CAP5415 Computer Vision

Yogesh S Rawat

yogesh@ucf.edu

HEC-241

Questions?

Object Detection

Lecture 13

Agenda

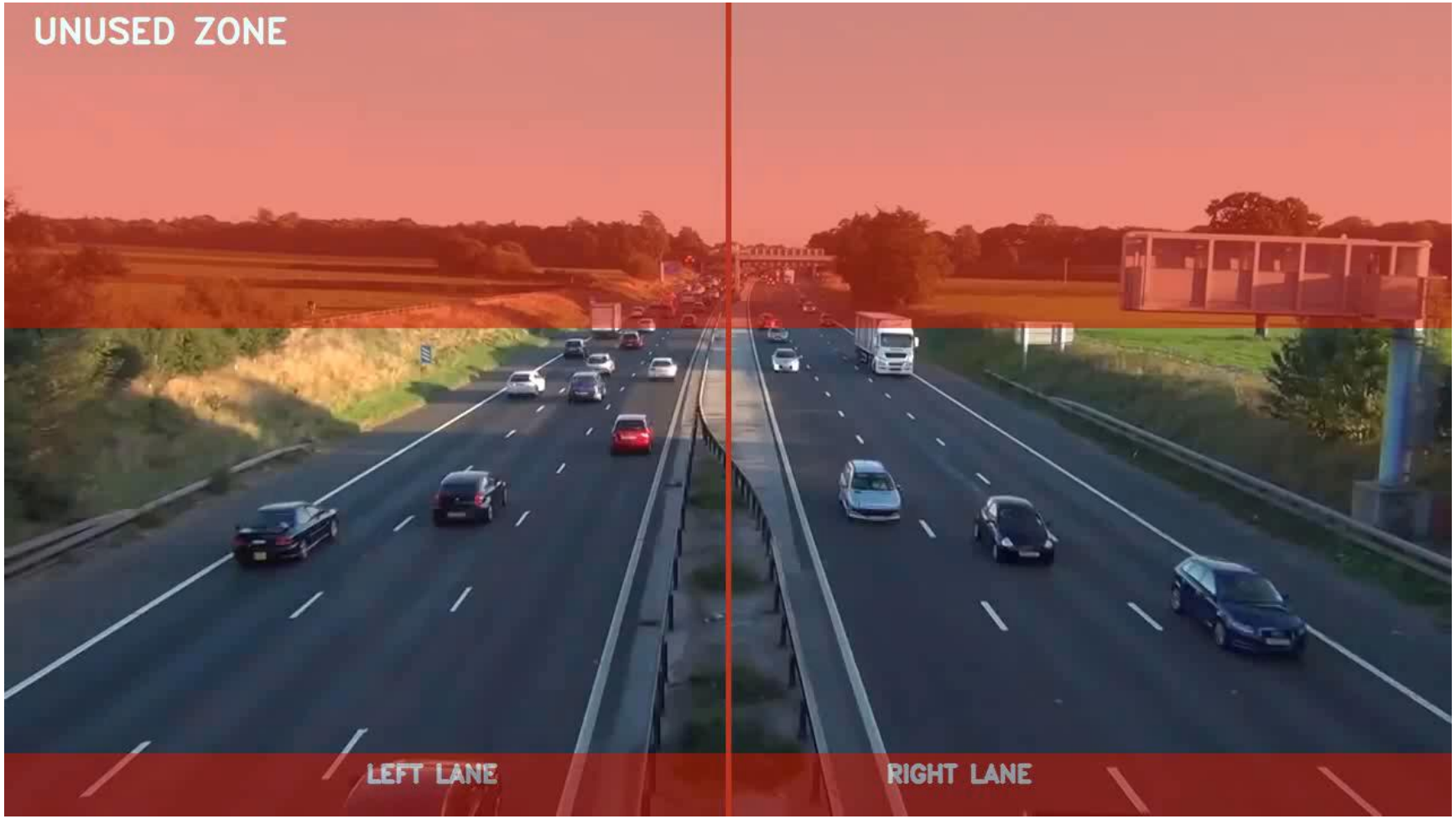
- Basics of object detection
- Simple solution
  - Sliding window approach
  - Evaluation
- CNN based solution
  - Basic CNN solution
  - RCNN
  - Fast-RCNN
  - Faster-RCNN

Object Detection

Lecture 13

Basics of object detection

Current state



Object recognition

- Problem: Given an image A, does A contain an image of a person?





