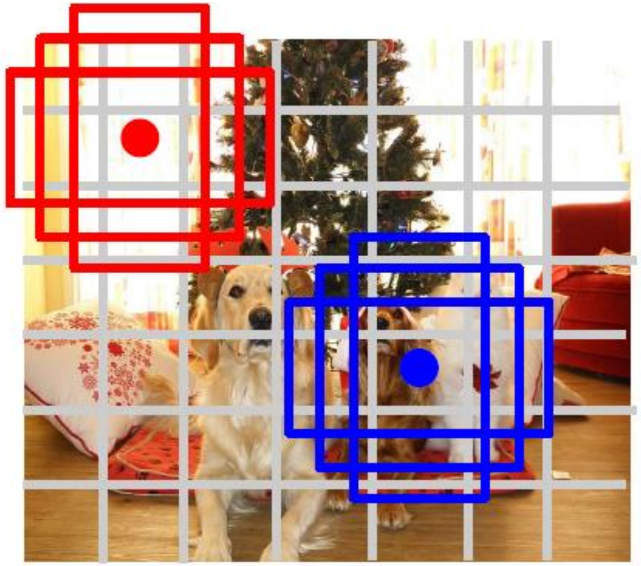
Ren et al., "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

Single-Stage Object Detectors: YOLO / SSD / RetinaNet

In fact, YOLO means “You Only Look Once”…

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

Input image
3 x H x W



Divide image into grid
7 x 7

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

Output: 7 x 7 x (5 * B + C) numbers

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

Object Detection: Lots of Variables…

Backbone Network
VGG
ResNet
Inception
EfficientNet
…

"Meta-Architecture"
Two-stage: Faster R-CNN
Single-stage: YOLO / SSD

Methodological choices
- # Region Proposals
- How are anchors determined?
- How do we sample positive/negative samples for training the RPN?
- …

Some of the main takeaways
Faster R-CNN is slower but more accurate

SSD is much faster but not as accurate

Deeper backbones work better

Huang et al, "Speed/accuracy trade-offs for modern convolutional object detectors", CVPR 2017
Dai et al, "R-FCN: Object Detection via Region-based Fully Convolutional Networks", NIPS 2016

Object detection evaluation

- At test time, predict bounding boxes, class labels, and confidence scores
- For each detection, determine whether it is a TP or FP
  - Common criterion: Area(GT ∩ Det) / Area(GT ∪ Det) > 0.5
  - For multiple detections of the same GT box, only one considered as a TP

