Vision and Cognitive Systems

SCQ1097939 - LM CS,DS,CYB,PD

Object Detection, Segmentation

Prof. Lamberto Ballan

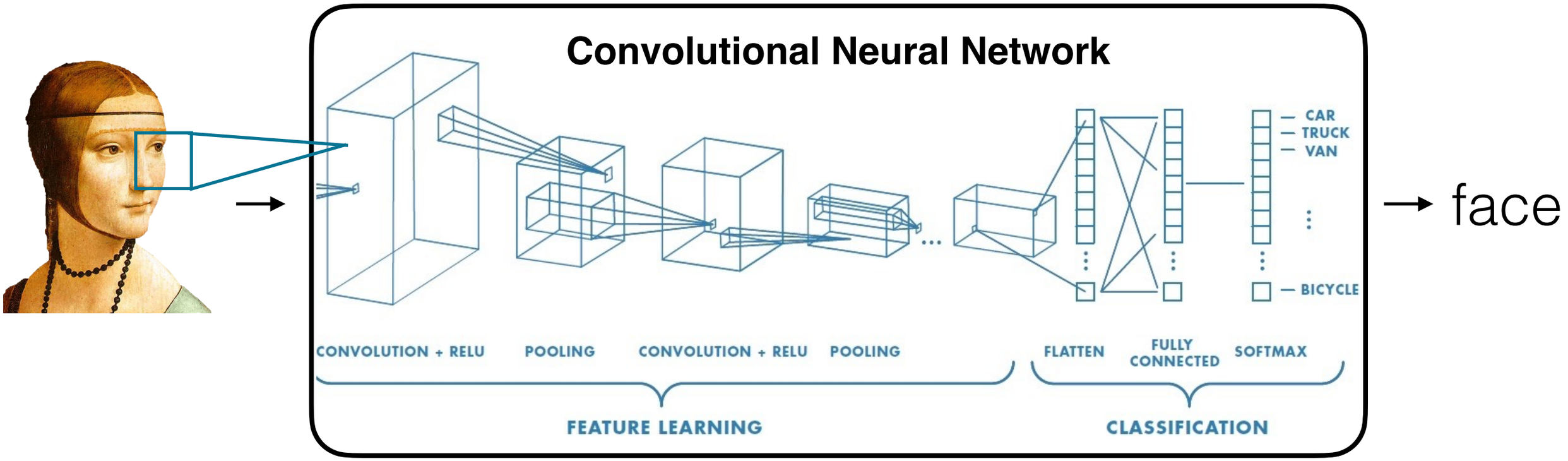
University of Padova, A.Y. 2023/24 December 11, 2023

DIPARTIMENTO MATEMATICA
Department of Mathematics “Tullio Levi-Civita”

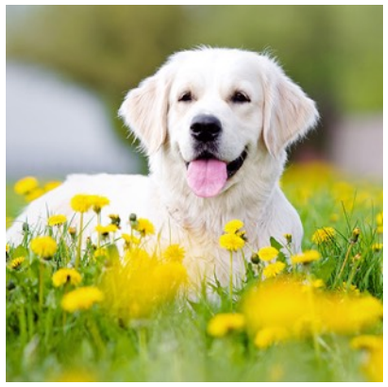
Recap: representation learning

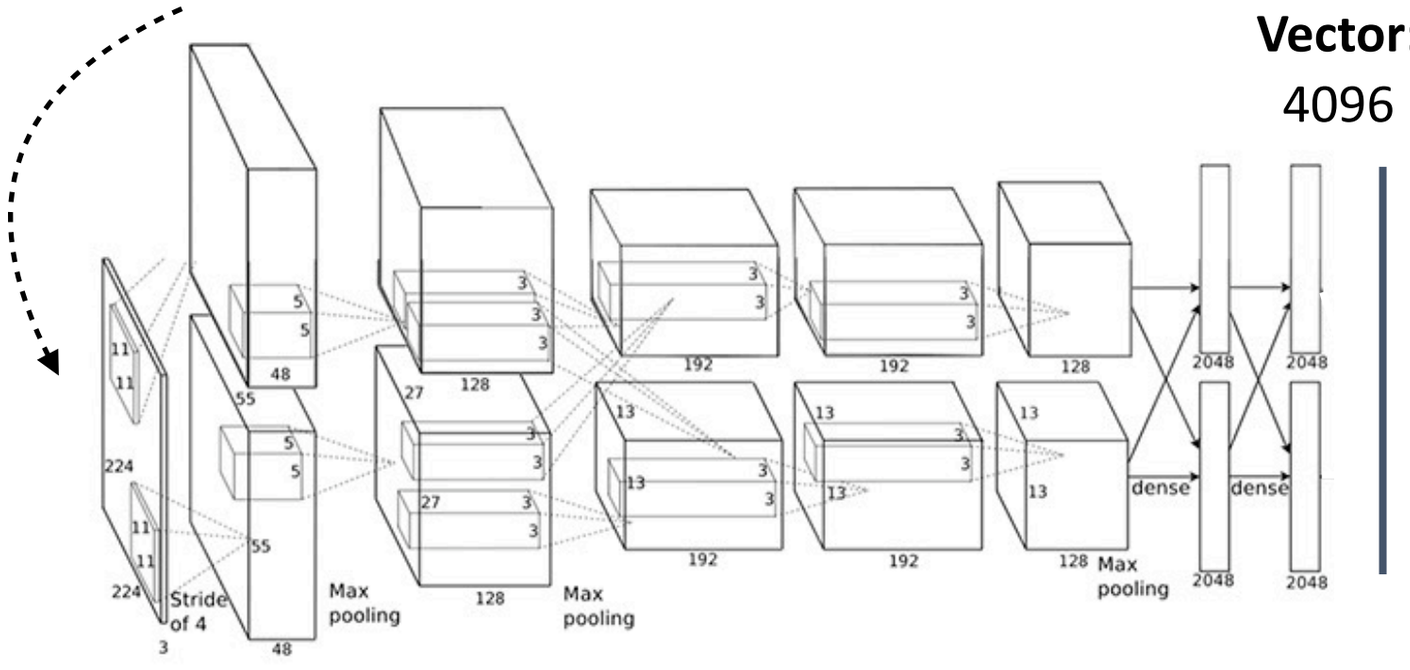
- Intuition: learn also the (image) representation directly from the data (*end-to-end learning)*

Deep Learning can be summarized as learning both the representation and the classifier out of data



So far: image classification



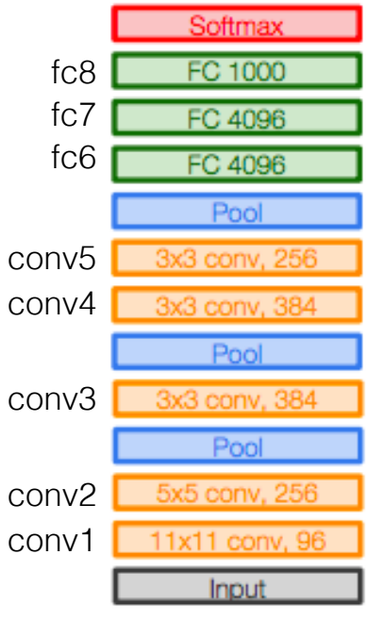


Vector:
4096

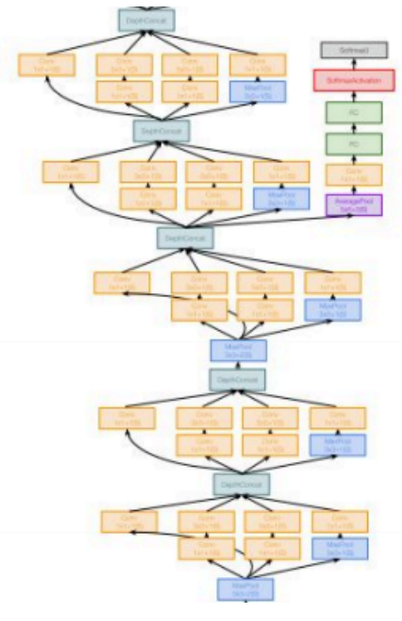
Fully-Connected:
4096 to 1000

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

We covered: CNN architectures, fine-tuning, …



AlexNet



GoogLeNet

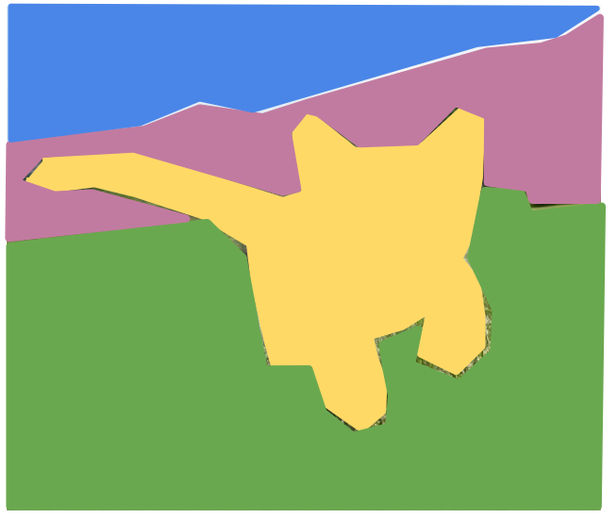
Computer vision tasks

Classification

CAT

No spatial extent

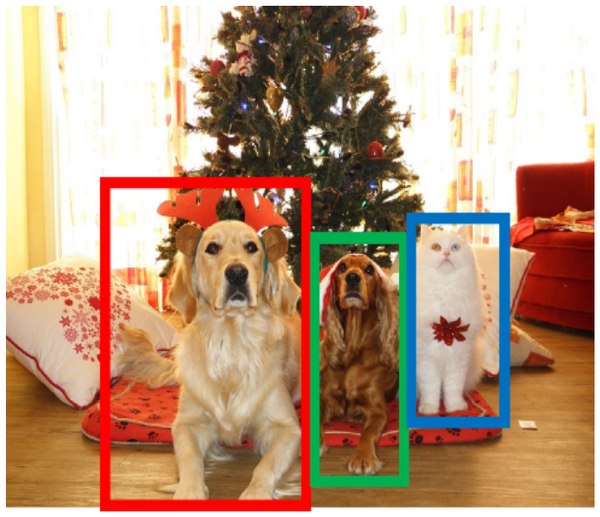
Semantic Segmentation



GRASS, CAT, TREE, SKY

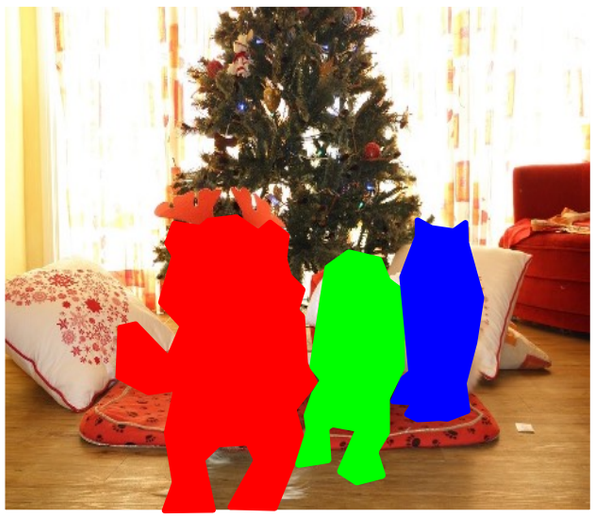
No objects, just pixels

Object Detection



DOG, DOG, CAT

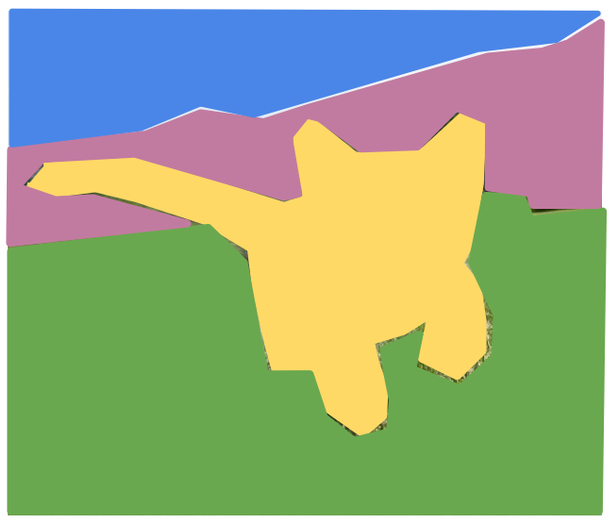
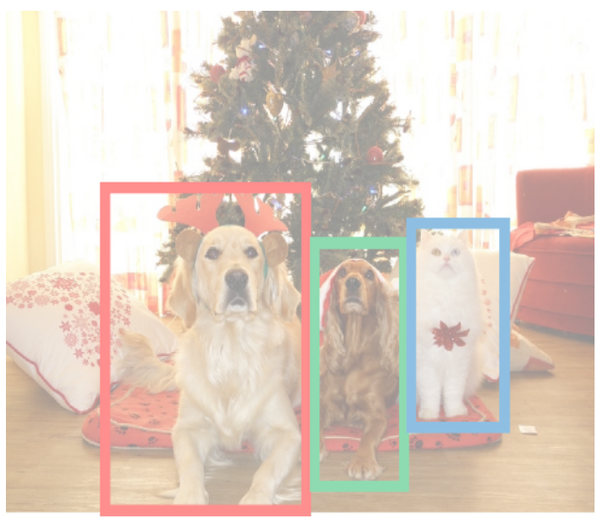
Instance Segmentation



DOG, DOG, CAT

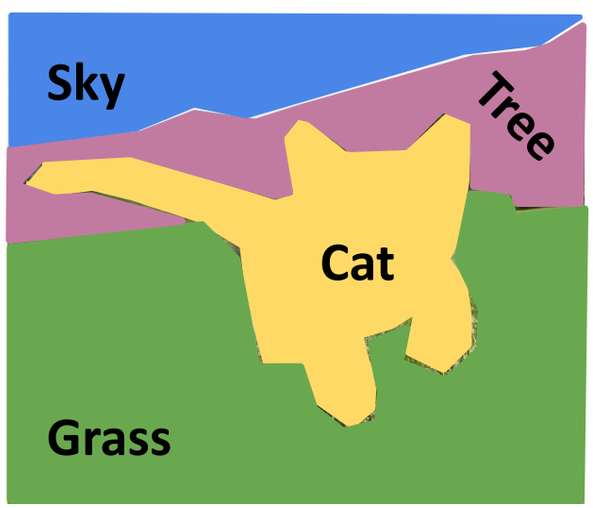
Multiple Objects

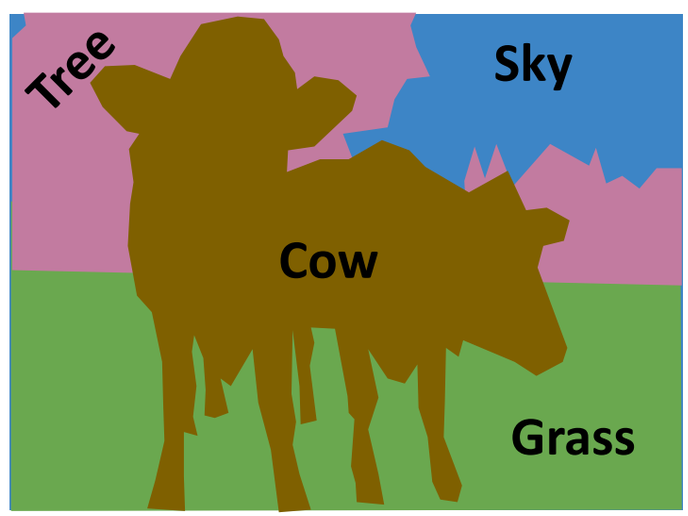
Computer vision tasks

Classification	Semantic Segmentation	Object Detection	Instance Segmentation
			
CAT	GRASS, CAT, TREE, SKY	DOG, DOG, CAT	DOG, DOG, CAT
No spatial extent	No objects, just pixels	Multiple Objects	