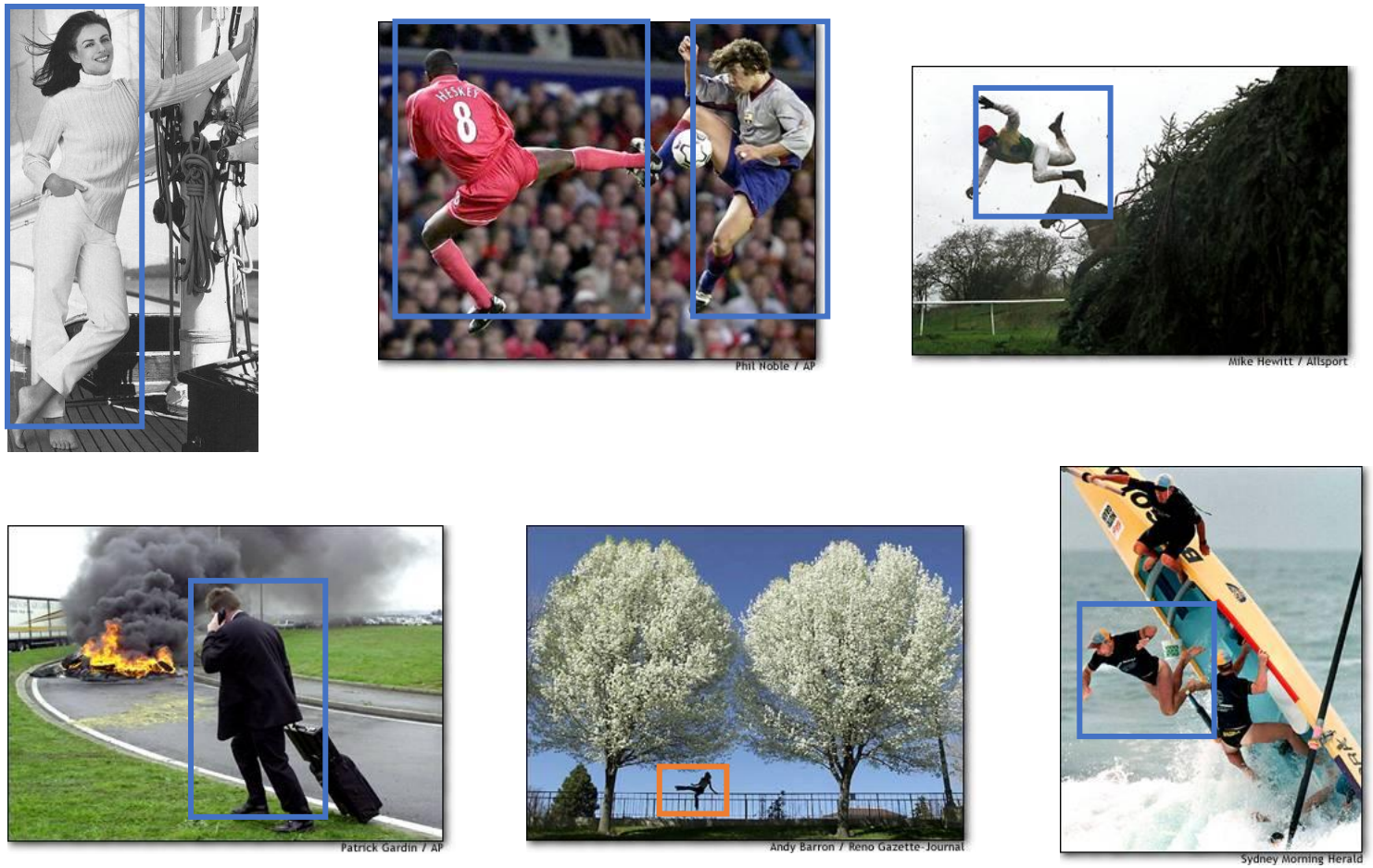






YES

Object localization/detection



Human detection

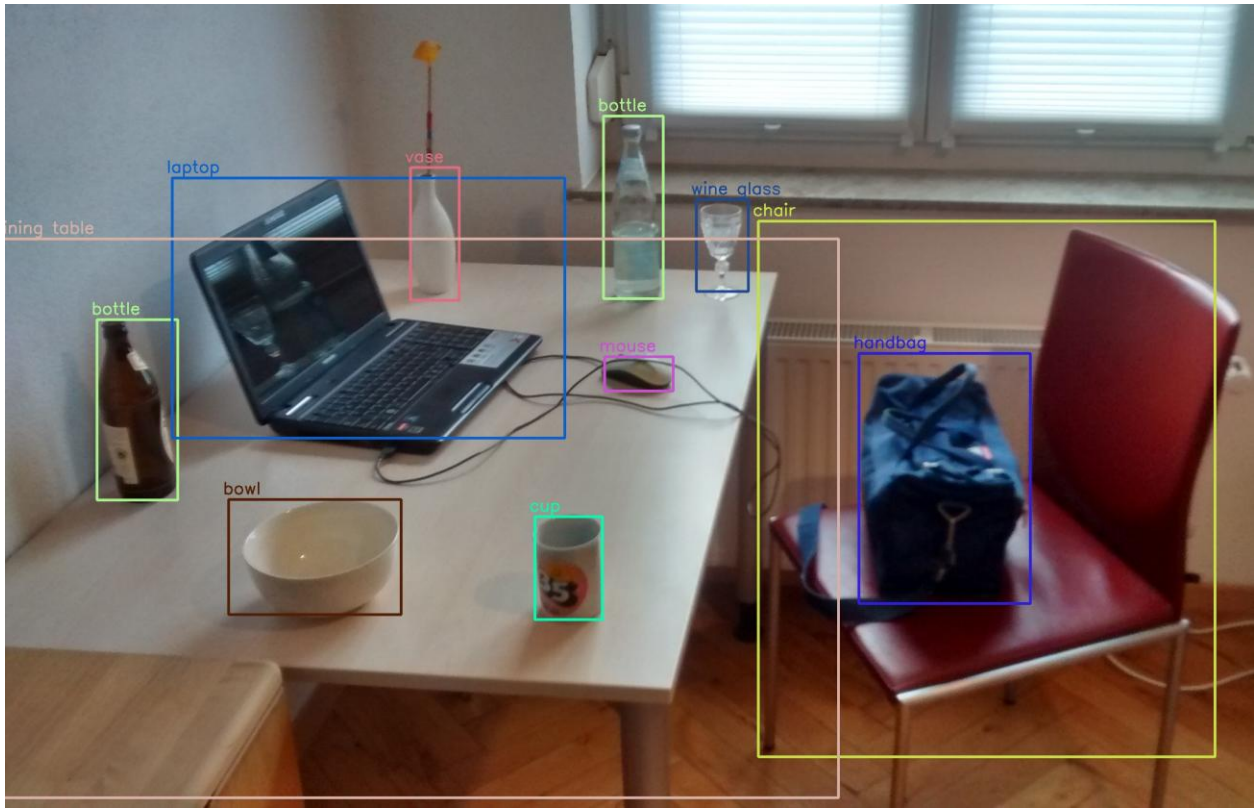