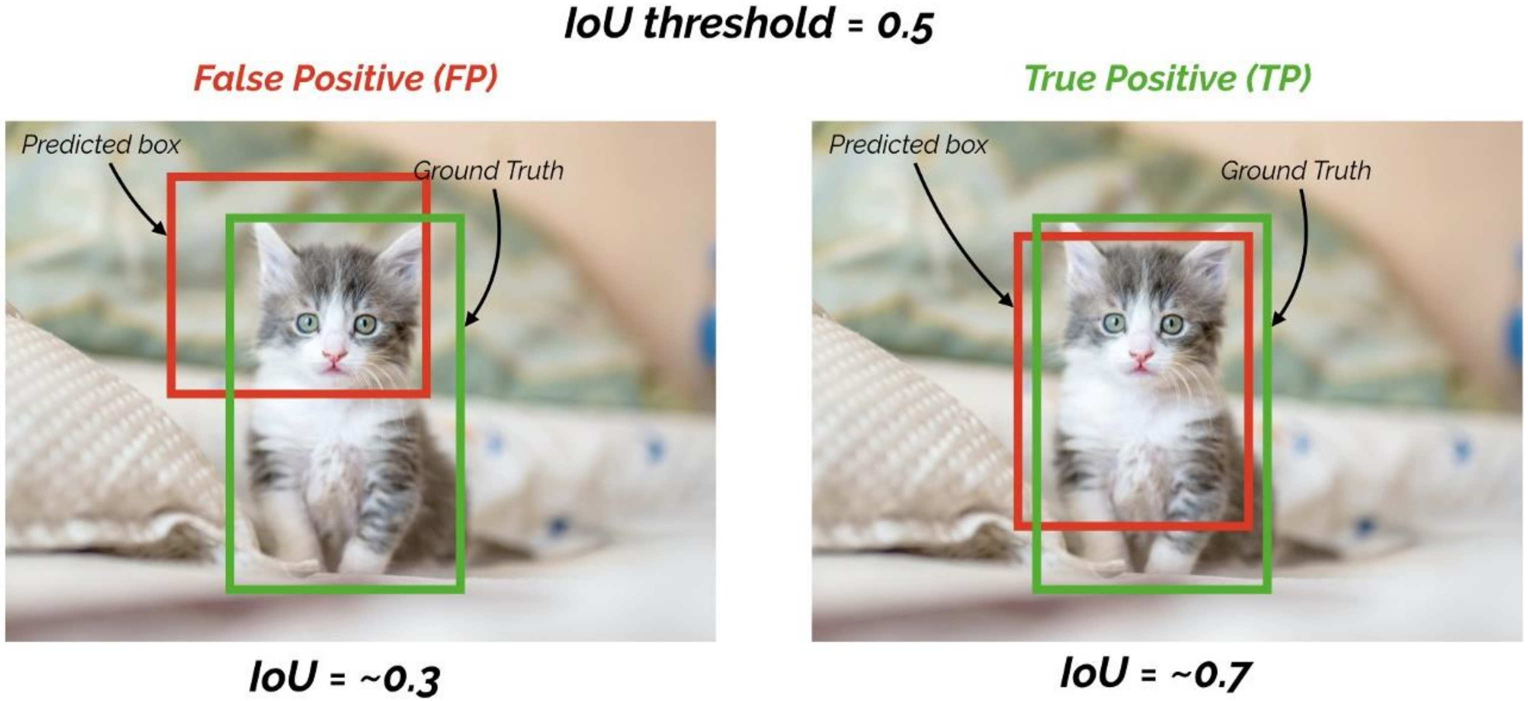


Image Source

Non-Maxima Suppression (NMS)

What if same object is detected multiple times?
NMS eliminates some candidates that are in fact different detections of the same object.

We eliminate boxes that overlap significantly with a higher scoring bounding box

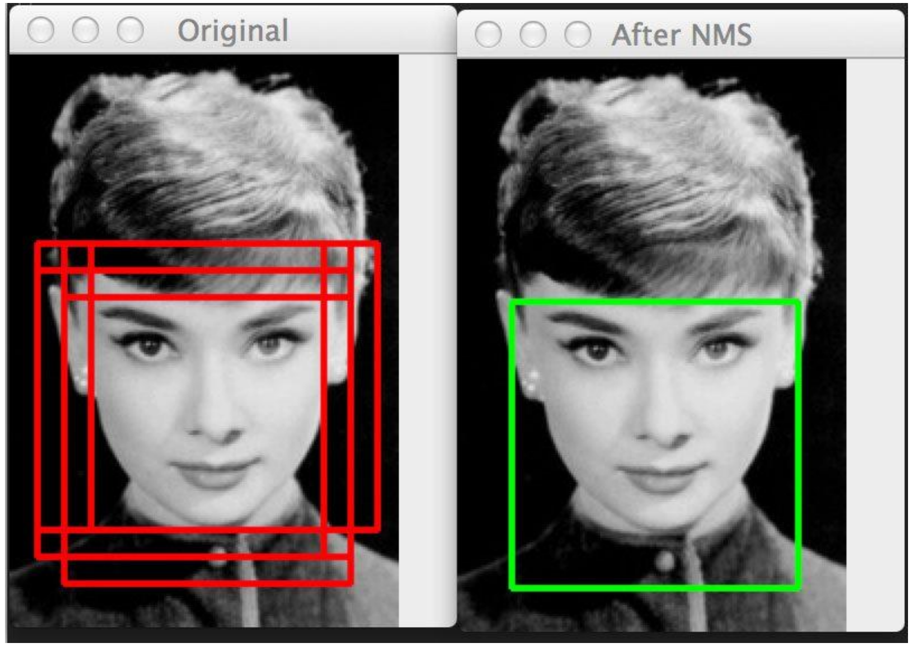


Image Source

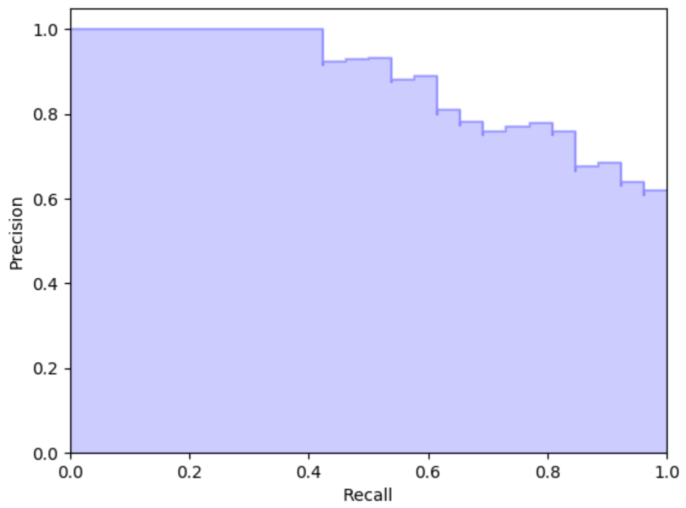
For each object class c:

1. Discard all boxes with $p_c \leq 0.6$

2. While there are remaining boxes:
   i. Keep the box with the largest p_c.
   ii. Discard any remaining box with IoU > 0.5 with the box selected in (i)

Object detection evaluation

- For each class, plot Recall-Precision curve and compute Average Precision (area under the curve)
- Take mean of AP over classes to get mAP



Precision:
true positive detections / total detections
TP / (TP+FP)

Recall:
true positive detections / total positive test instances
TP / (TP+FN)

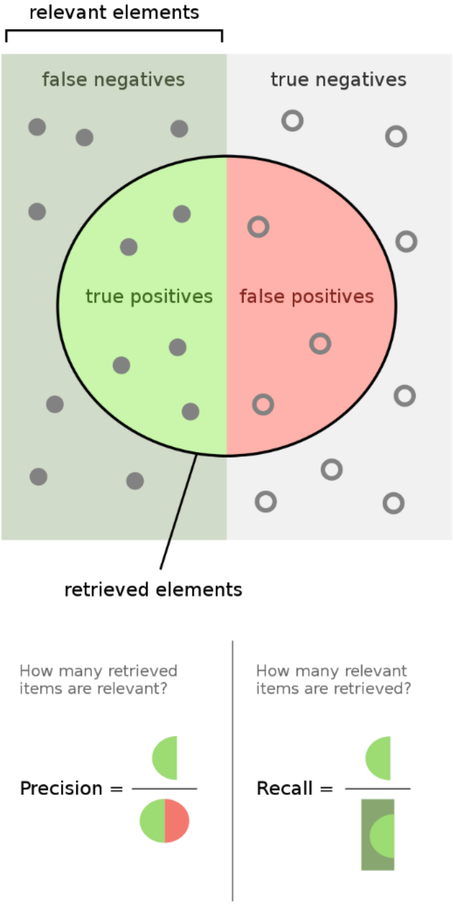
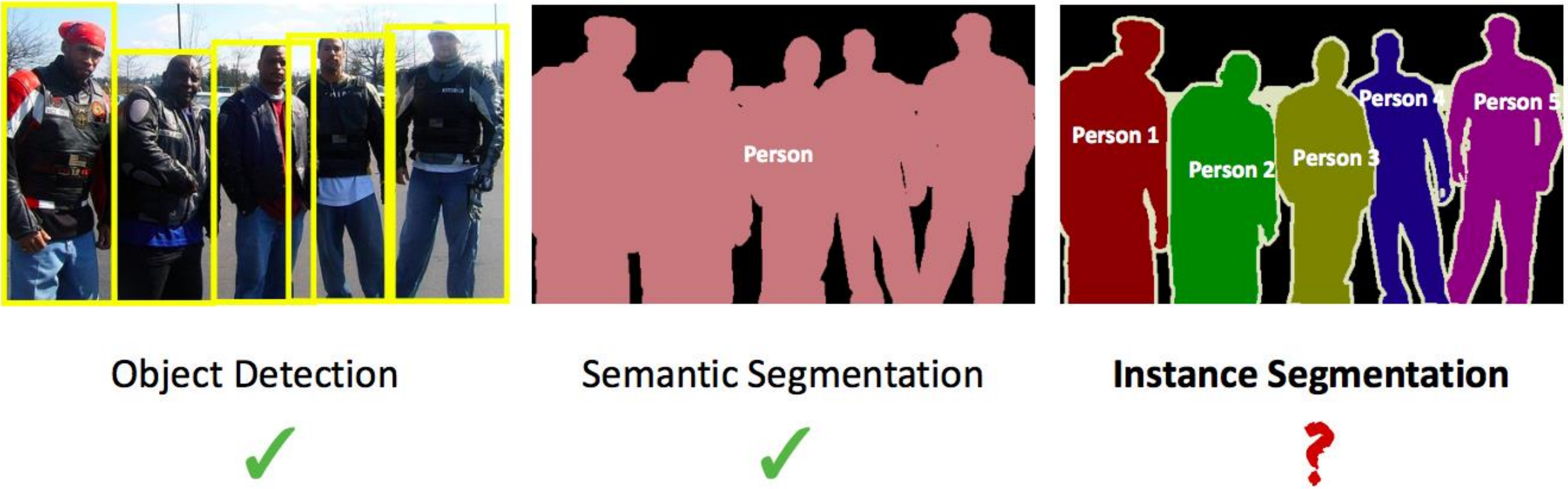


Image Source

An object detector is considered good if its precision stays high as recall increases, which means that **if you vary the confidence threshold, the precision and recall will still be high**.