Semantic segmentation

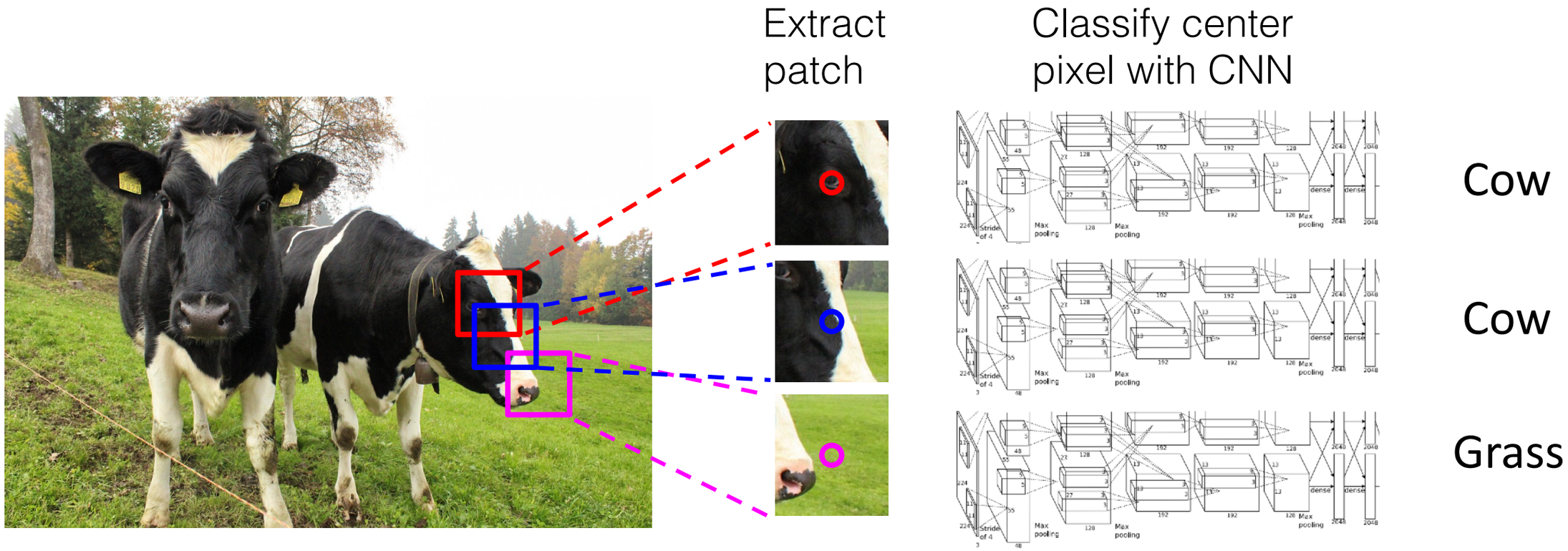
- Goal: label each pixel in the image with a class
  - i.e. don't differentiate instances, only care about pixels





Semantic segmentation

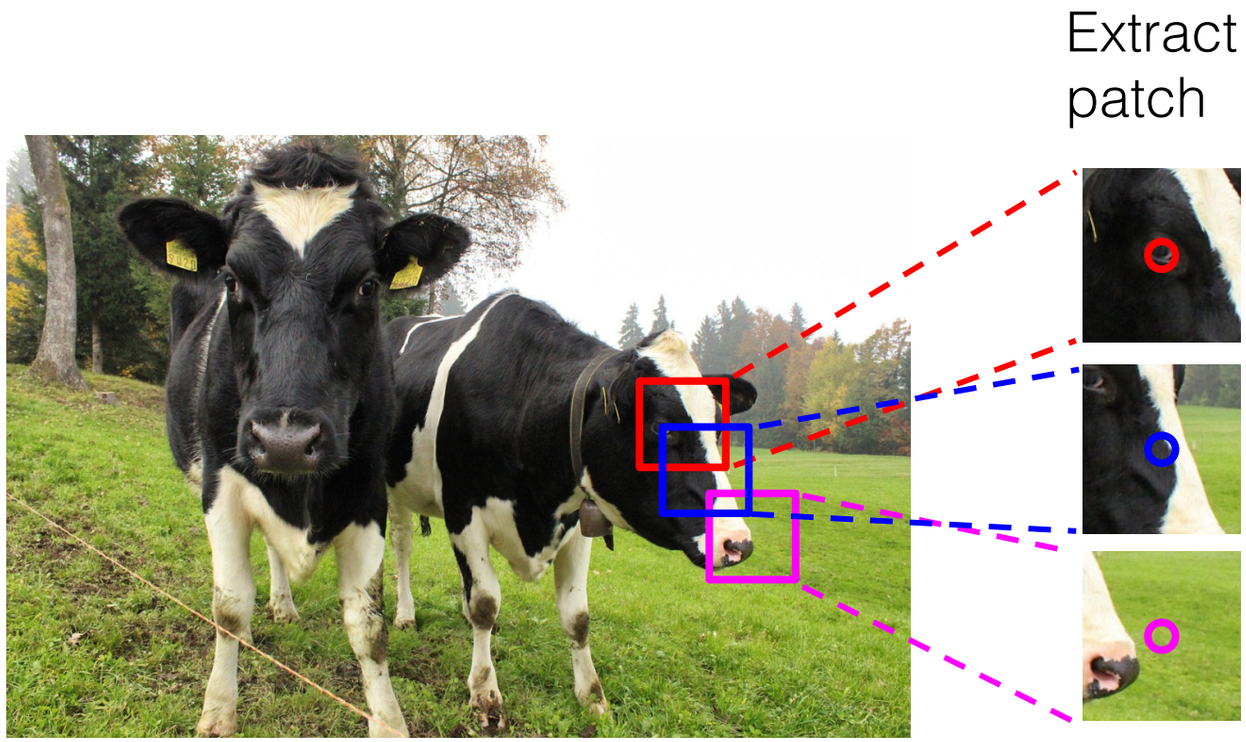
- Idea: use a (CNN based) sliding window approach



Farabet *et al*, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013
Pinheiro and Collobert, “Recurrent CNNs for Scene Labeling”, ICML 2014

Semantic segmentation

- Idea: use a (CNN based) sliding window approach

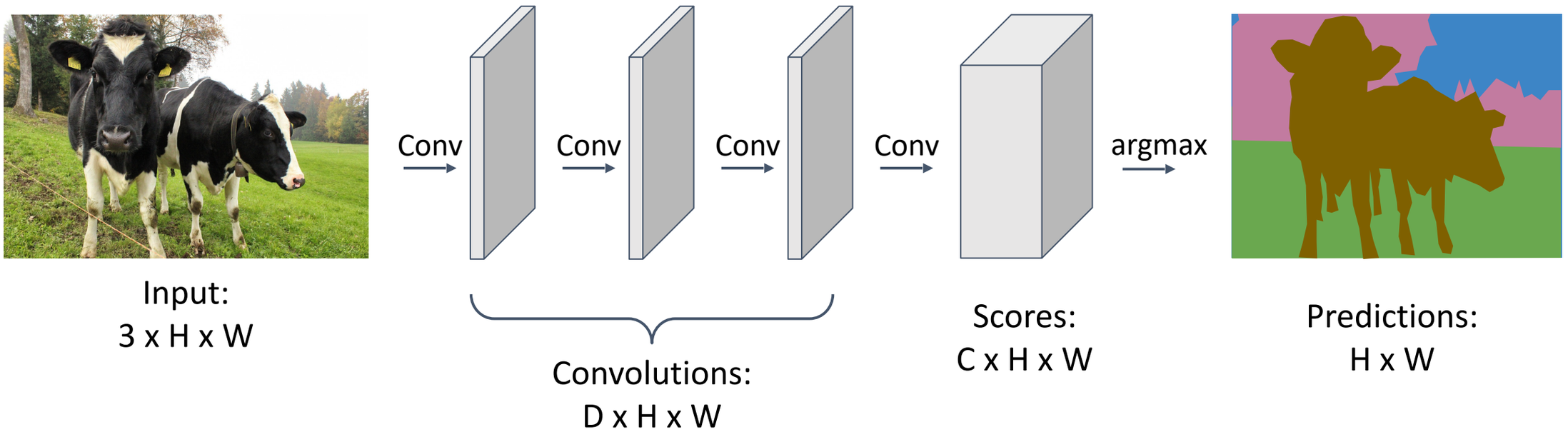


Problem: this procedure is very inefficient!
Not reusing shared features between overlapping patches

Slide credit: J. Johnson

Fully Convolutional Network

- Idea: design a network as a bunch of convolutional layers to make predictions for pixels all at once!

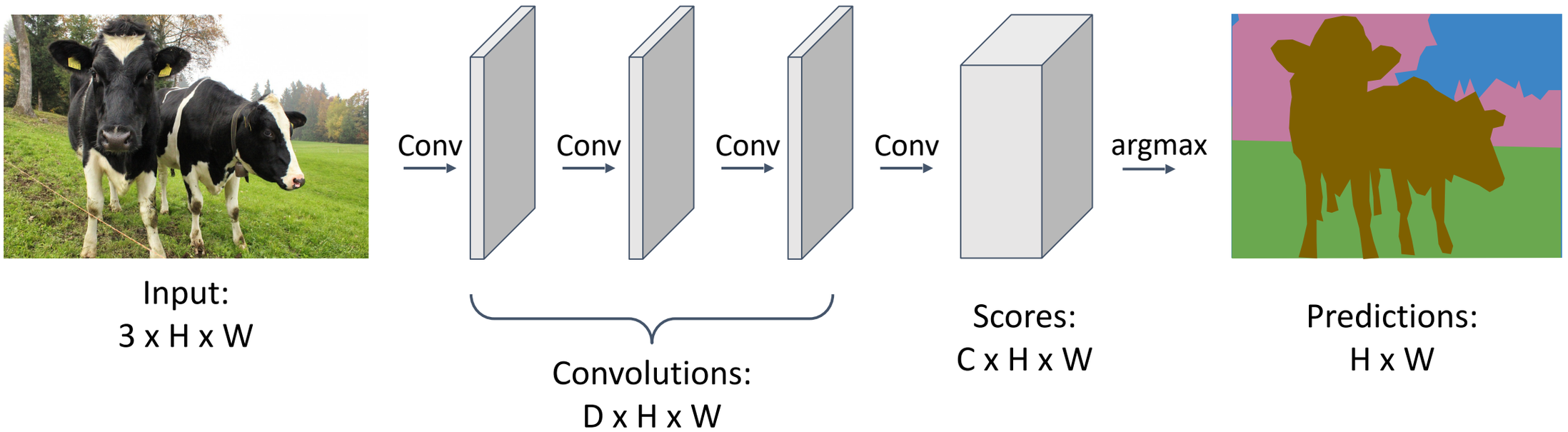


Loss function: Per-Pixel cross-entropy

Long *et al*, “Fully convolutional networks for semantic segmentation”, CVPR 2015

Fully Convolutional Network

- Idea: design a network as a bunch of convolutional layers to make predictions for pixels all at once!

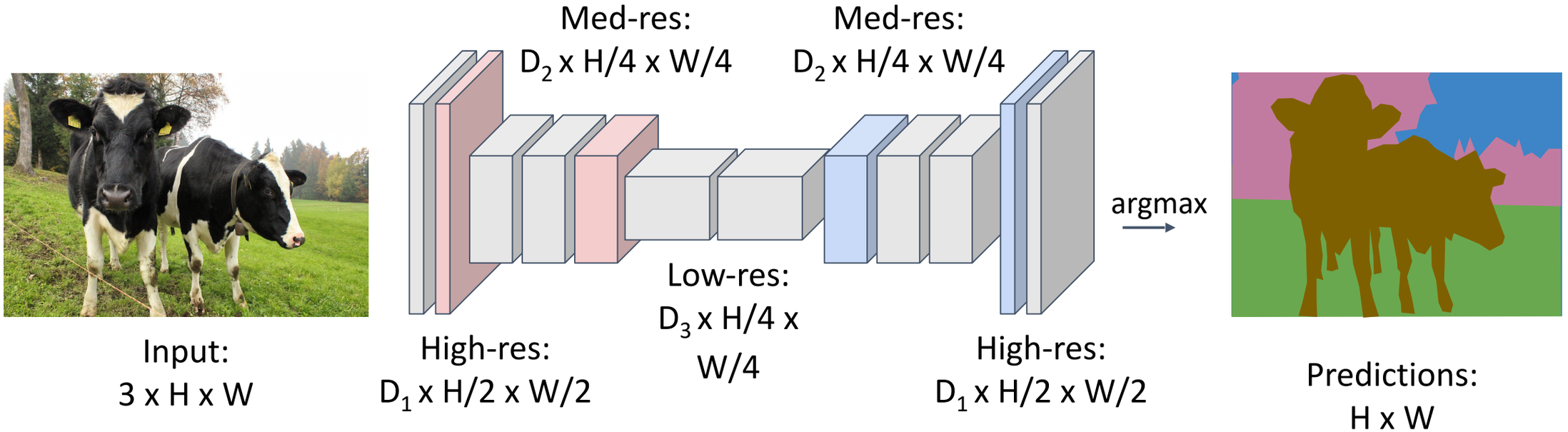


Problem #1: Effective receptive field size is linear in number of conv layers: With L 3x3 conv layers, receptive field is 1+2L

Problem #2: Convolution on high res images is expensive!

Fully Convolutional Network

- Design network as a bunch of conv layers, with downsampling and upsampling inside the network!



Downsampling: Pooling, strided convolution

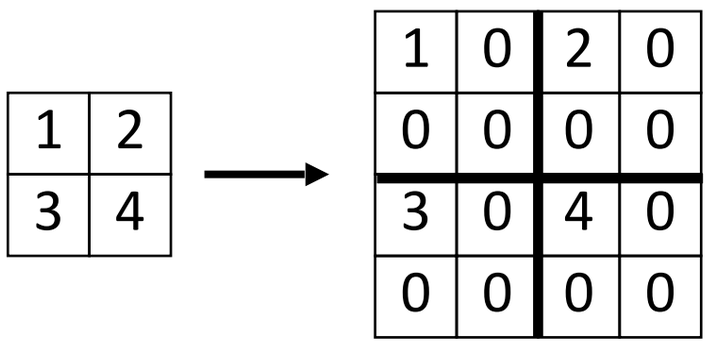
Upsampling: ???