- UAV images
- Surveillance images





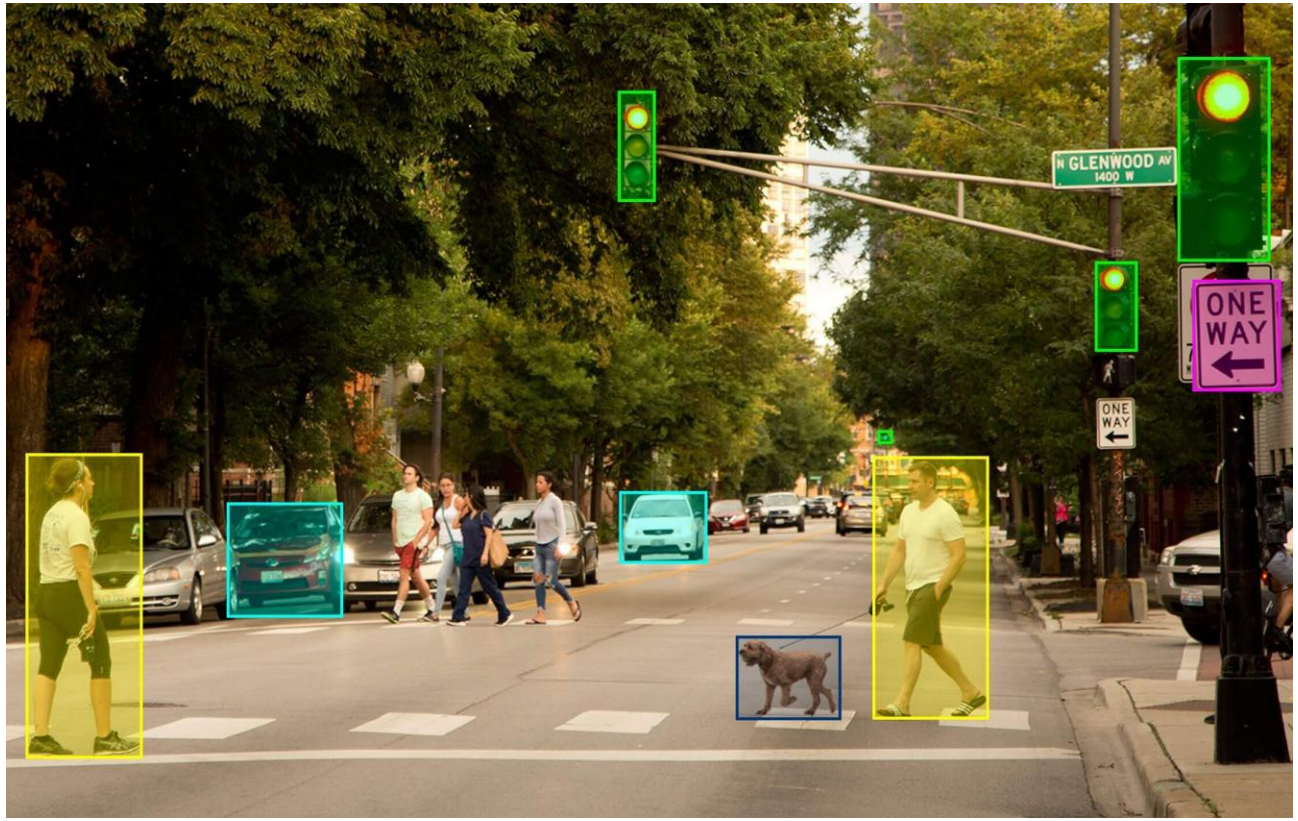
Object detection

- Multiple objects



Why detection?