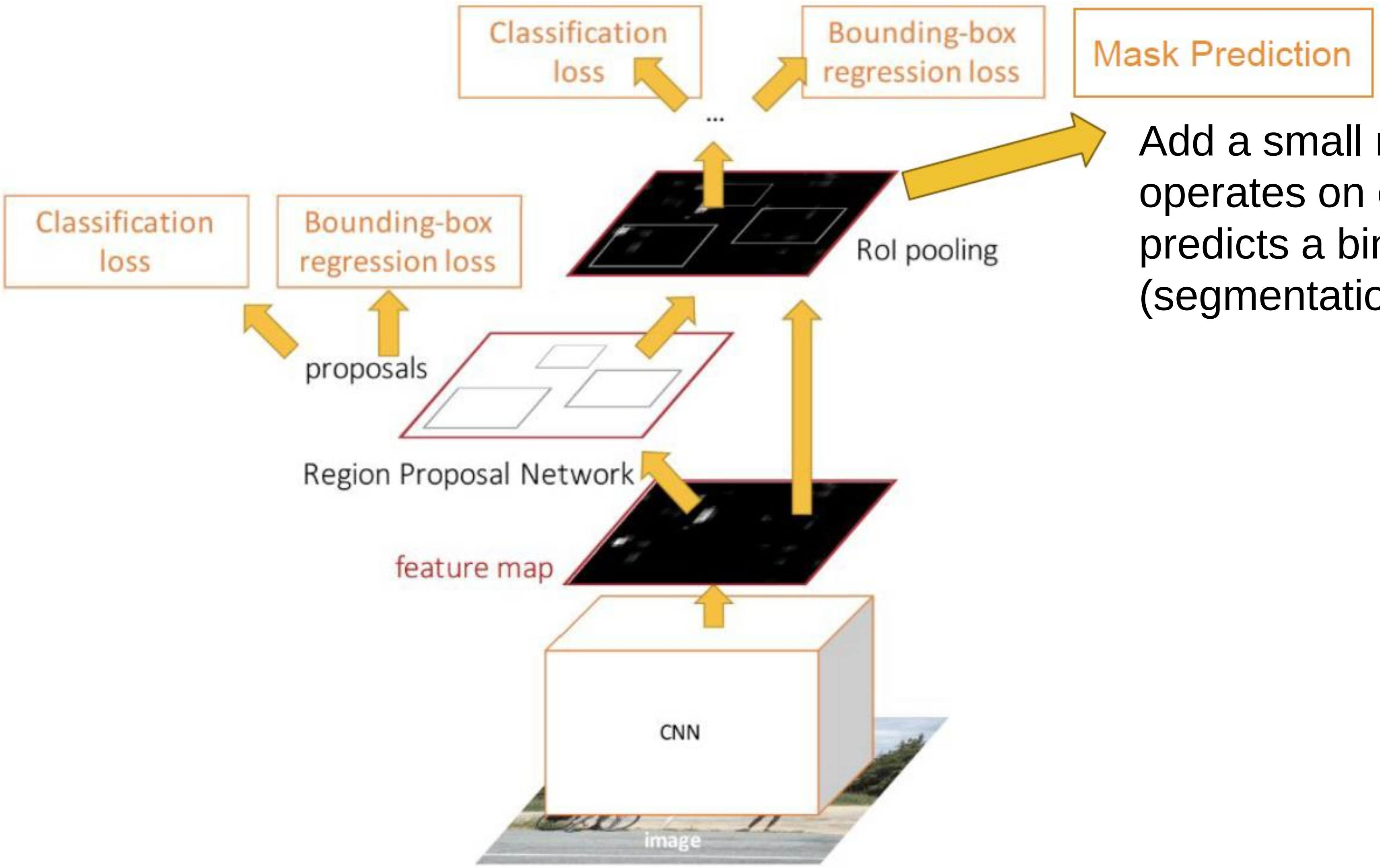
Instance segmentation: Mask R-CNN

Instance Segmentation: **image segmentation distinguishing between different objects/instances of the same class**