Long *et al*, "Fully convolutional networks for semantic segmentation", CVPR 2015
Noh *et al*, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

In-Network Upsampling

a.k.a. "Unpooling"

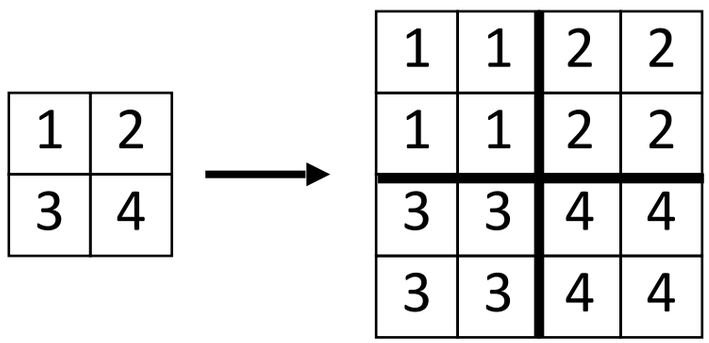
Bed of Nails



Input
C x 2 x 2

Output
C x 4 x 4

Nearest Neighbor



Input
C x 2 x 2

Output
C x 4 x 4

Bilinear interpolation

Input: C x 2 x 2

Output: C x 4 x 4

$$f_{x,y} = \sum_{i,j} f_{i,j} \max(0, 1 - |x - i|) \max(0, 1 - |y - j|)$$

$$i \in \{\lfloor x \rfloor - 1, \ldots, \lceil x \rceil + 1\}$$

$$j \in \{\lfloor y \rfloor - 1, \ldots, \lceil y \rceil + 1\}$$

Use two closest neighbors in x and y to construct linear approximations

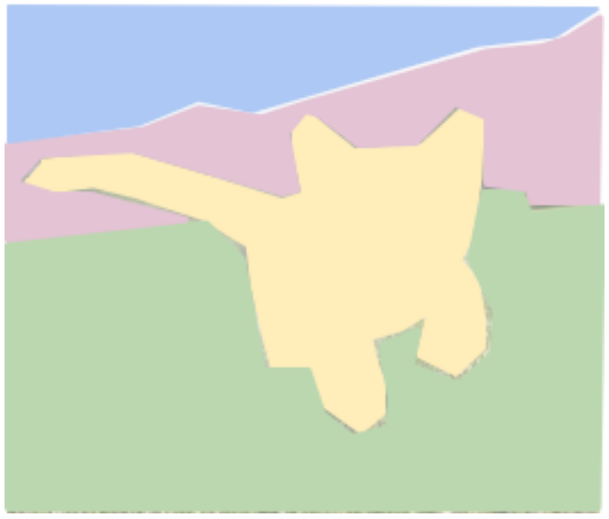
Computer vision tasks

Classification

CAT

No spatial extent

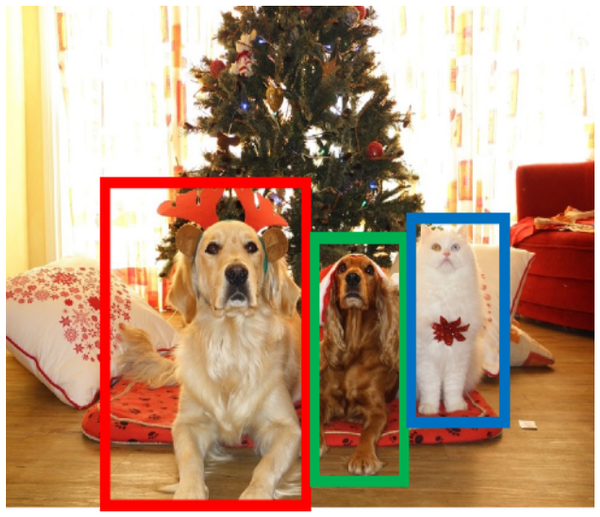
Semantic Segmentation



GRASS, CAT, TREE, SKY

No objects, just pixels

Object Detection



DOG, DOG, CAT

Multiple Objects

Instance Segmentation

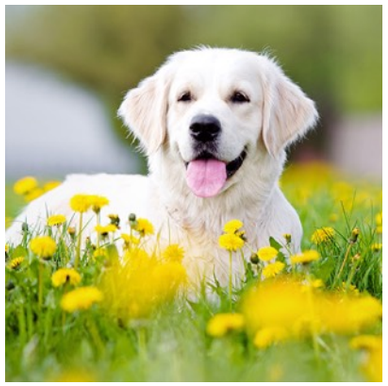
DOG, DOG, CAT

Object detection

- Task: **recognizing** and **localizing** generic objects from various categories (such as cars, people, etc.)

- Challenges:
  - Illumination
  - Viewpoint
  - Deformations
  - Intra-class variability

Detecting a single object



Pretrained (on ImageNet)

Vector:
4096

Fully-Connected:
4096 to 1000

Fully-Connected:
4096 to 4

"What"

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

Softmax Loss

Ground-truth label: Dog

Weighted Sum → **Multitask Loss**

Box Coordinates
(x, y, w, h)

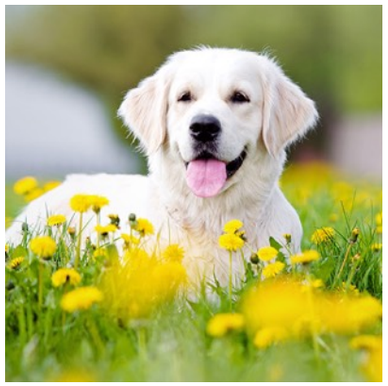
L2 Loss

Ground-truth box:
(x', y', w', h')

"Where"

Idea: treat localization as a regression problem!

Detecting a single object



Pretrained (on ImageNet)

Vector:
4096

Fully-Connected:
4096 to 1000

Fully-Connected:
4096 to 4

"What"

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

Ground-truth label: Dog

Softmax Loss

Weighted Sum

Multitask Loss

Box Coordinates
(x, y, w, h)

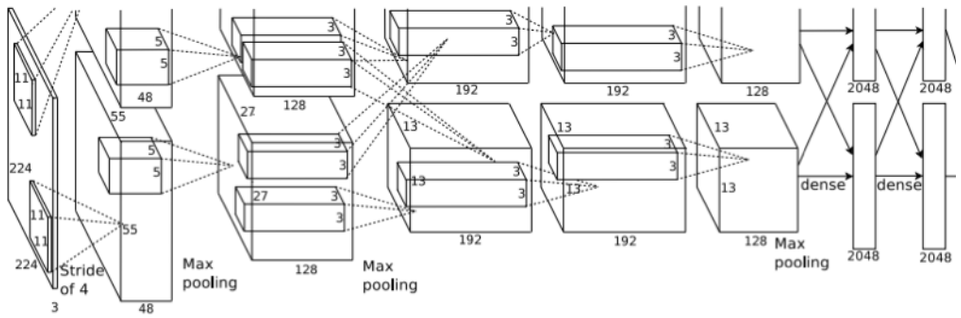
L2 Loss

Ground-truth box:
(x', y', w', h')

"Where"

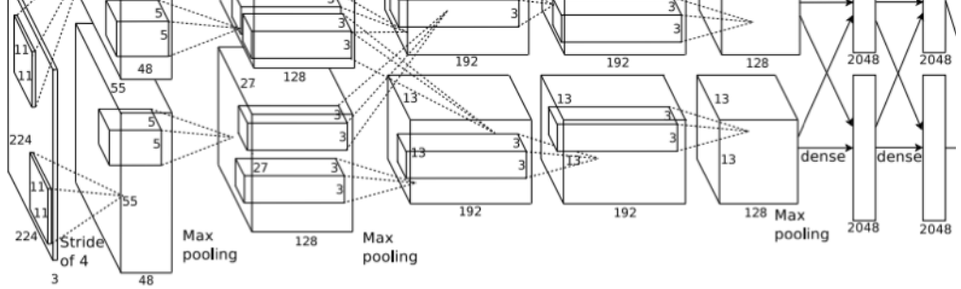
Problem: Images can have more than one object!

Detecting multiple objects



DOG: (x, y, w, h)

4 numbers

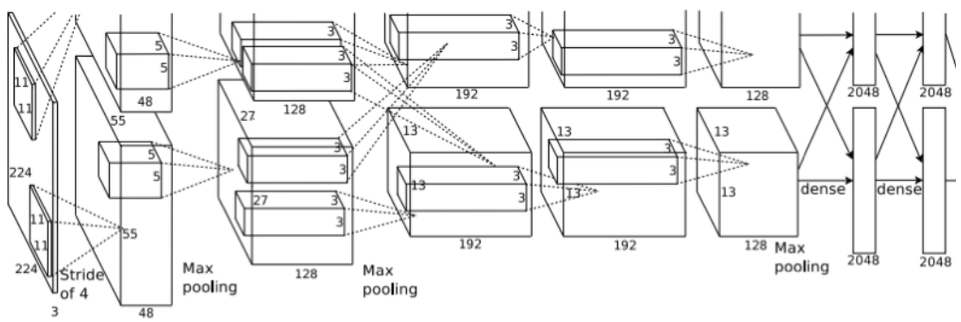


DOG: (x, y, w, h)

DOG: (x, y, w, h)

CAT: (x, y, w, h)

16 numbers



DUCK: (x, y, w, h)

DUCK: (x, y, w, h)

....

Many numbers!

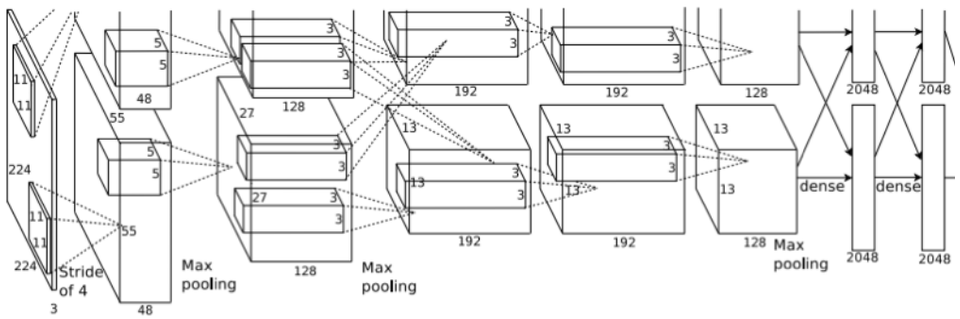
Need different numbers of outputs per image!

Object detection with sliding window

- **Sliding window** approach: slide through the image and check if there is an object at every location
  - Should slide over:
    - locations
    - scales
    - windows size

Object detection with sliding window

- **Sliding window** approach: slide through the image and check if there is an object at every location
  - Apply a CNN to many different crops of the image
  - CNN classifies each crop as object or background



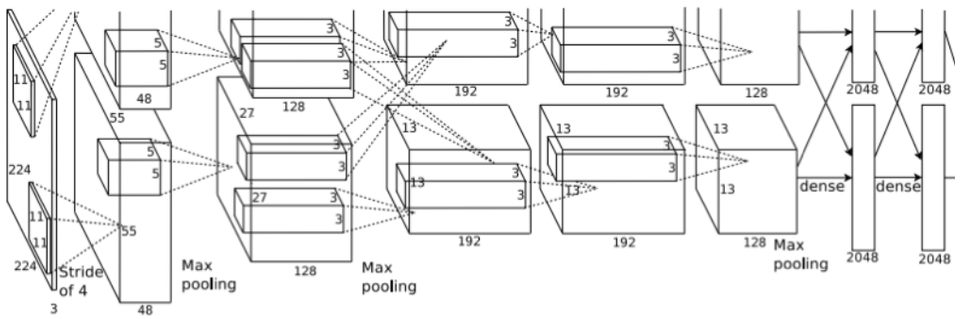
DOG? YES

CAT? NO

Background? NO

Object detection with sliding window

- **Sliding window** approach: slide through the image and check if there is an object at every location
  - Apply a CNN to many different crops of the image
  - CNN classifies each crop as object or background



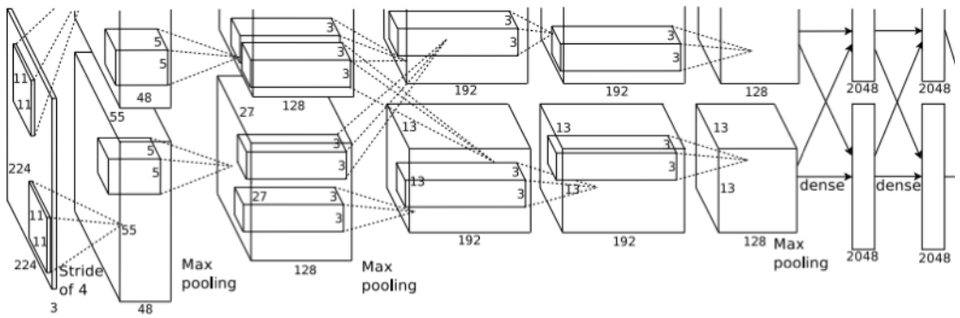
DOG? YES

CAT? NO

Background? NO

Object detection with sliding window

- **Sliding window** approach: slide through the image and check if there is an object at every location
  - Apply a CNN to many different crops of the image
  - CNN classifies each crop as object or background



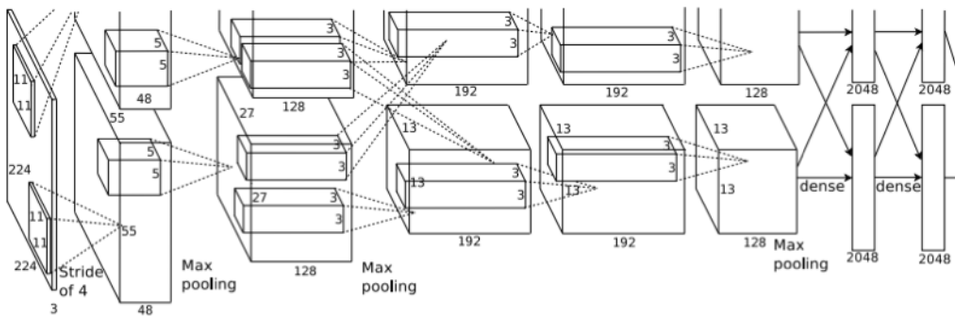
DOG? NO

CAT? YES

Background? NO

Object detection with sliding window

- **Sliding window** approach: slide through the image and check if there is an object at every location
  - Apply a CNN to many different crops of the image
  - CNN classifies each crop as object or background



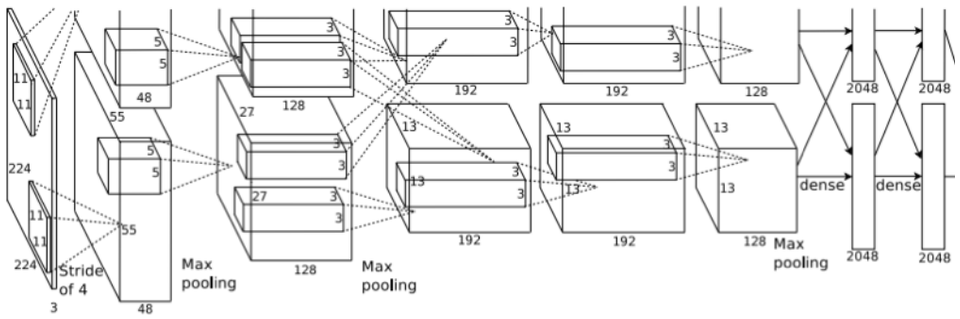
DOG? NO

CAT? YES

Background? NO

Q: How many possible boxes are there in an image of size H x W?

Object detection with sliding window



DOG? NO

CAT? YES

Background? NO

Q: How many possible boxes are there in an image of size H x W?

Consider a box of size h x w:

Possible x positions: W – w + 1

Possible y positions: H – h + 1

Possible positions:
(W – w + 1) * (H – h + 1)

Total possible boxes:

$$\sum_{h=1}^{H}\sum_{w=1}^{W}(W-w+1)(H-h+1)$$

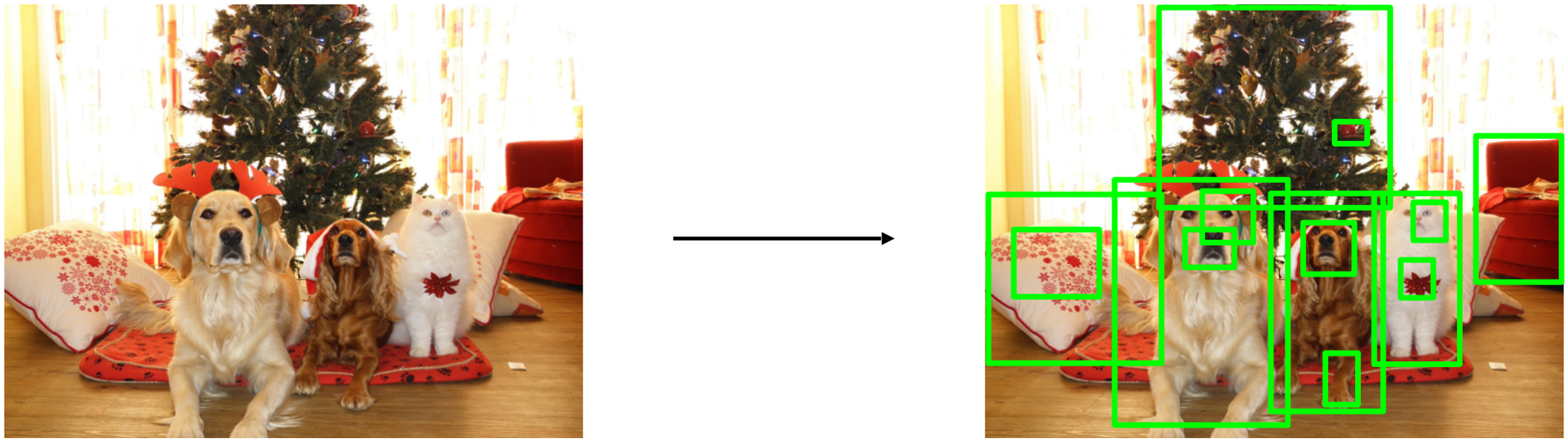
$$=\frac{H(H+1)}{2}\frac{W(W+1)}{2}$$

800 x 600 image has ~58M boxes!