- Self-driving car



Questions?

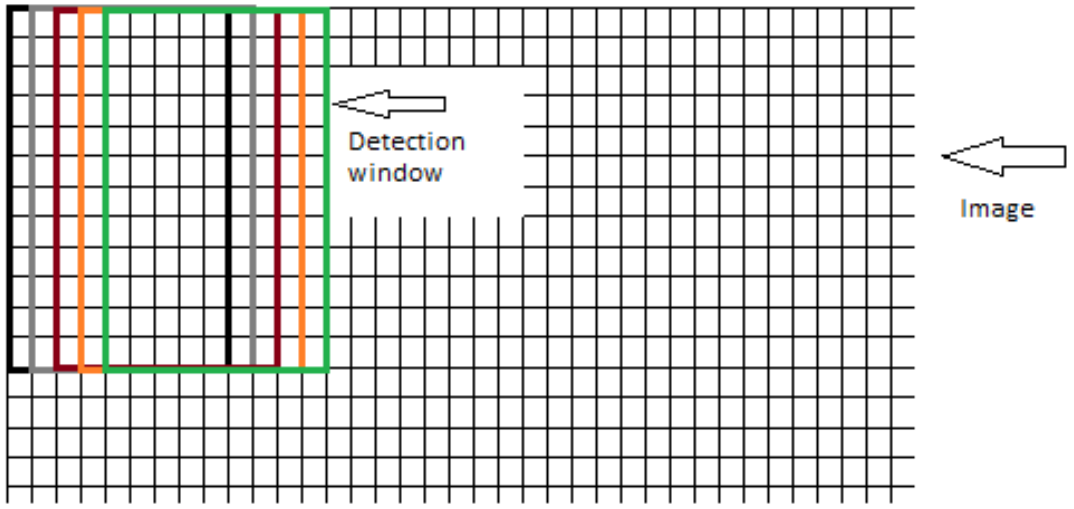
Object Detection

Lecture 13

A simple solution
Sliding window approach

Sliding window

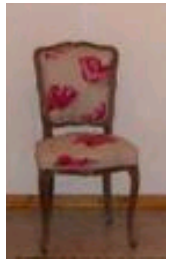
- Slide a box around image and classify each crop



- There are many interesting works
  - Viola and Jones (2001): Face detector (22K citations)
  - Dalal and Triggs (2005): HOG (33K citations)
  - Felzenswalb et al. (2010): Deformable part-based model (10K citations)

Naïve approach: Template Matching

This is a chair



Find the chair in this image



Output of correlation

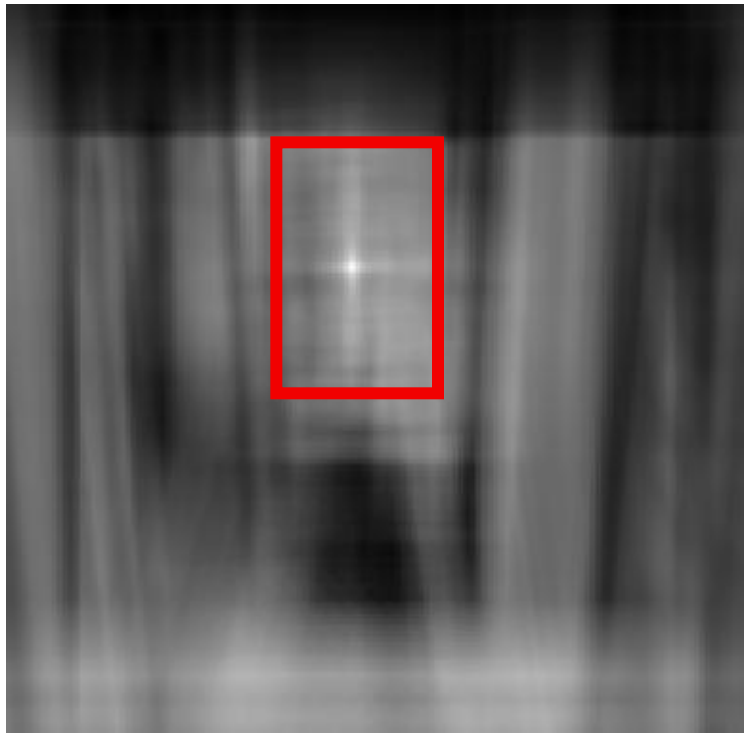
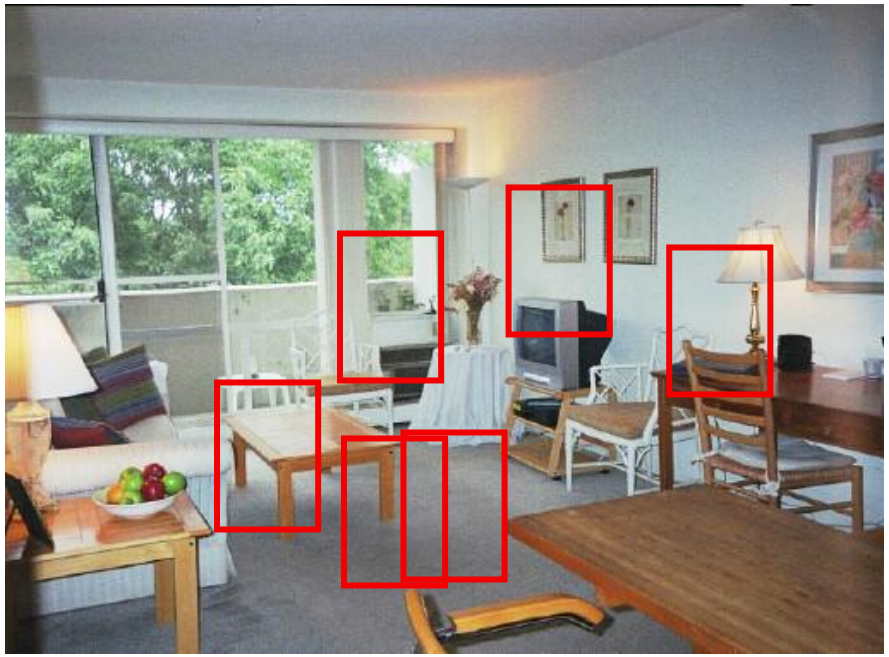
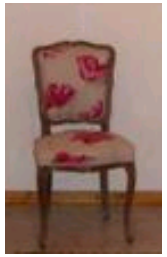