He et al, “Mask R-CNN”, ICCV 2017

Instance segmentation: Mask R-CNN

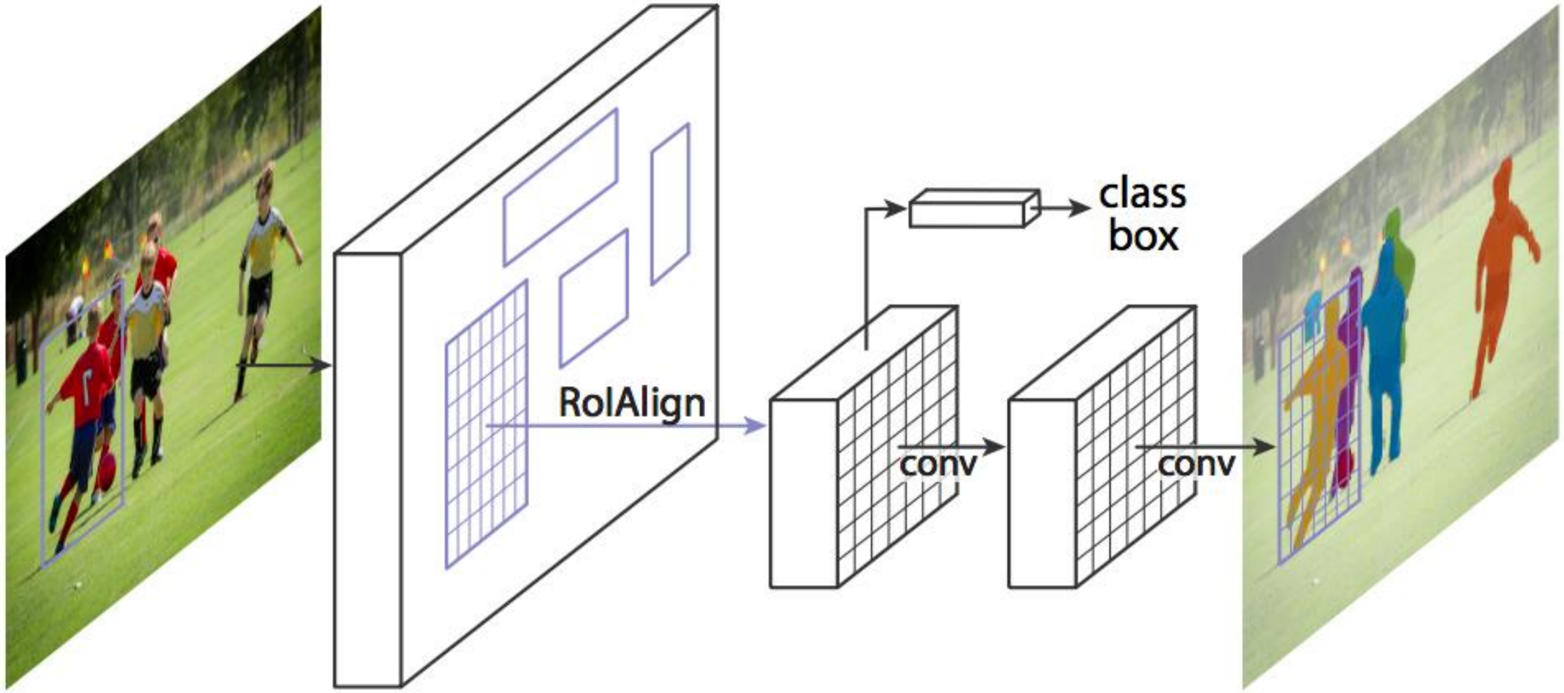


Add a small mask network that operates on each RoI and predicts a binary mask (segmentation)

He et al, “Mask R-CNN”, ICCV 2017

Instance segmentation: Mask R-CNN

Mask R-CNN = Faster R-CNN + FCN on RoIs



Mask branch: separately predict segmentation for each possible class

He et al, “Mask R-CNN”, ICCV 2017

Object Detection

Pablo Mesejo

pmesejo@go.ugr.es

Universidad de Granada

Departamento de Ciencias de la Computación e Inteligencia Artificial

UNIVERSIDAD DE GRANADA

DECSAI

DaSCI

Instituto Andaluz de Investigación en

Data Science and **Computational Intelligence**