No way we can evaluate them all

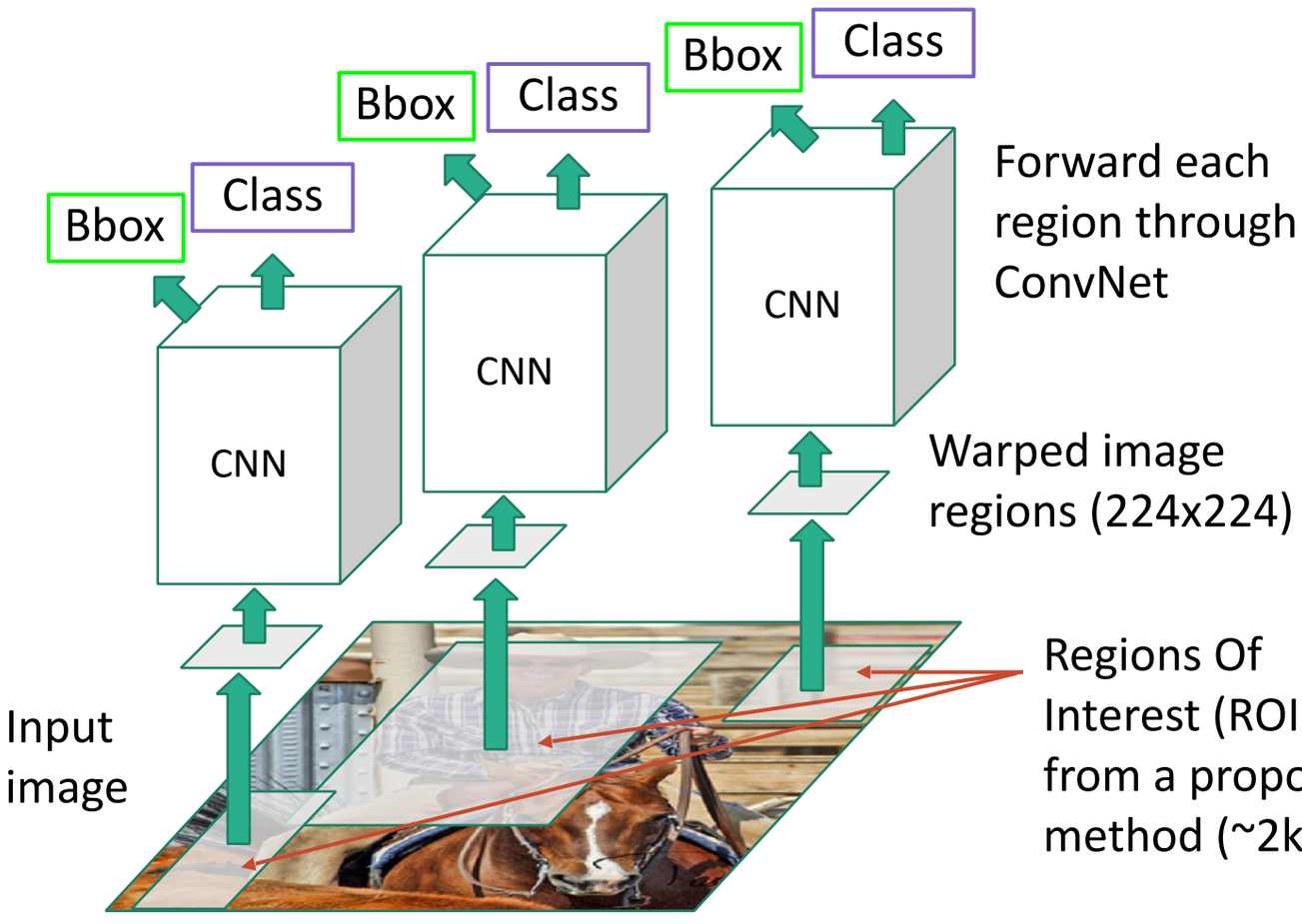
Object detection and region proposals

- Idea: find a small set of boxes (region proposals) that are likely to cover all objects
  - Often based on heuristics: e.g. look for “blob-like” regions
  - Relatively fast to run; e.g. Selective Search gives 2000 region proposals in a few seconds on CPU



Uijlings *et al*, “Selective Search for Object Recognition”, IJCV 2013
Zitnick, Dollar, “Edge boxes: Locating object proposals from edges”, ECCV 2014

R-CNN: Region-Based CNN



Classify each region

Bounding box regression:
Predict "transform" to correct the ROI: 4 numbers (t_x, t_y, t_h, t_w)

Region proposal: (p_x, p_y, p_h, p_w)
Transform: (t_x, t_y, t_h, t_w)
Output box: (b_x, b_y, b_h, b_w)

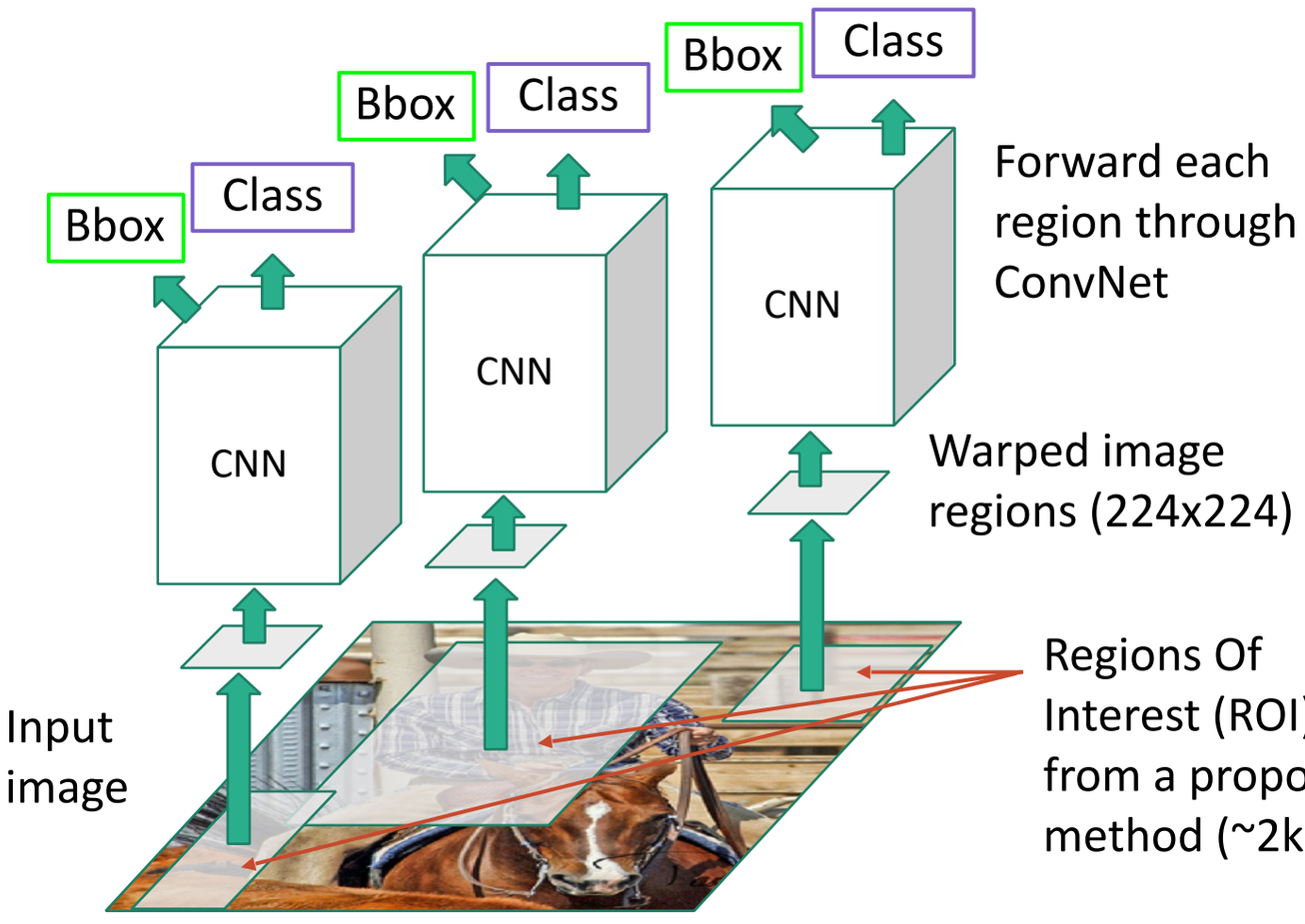
Translate relative to box size:
$b_x = p_x + p_w t_x$ $\quad$ $b_y = p_y + p_h t_y$

Log-space scale transform:
$b_w = p_w \exp(t_w)$ $\quad$ $b_h = p_h \exp(t_h)$

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

R-CNN: Region-Based CNN

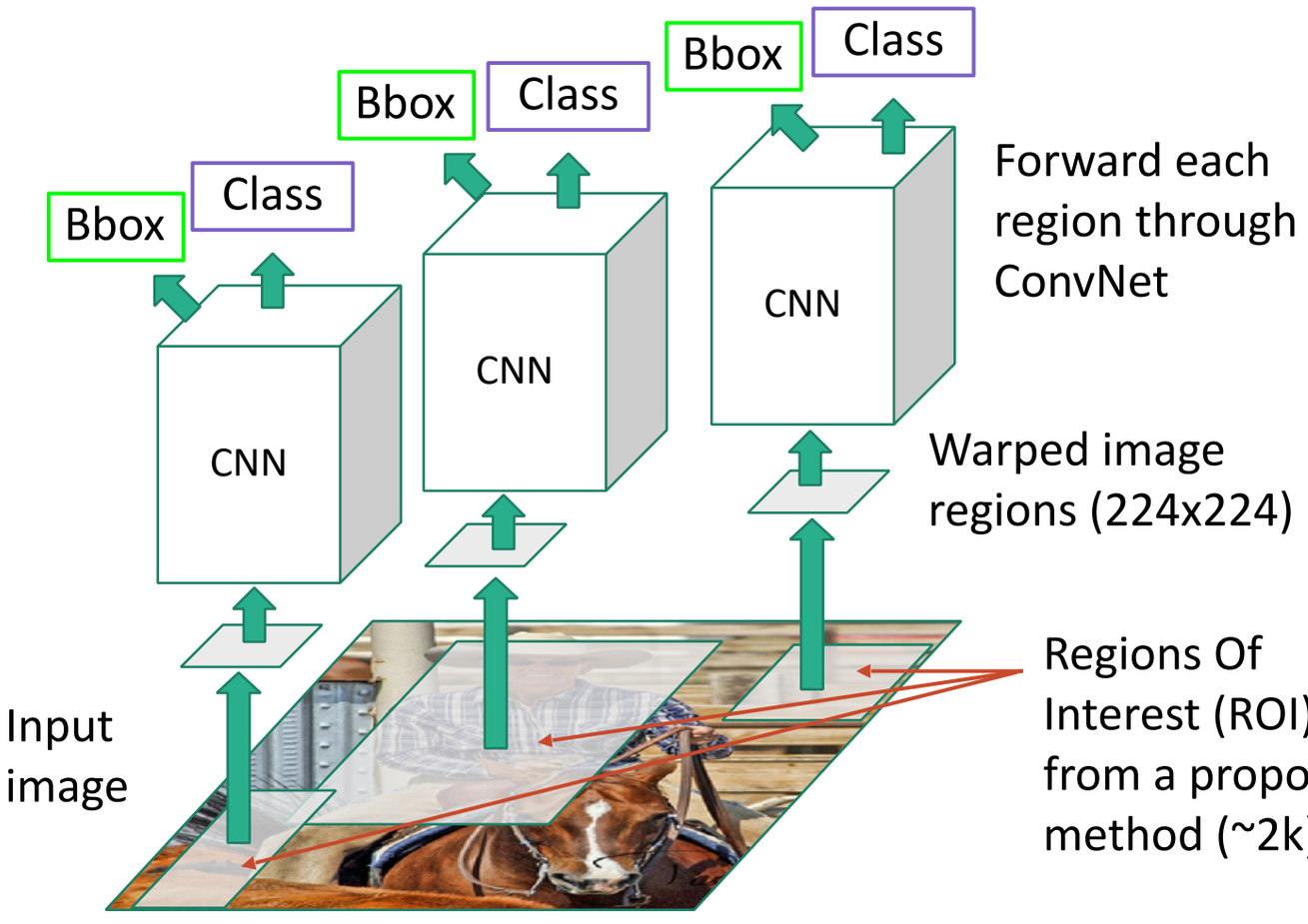
- R-CNN at test-time:



- Run region proposal method to compute ~2000 proposals
- Resize each region to 224x224 and run independently through CNN to predict class scores and bbox transform
- Use scores to select a subset of proposals to output
  (Many choices here: threshold on background, or per-category? Or take top K proposals per image?)
- Compare with groundtruth boxes

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

R-CNN: Region-Based CNN

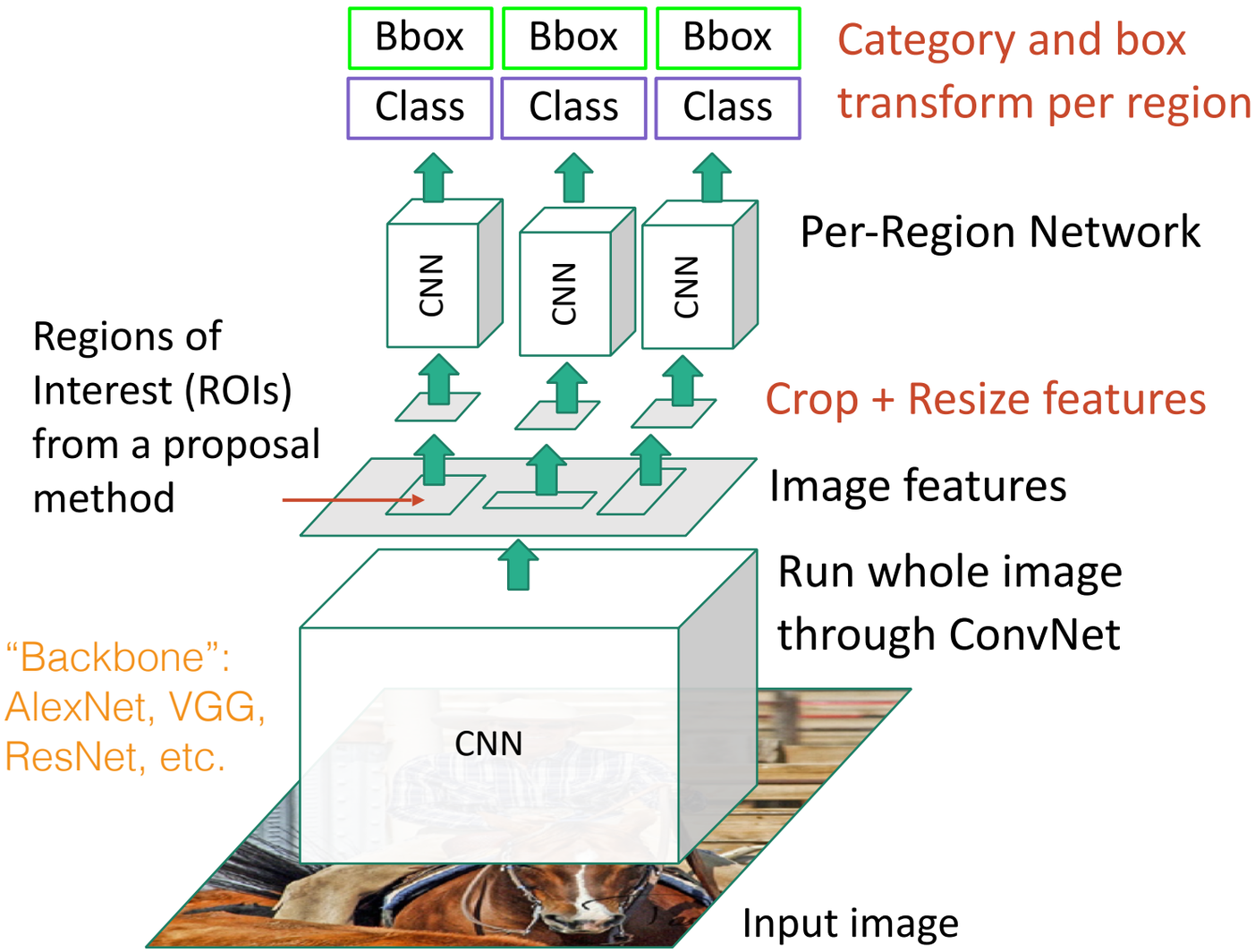


Problem: very slow! Need to do ~2K forward passes for each image!