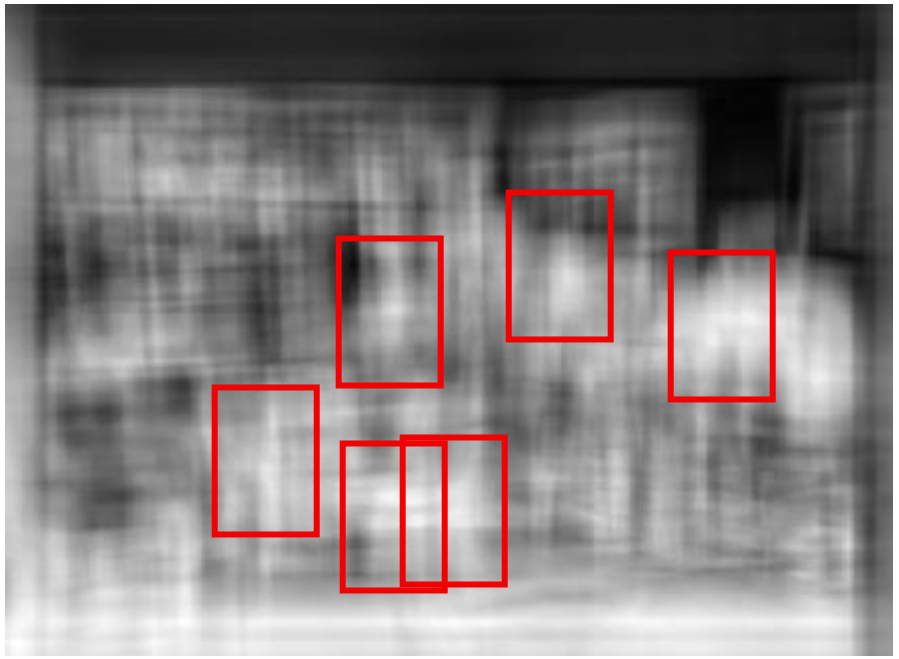
Image source: Antonio Torralba

Template Matching

Find the chair in this image







Epic fail!
Simple template matching is not going to make it

Image source: Antonio Torralba

Sliding window

- General approach
  - Scan all possible locations
  - Extract features
  - Classify features
  - Post-processing



Sliding window

- General approach
  - **Scan all possible locations**
  - Extract features
  - Classify features
  - Post-processing

- Inspect all windows
- Size of window is fixed



Sliding window

- General approach
  - **Scan all possible locations**
  - Extract features
  - Classify features
  - Post-processing

- People at different sizes?



Objects can be of very different sizes (scales), even in the same image. How do we deal with that?

Sliding window

- General approach
  - **Scan all possible locations**
  - Extract features
  - Classify features
  - Post-processing