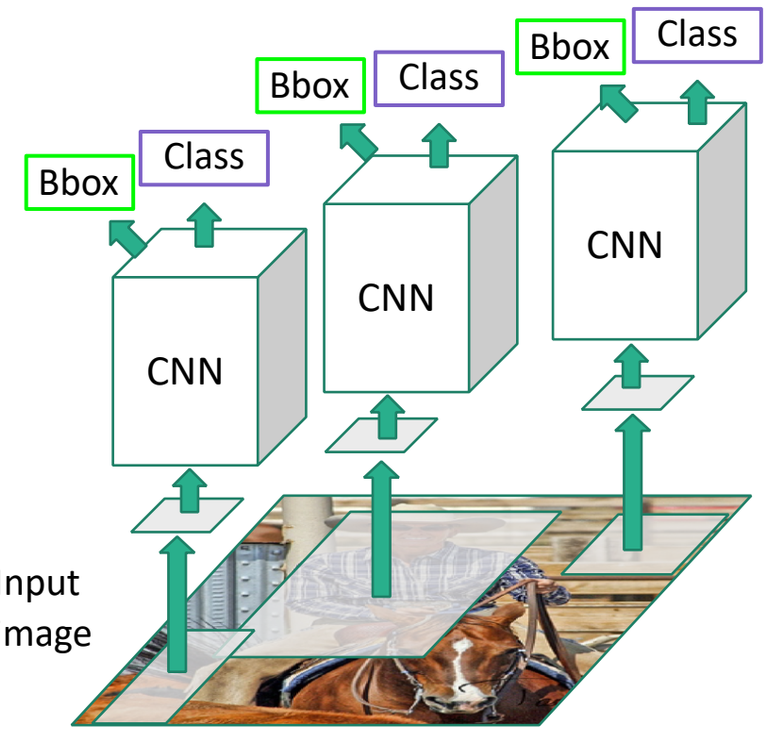
Solution: run CNN before cropping!

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

Fast R-CNN

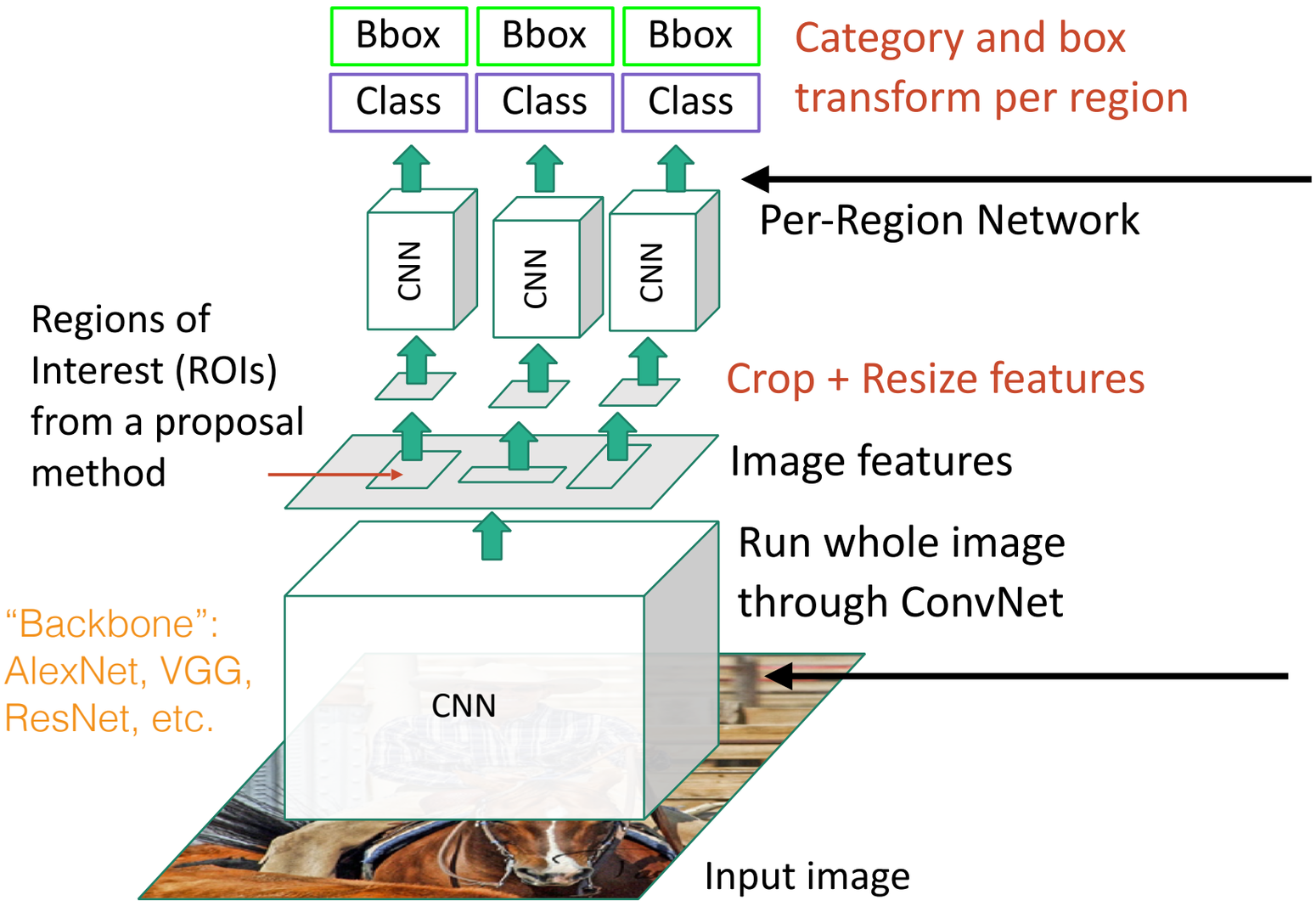


"Slow" R-CNN
Process each region independently



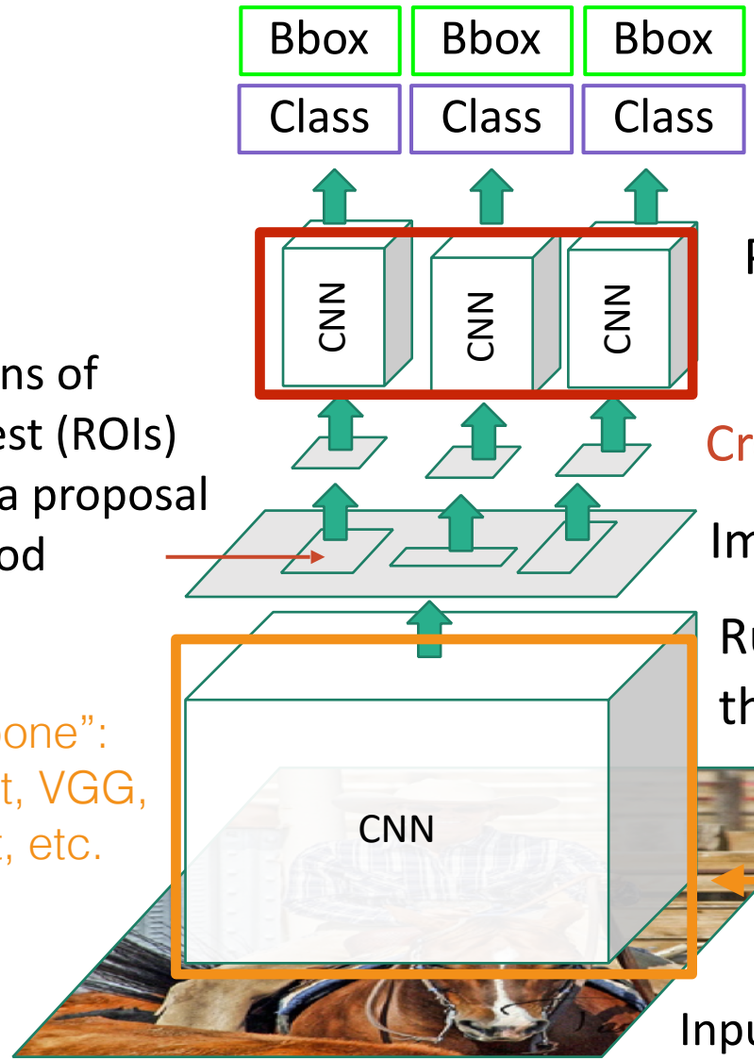
Girshick, "Fast R-CNN", ICCV 2015

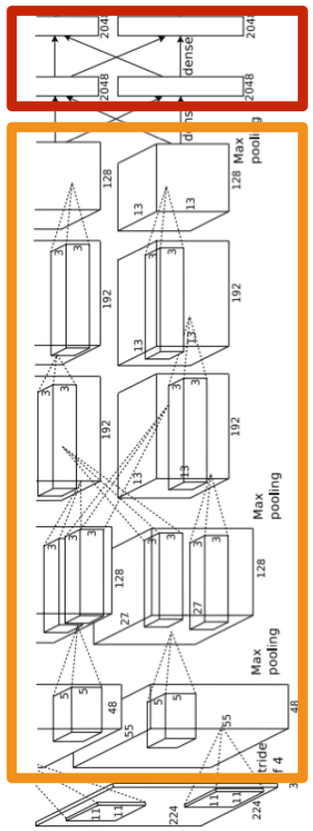
Fast R-CNN



Girshick, "Fast R-CNN", ICCV 2015

Fast R-CNN



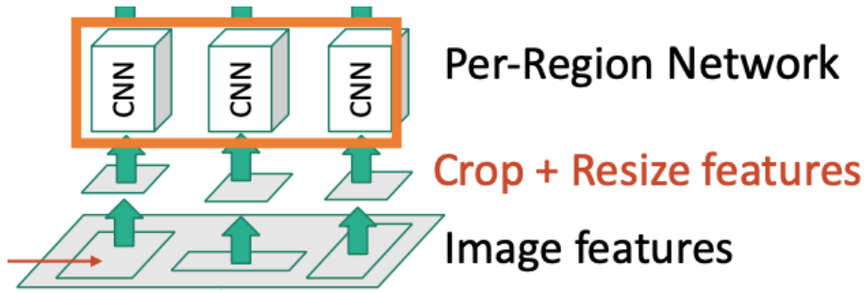


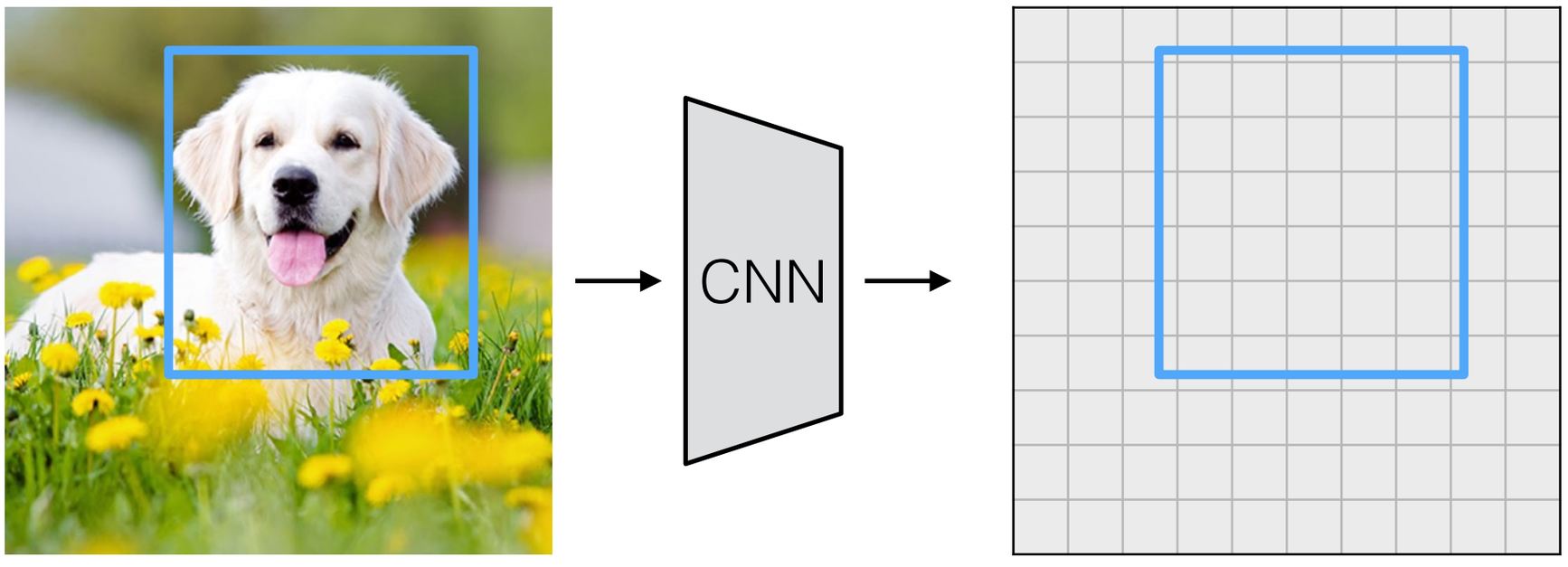
An example:
When using AlexNet for detection, five conv layers are used for backbone and two FC layers are used for per-region network

Girshick, "Fast R-CNN", ICCV 2015

Cropping features

- How to crop features?





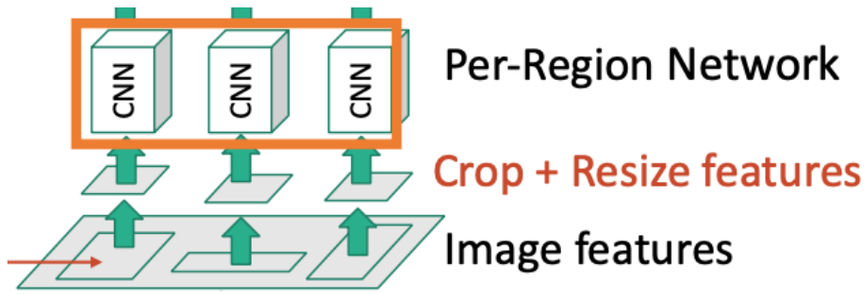
Input image
(e.g. 3x240x240)

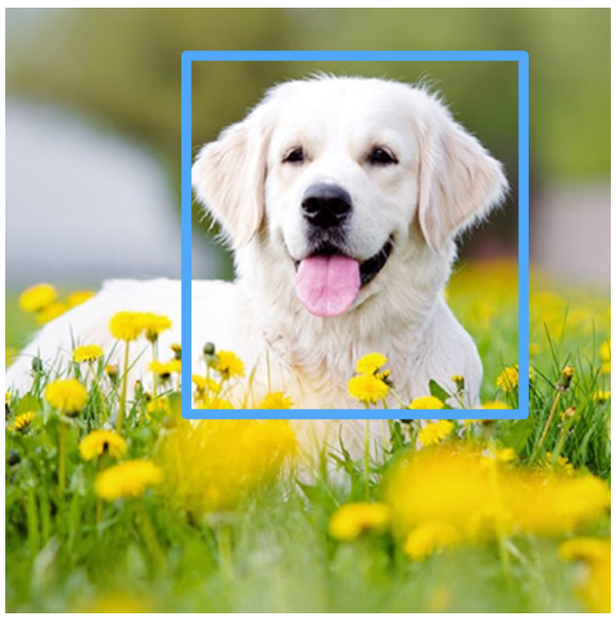
Image features
(e.g. 512x20x15)

1. Project proposal onto features

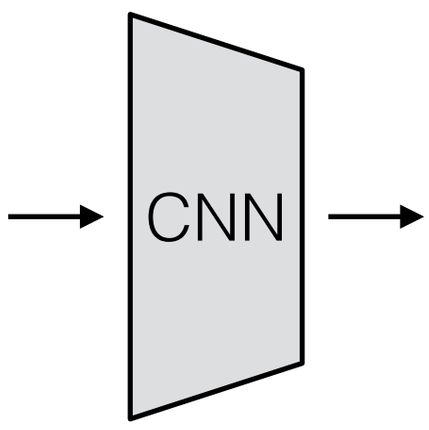
Cropping features

- How to crop features?