- Down-scale the image

Sliding window

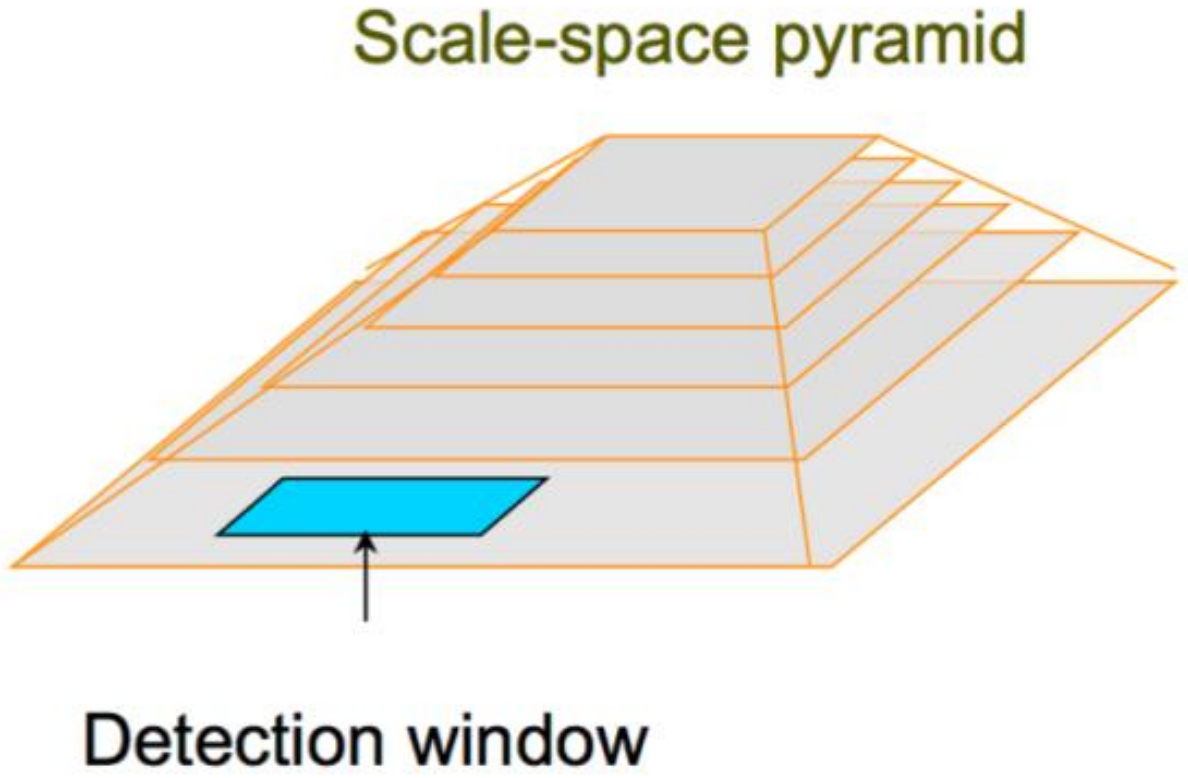
- General approach
  - **Scan all possible locations**
  - Extract features
  - Classify features
  - Post-processing