Input image
(e.g. 3x240x240)



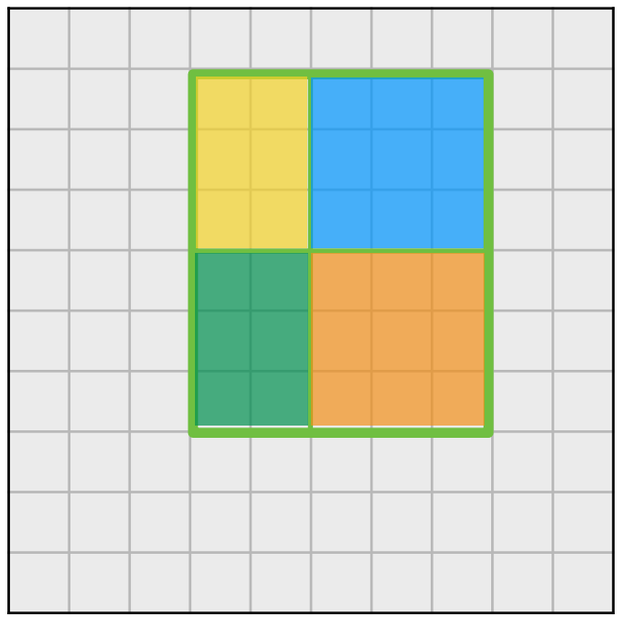


Image features
(e.g. 512x20x15)

2. "Snap" to grid cells

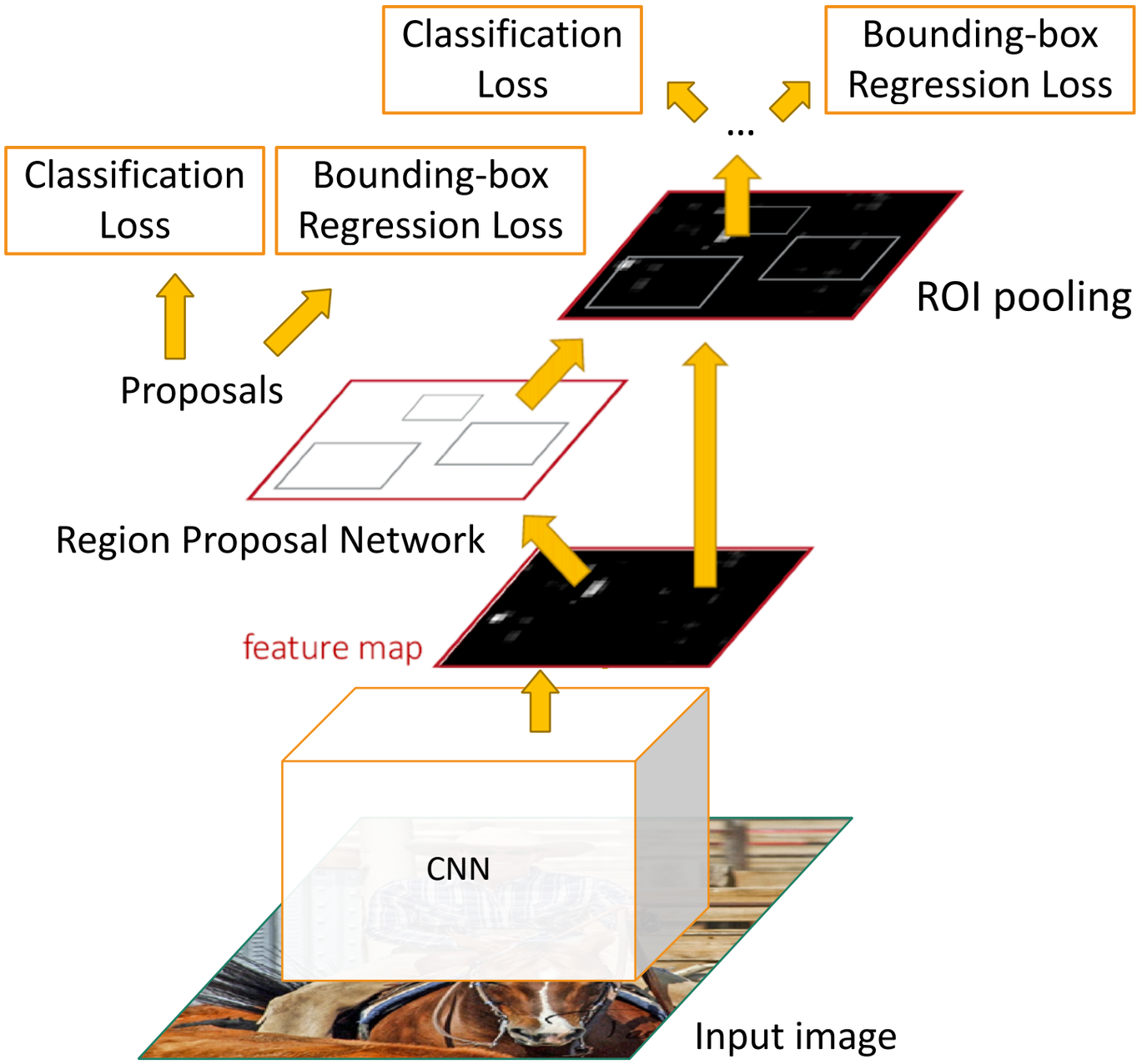
3. Divide into grid of ~equal subregions (e.g. 2x2)

Max pooling

Region features
(here: 512x2x2)
(in practice: e.g. 512x7x7)

1. Project proposal onto features

Faster R-CNN: learnable proposals

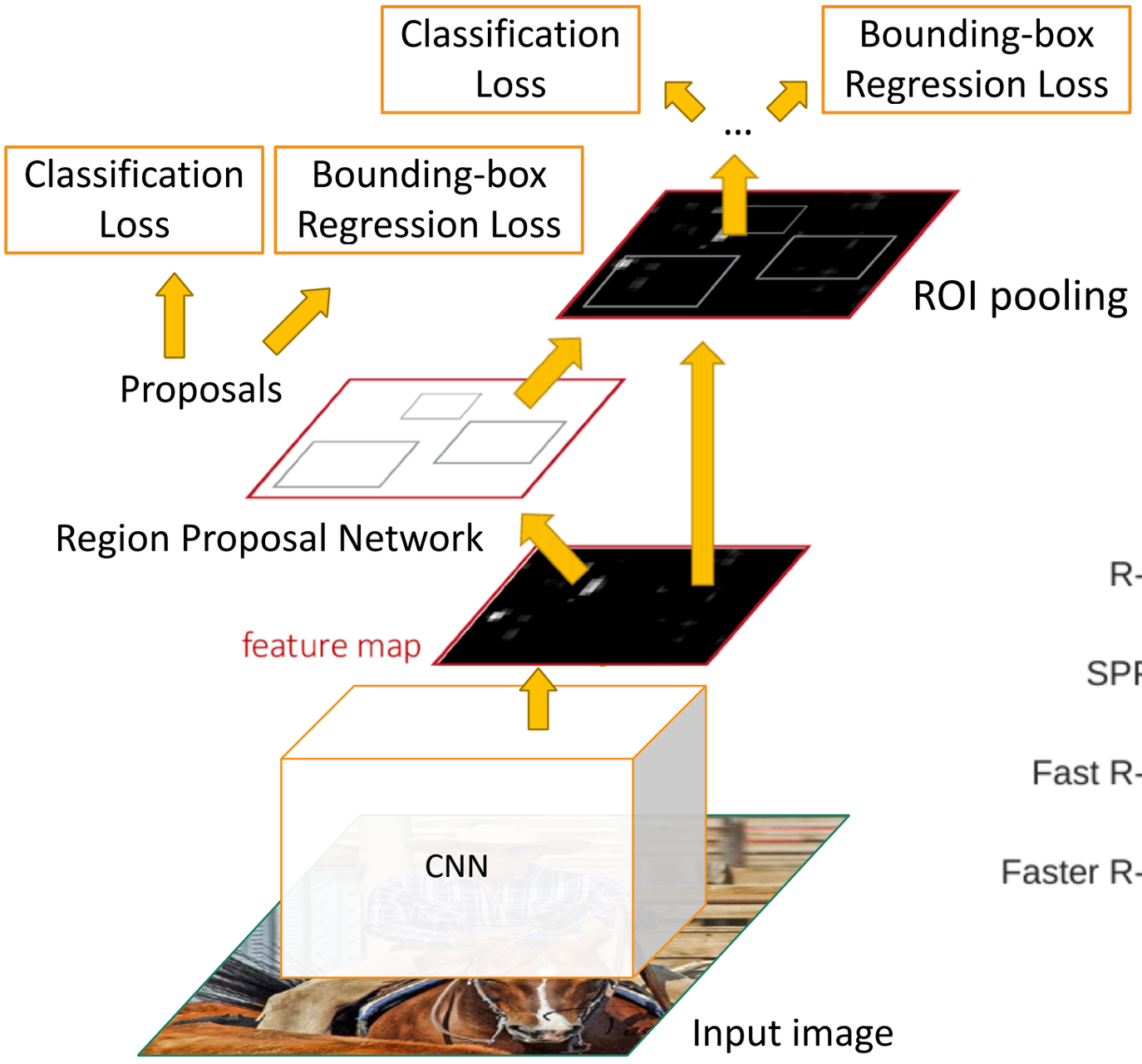


Faster R-CNN is a
Two-stage object detector

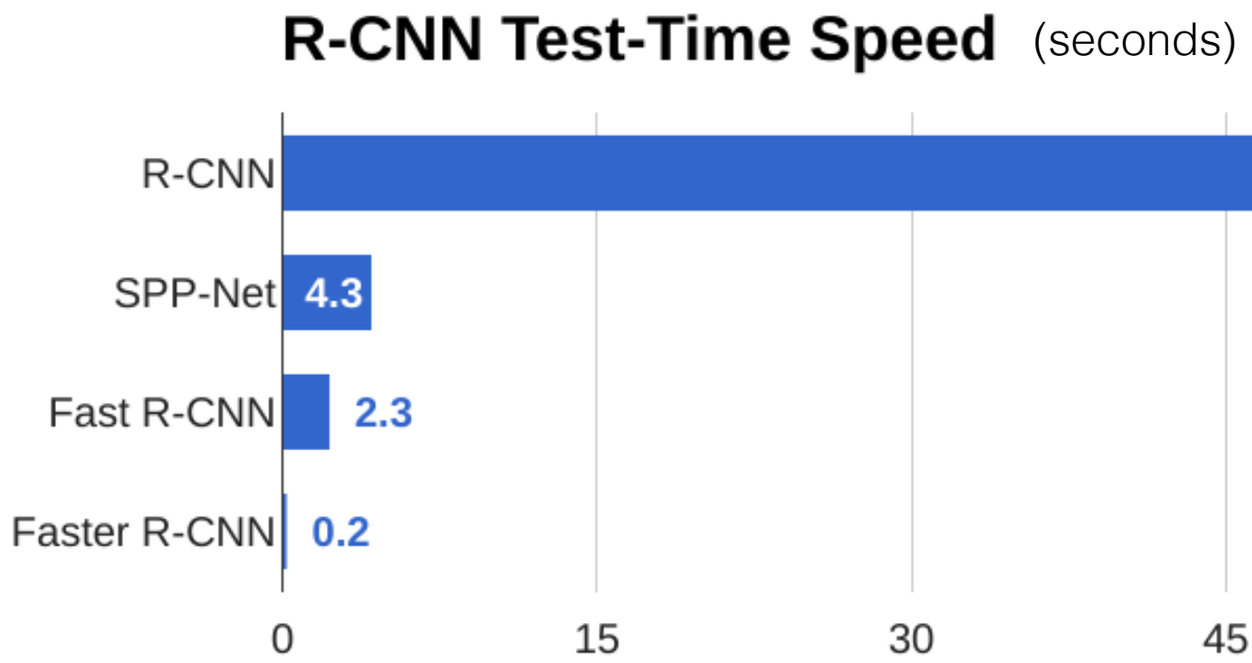
- Insert Region Proposal Network (RPN) to predict proposals from features
- Otherwise same as Fast R-CNN: Crop features for each proposal, classify each one

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

Faster R-CNN: learnable proposals



How fast is “faster”?



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

Object detection: evaluation

- How do we evaluate object detection?

predictions

ground truth

Overlap between predicted and groundtruth boxes > *Threshold* (e.g. 0.5)

Object detection: evaluation

- How do we evaluate object detection?

How can we compare the overlap between boxes?

Intersection over Union (IoU)

(Also called “Jaccard similarity” or “Jaccard index”):

$$\frac{Area\ of\ Intersection}{Area\ of\ Union}$$

Object detection: evaluation

- How do we evaluate object detection?

How can we compare the overlap between boxes?

Intersection over Union (IoU)

(Also called “Jaccard similarity” or “Jaccard index”):

$$\frac{Area\ of\ Intersection}{Area\ of\ Union}$$

IoU > 0.5 is “decent”,

IoU > 0.7 is “pretty good”,

Object detection: evaluation

- How do we evaluate object detection?

How can we compare the overlap between boxes?

Intersection over Union (IoU)

(Also called “Jaccard similarity” or “Jaccard index”):

$$\frac{Area\ of\ Intersection}{Area\ of\ Union}$$

IoU > 0.5 is “decent”,

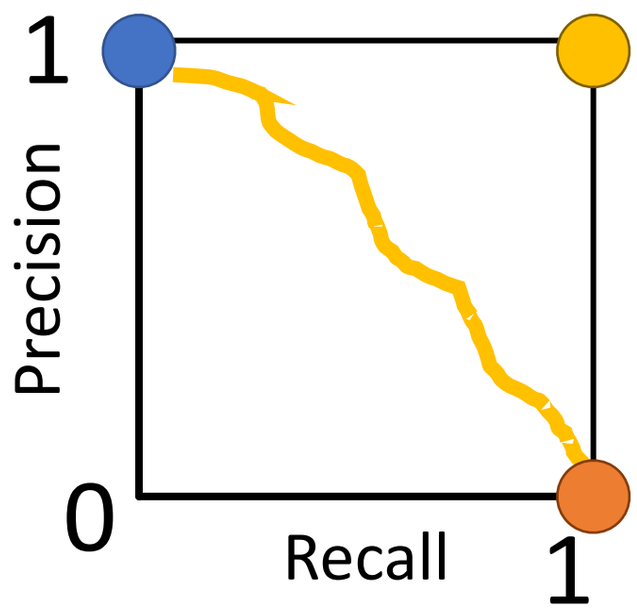
IoU > 0.7 is “pretty good”,

IoU > 0.9 is “almost perfect”

Object detection: evaluation

- Evaluating (object) detectors:
  - True detection: high intersection over union
  - Precision: #true detections / #detections
  - Recall: #true detections / #true positives

Reject everything: no mistakes



Ideal!

Summarize by area under curve (avg. precision)

Accept everything: Miss nothing

Object detection: evaluation

- Evaluating (object) detectors:

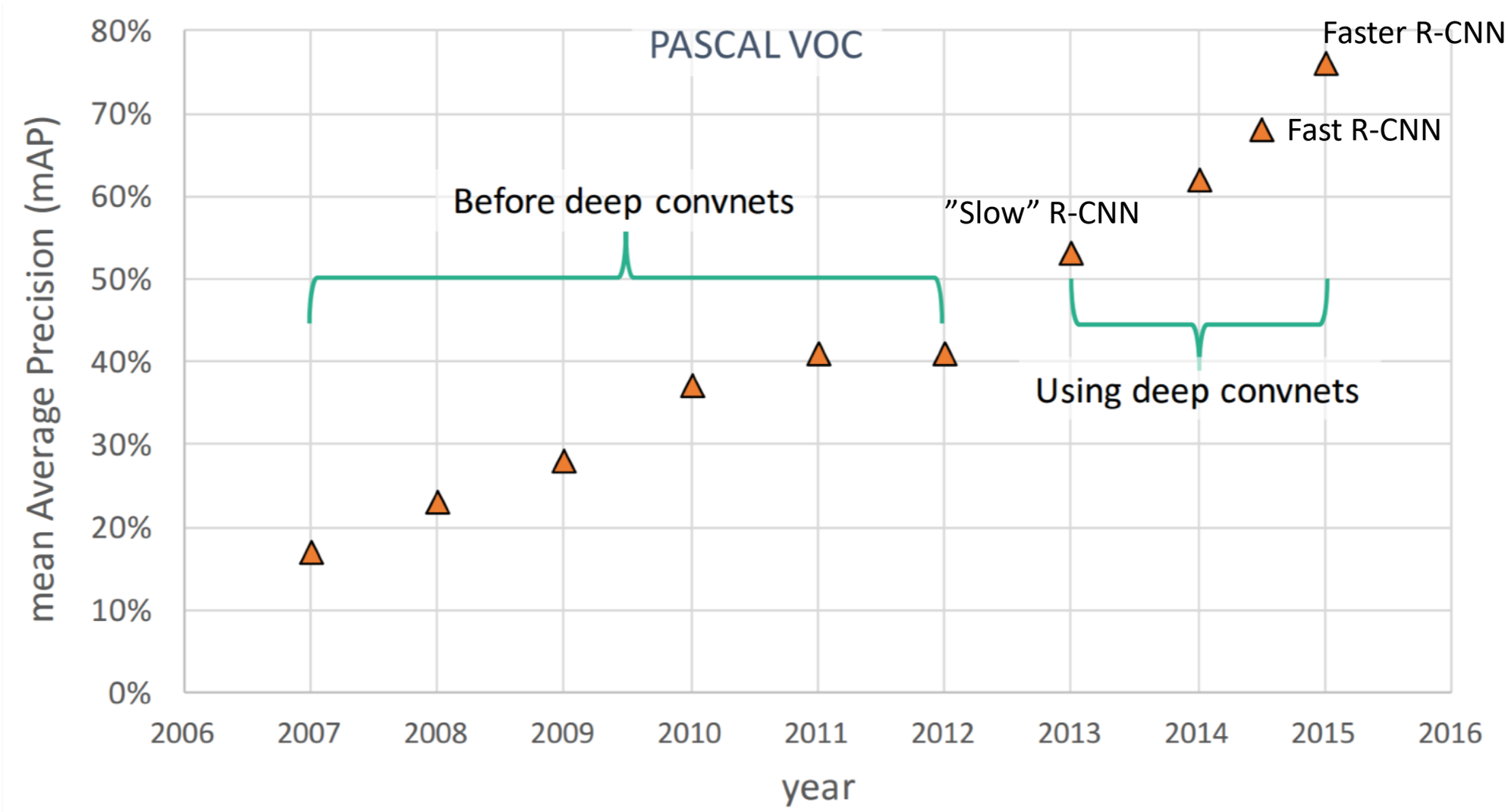
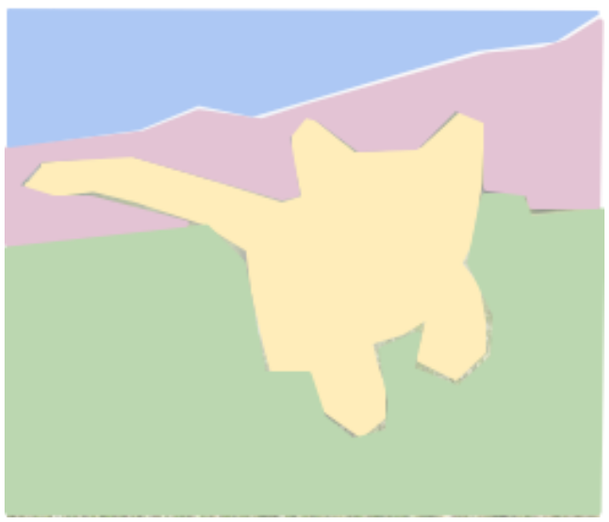
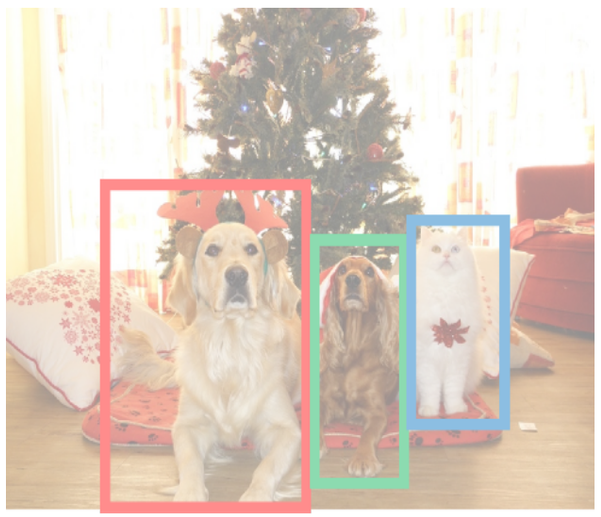
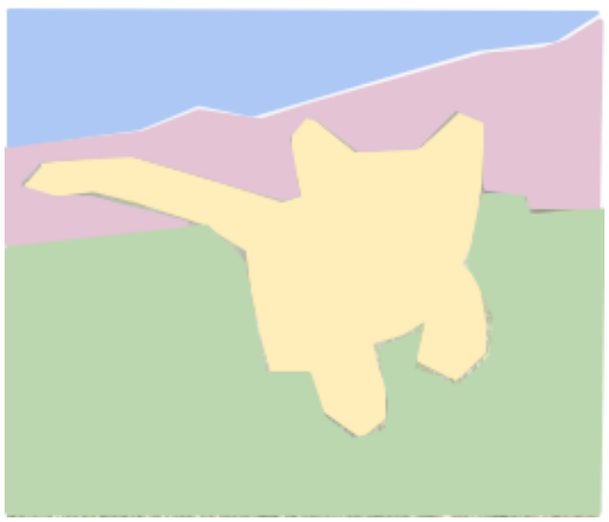
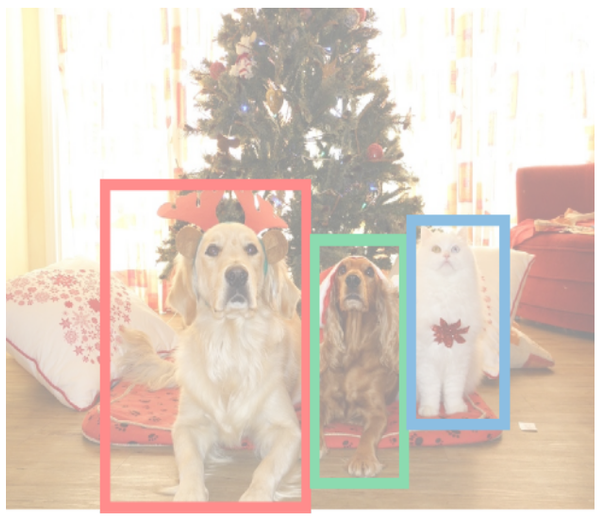


Image credit: R. Girshick

Computer vision tasks

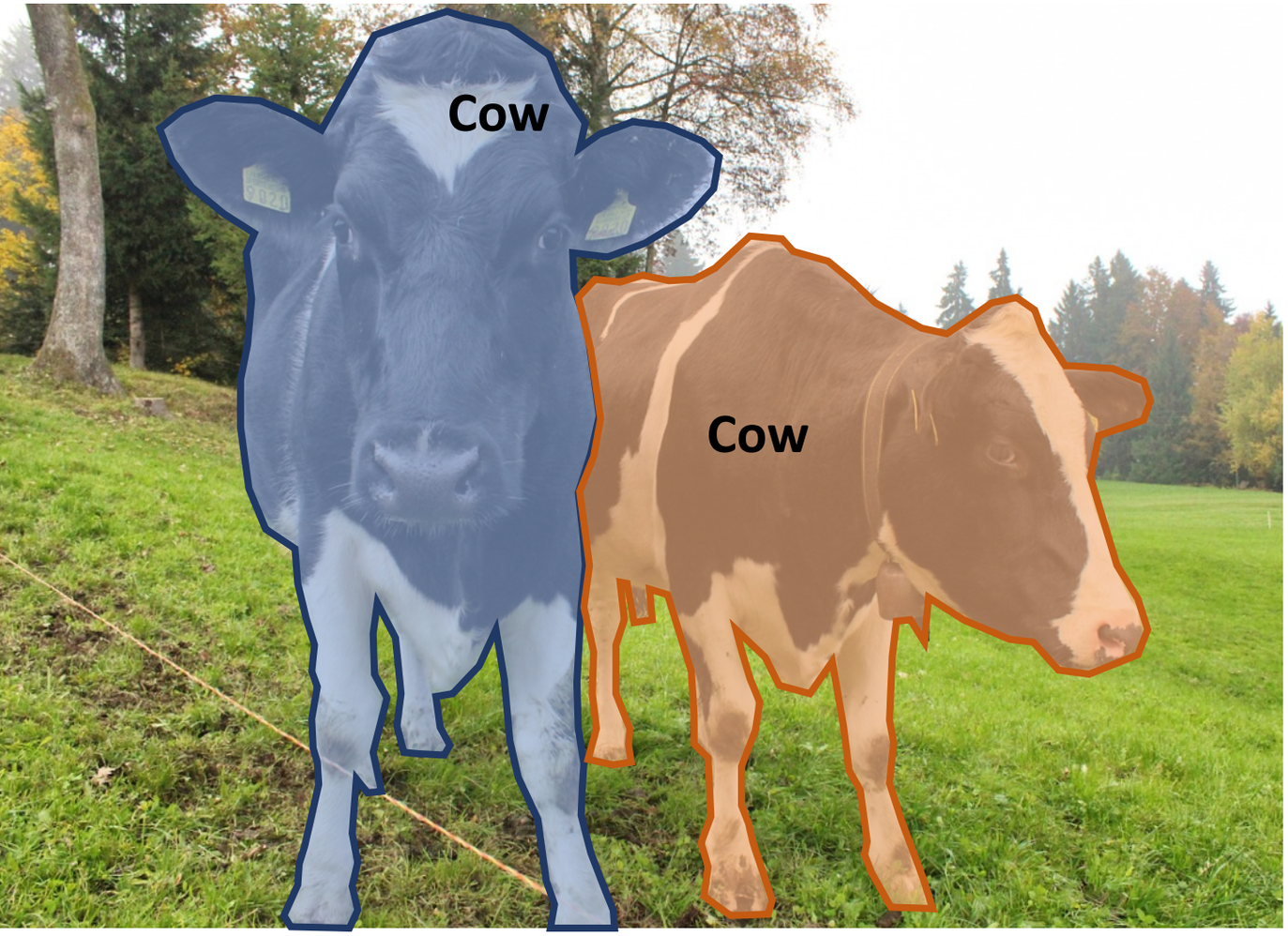
Classification	Semantic Segmentation	Object Detection	**Instance Segmentation**
			
CAT	GRASS, CAT, TREE, SKY	DOG, DOG, CAT	DOG, DOG, CAT
No spatial extent	No objects, just pixels	Multiple Objects	

Computer vision tasks

Classification	Semantic Segmentation	Object Detection	**Instance Segmentation**
			
CAT	GRASS, CAT, TREE, SKY	DOG, DOG, CAT	DOG, DOG, CAT
No spatial extent	No objects, just pixels	Multiple Objects	

kind of a "hybrid task"

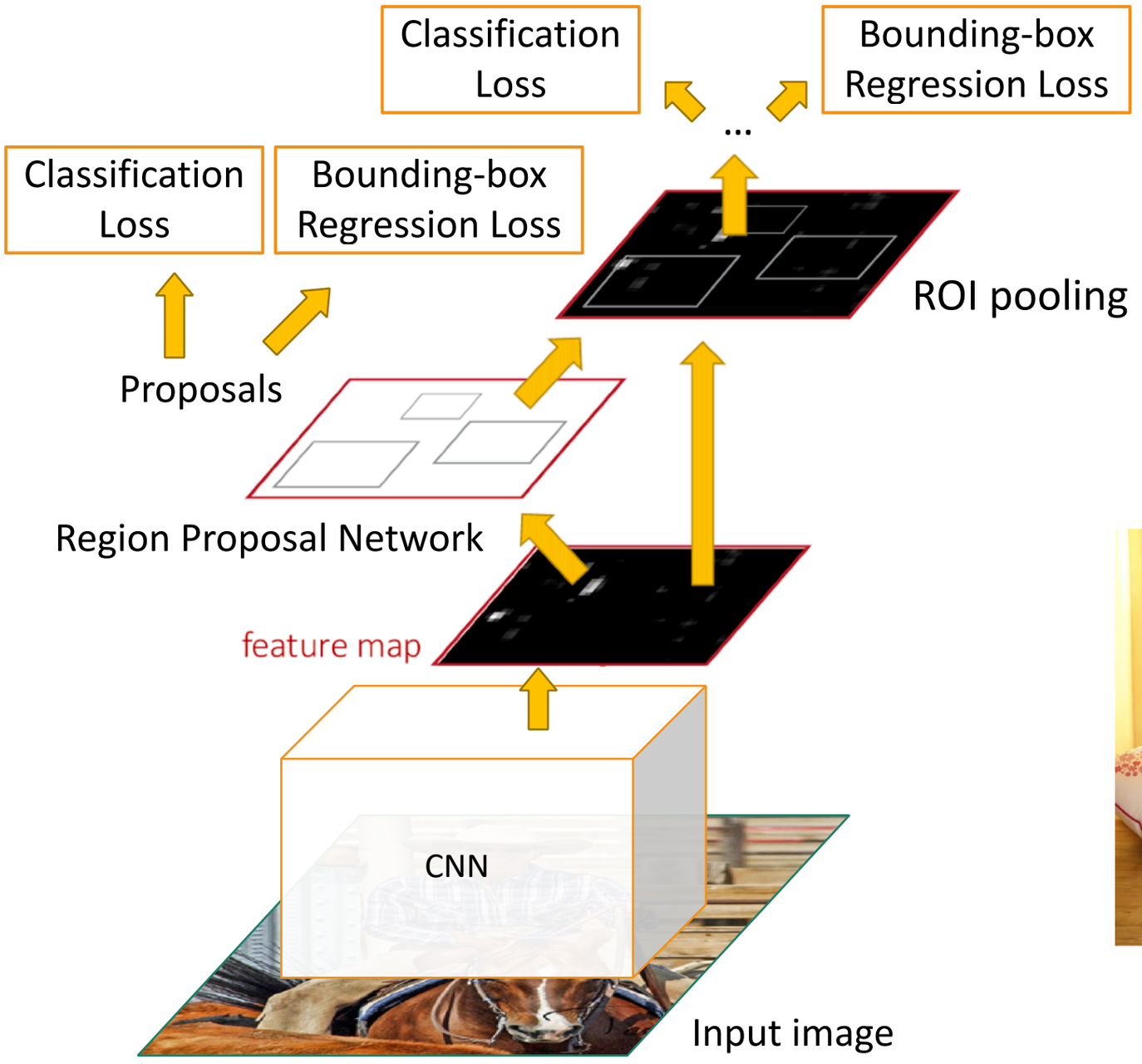
Instance segmentation

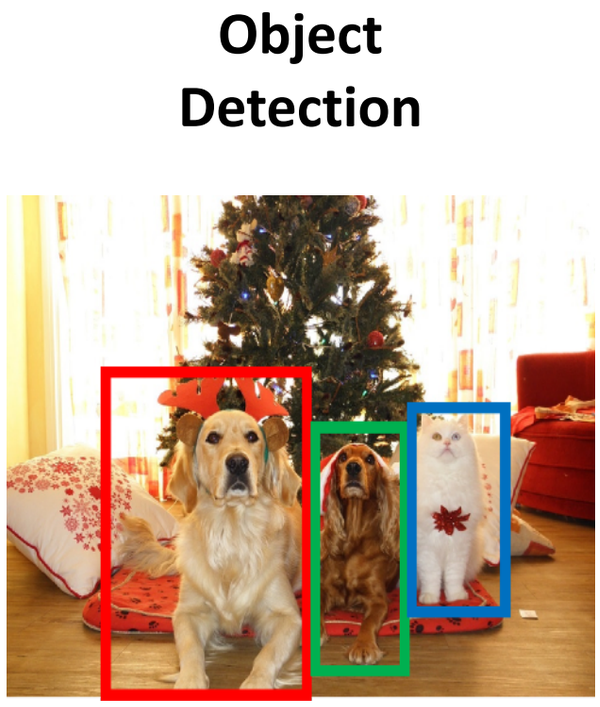


- **Instance Segmentation:** Detect all objects in the image, and identify the pixels that belong to each object

- **Approach:** Perform object detection, then predict a segmentation mask for each object!