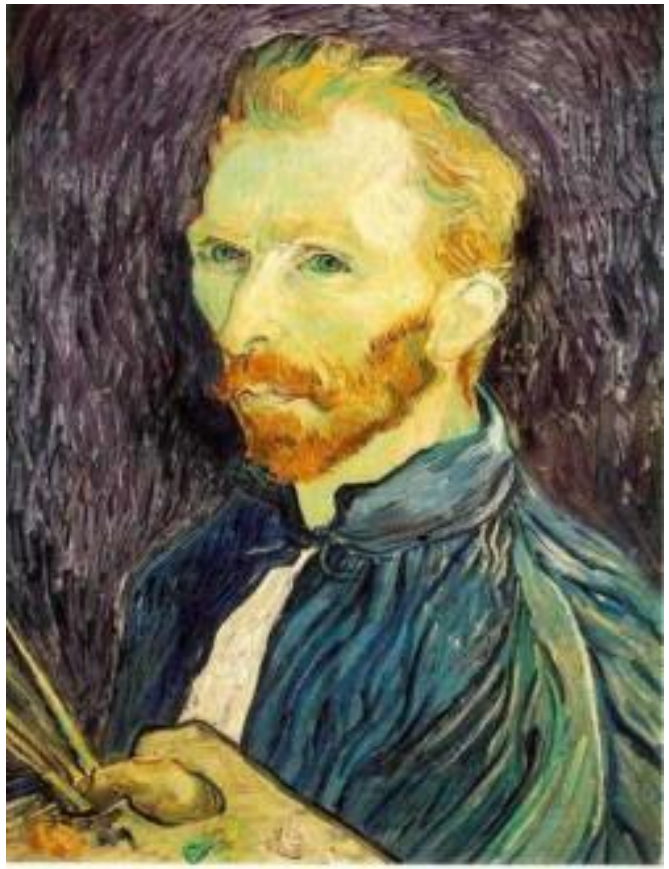
- Keep shrinking

Scale-space pyramid

- Full image pyramid



Gaussian pre-filtering



Gaussian 1/2



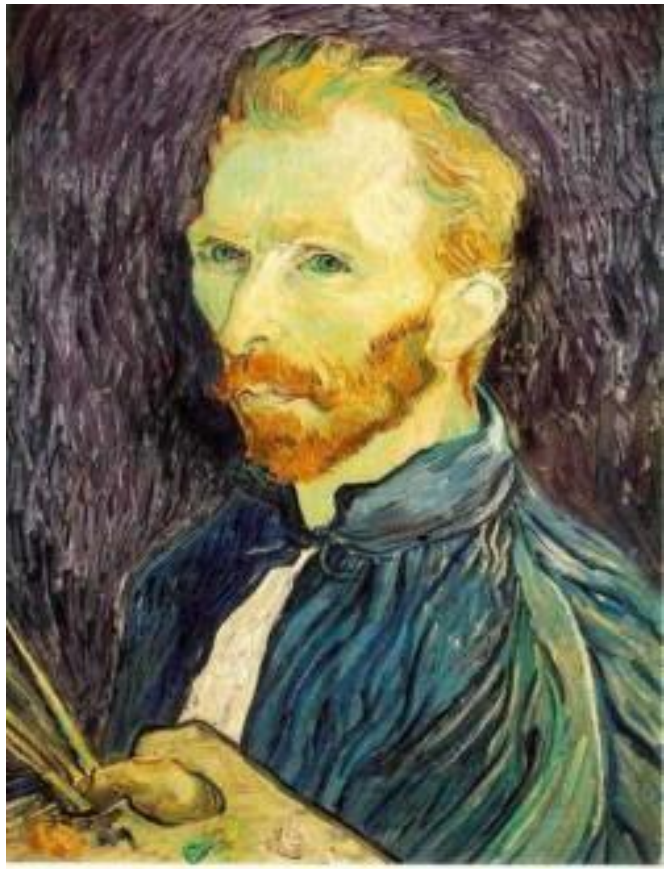
G 1/4



G 1/8

Solution: filter the image, *then* subsample

Gaussian pre-filtering



Gaussian 1/2



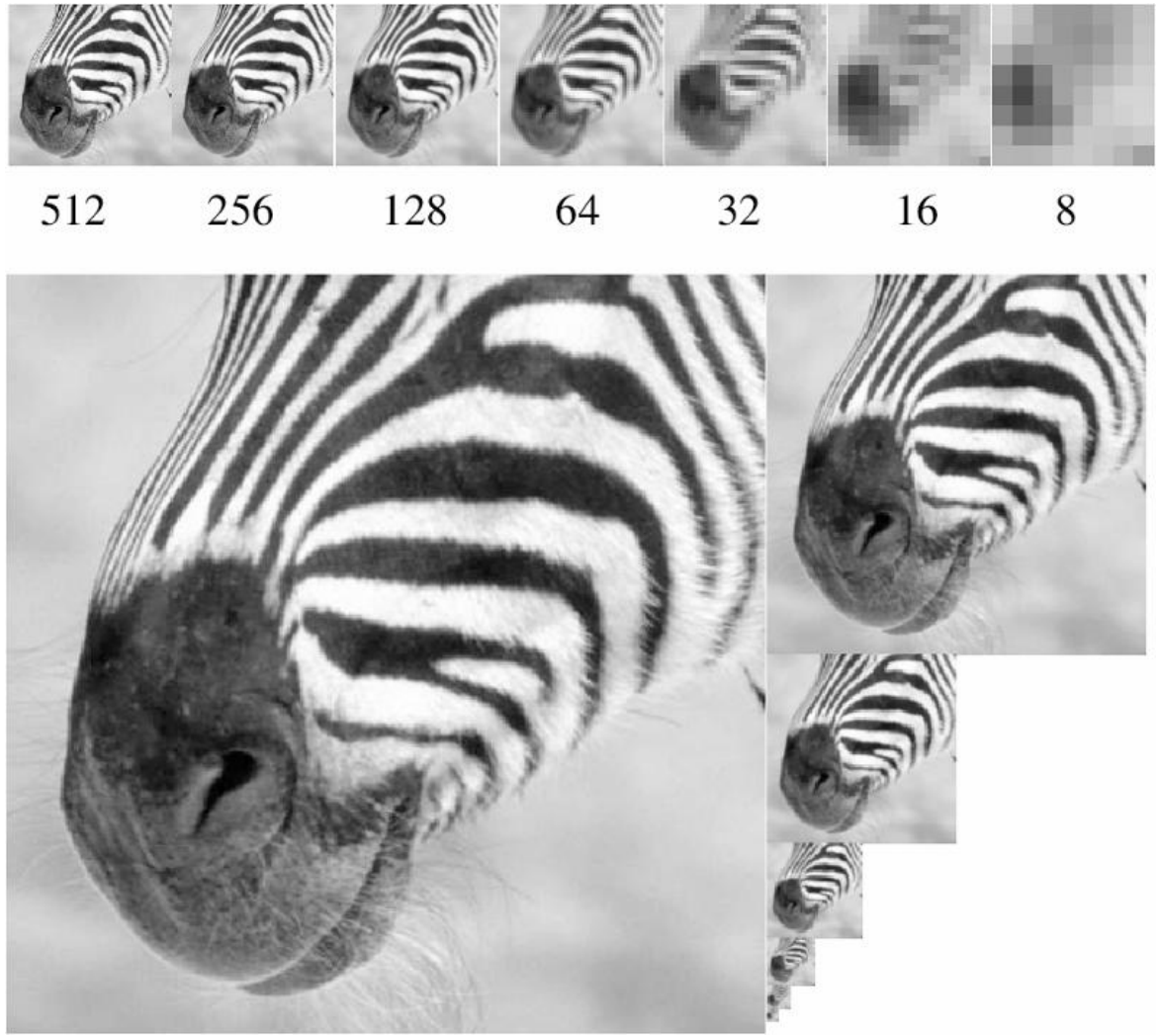
G 1/4



G 1/8

Solution: filter the image, *then* subsample

Image pyramid



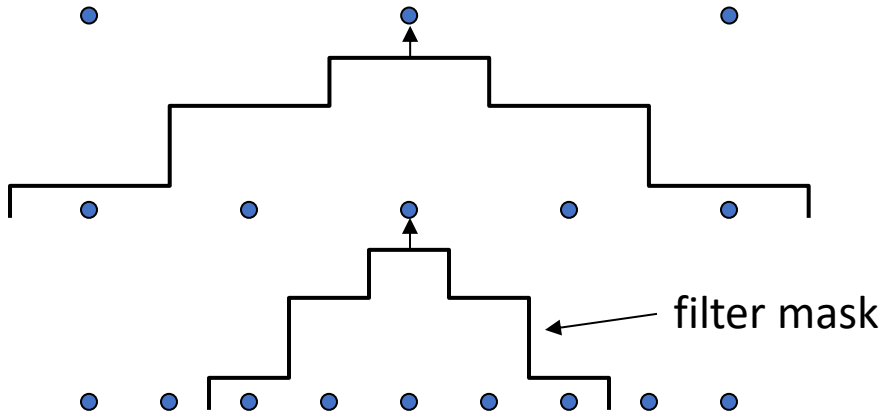
A bar in the big images is a hair on the zebra's nose;

in smaller images, a stripe;