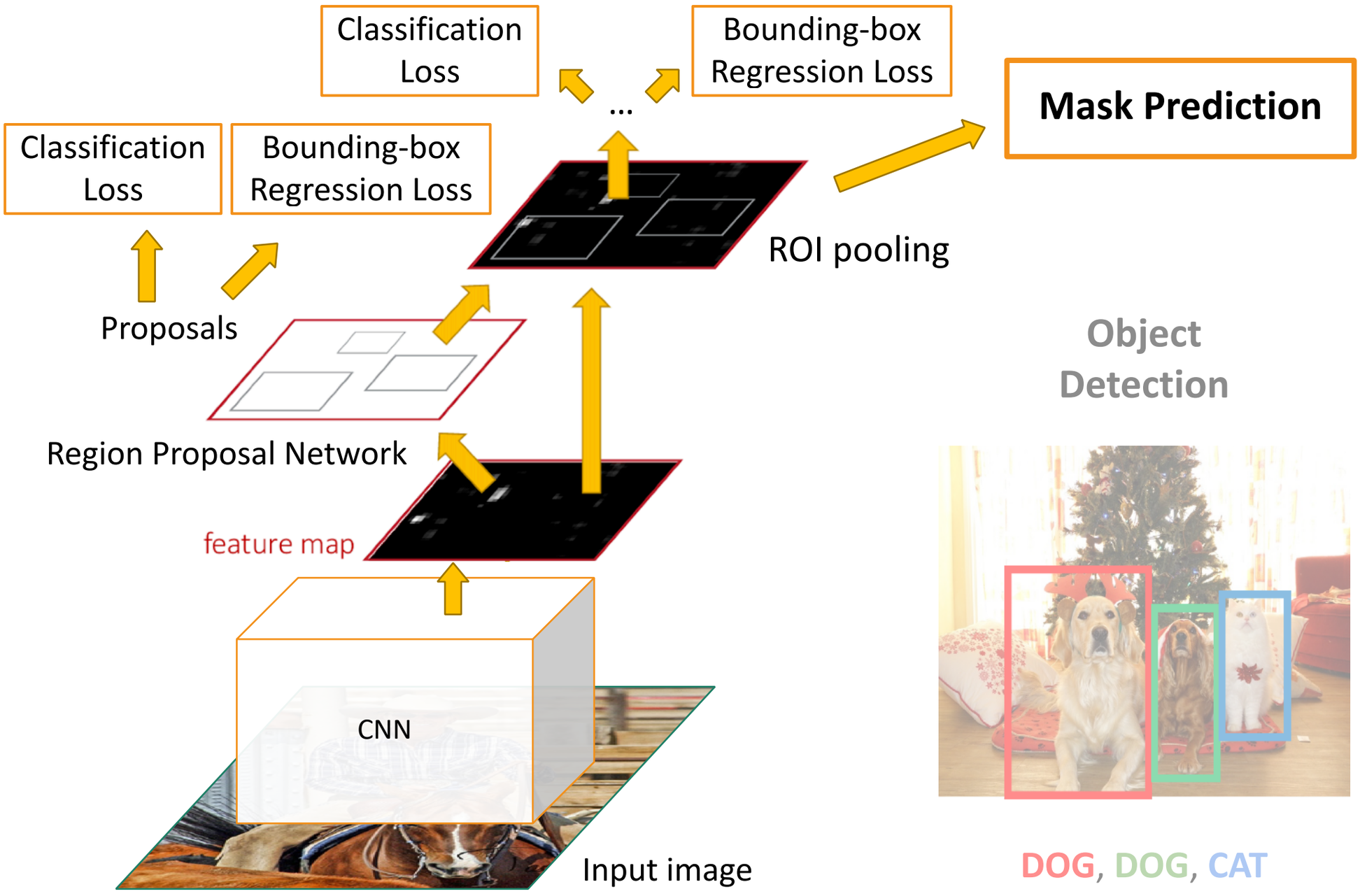
Object detection: Faster R-CNN

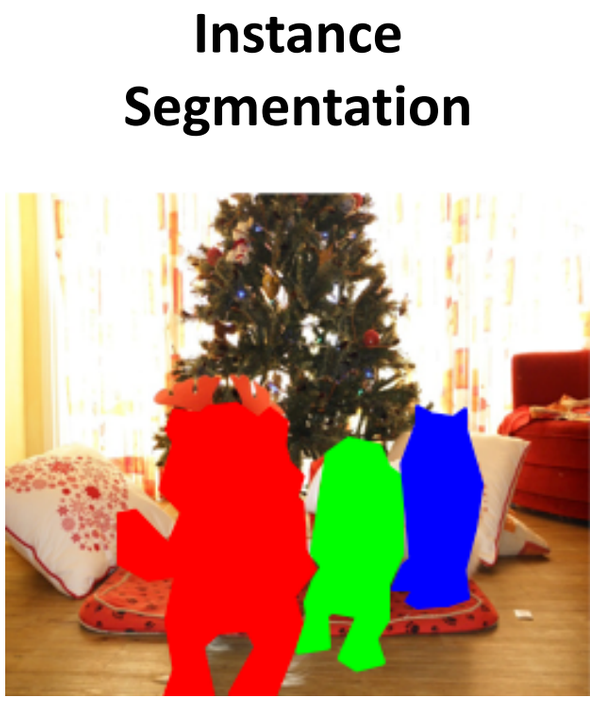




Ren *et al*, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

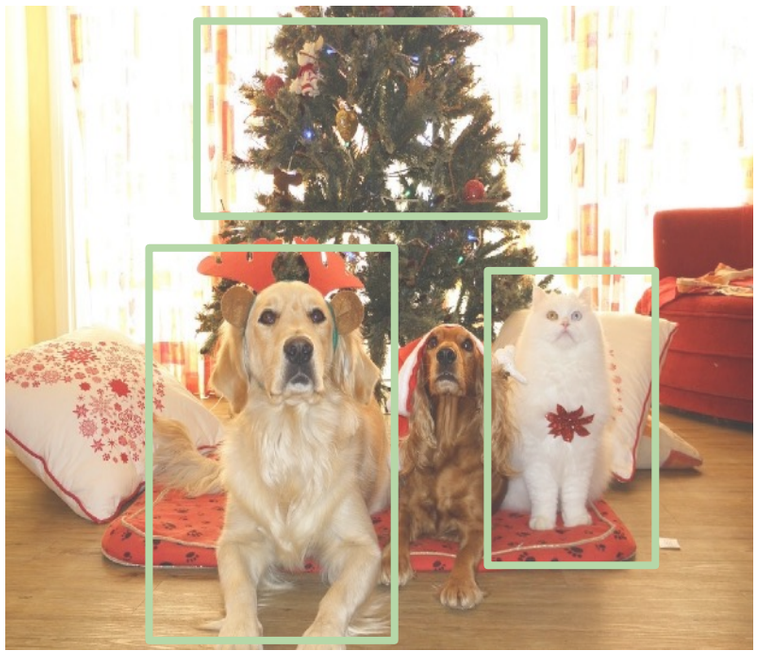
Instance segmentation: Mask R-CNN

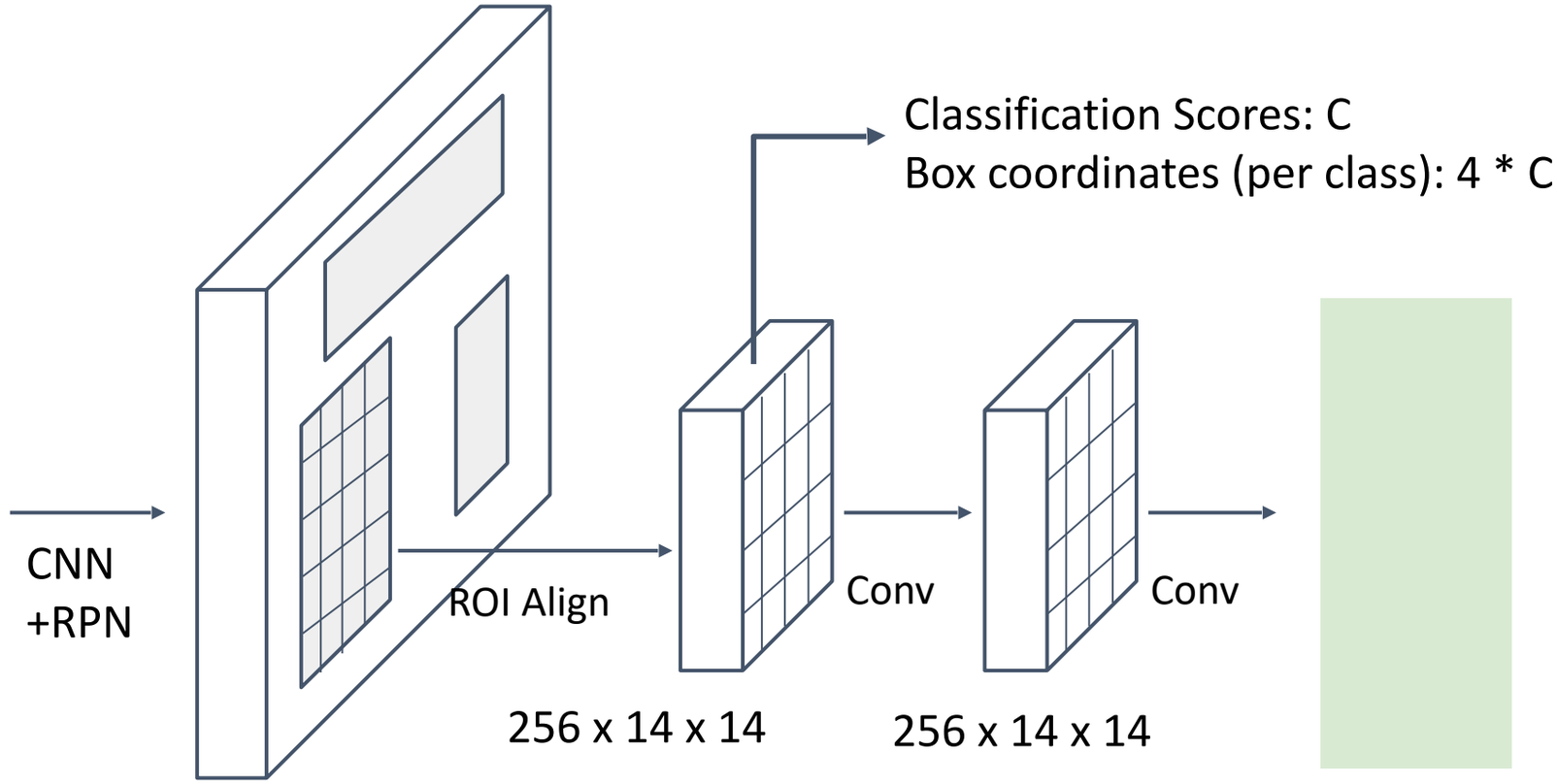




He *et al*, "Mask R-CNN", ICCV 2017

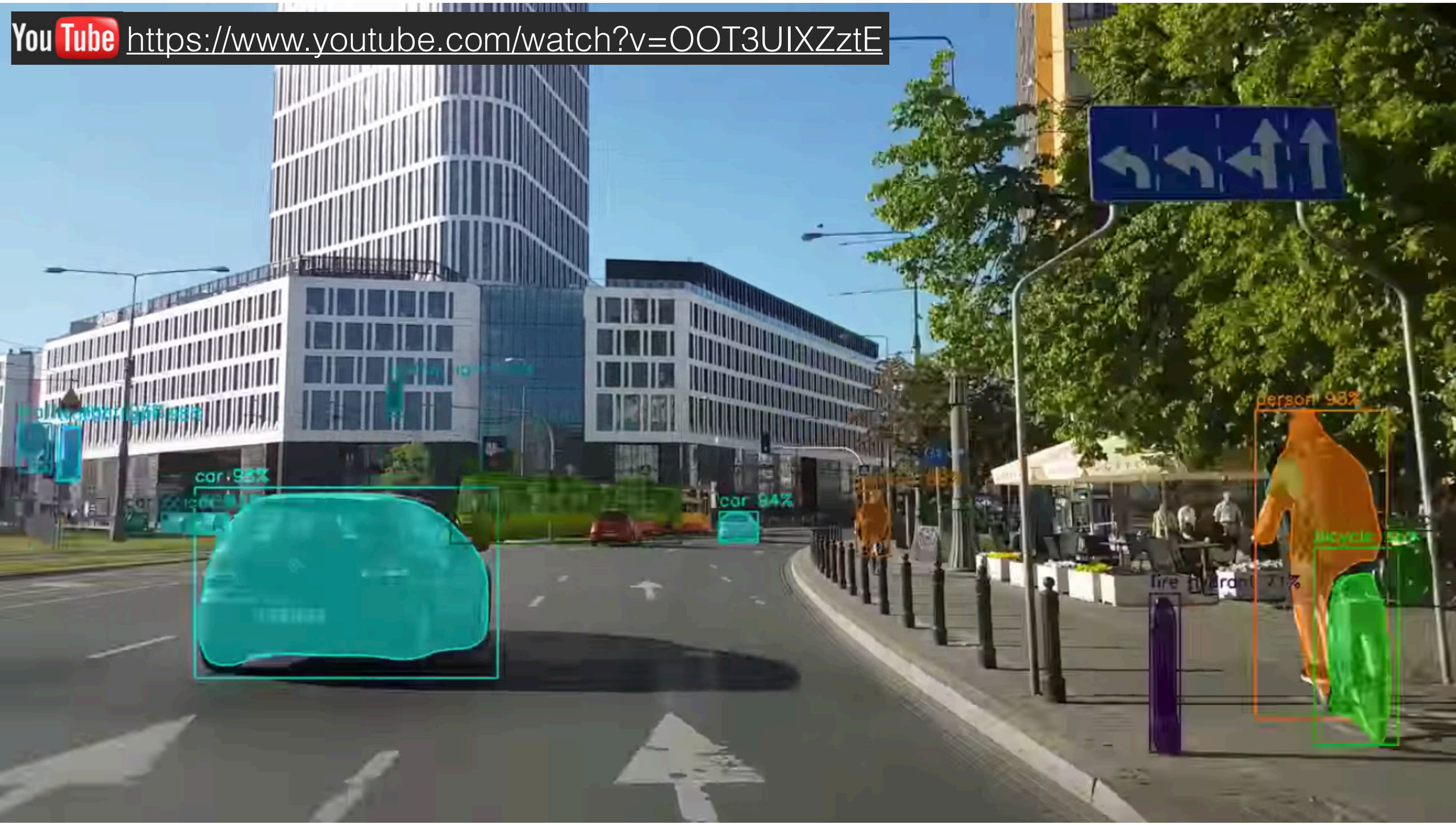
Instance segmentation: Mask R-CNN





He *et al*, “Mask R-CNN”, ICCV 2017

Demo: Mask R-CNN



Contact

- **Office:** Torre Archimede, room 6CD3
- **Office hours** (ricevimento)**:** Friday 9:00-11:00

✉ lamberto.ballan@unipd.it

🏠 http://www.lambertoballan.net

🏠 http://vimp.math.unipd.it

@ twitter.com/lambertoballan