in the smallest, the animal's nose

Figure from David Forsyth

Gaussian pyramid construction



Repeat

- Filter
- Subsample

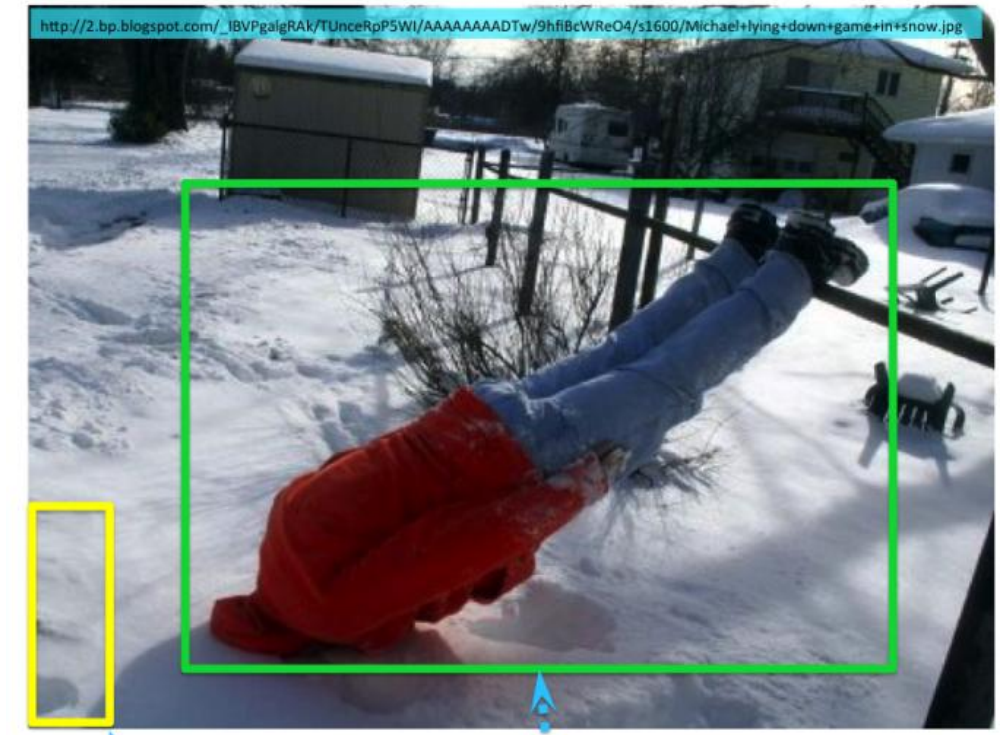
Until minimum resolution reached

- can specify desired number of levels (e.g., 3-level pyramid)

The whole pyramid is only 4/3 the size of the original image!

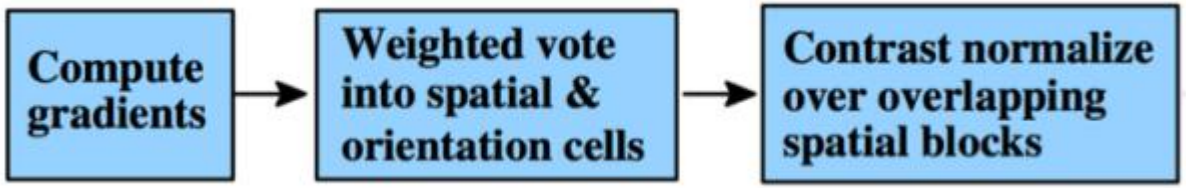
Challenges

- Different aspect-ratio?



Sliding window

- General approach
  - Scan all possible locations
  - **Extract features**
  - Classify features
  - Post-processing

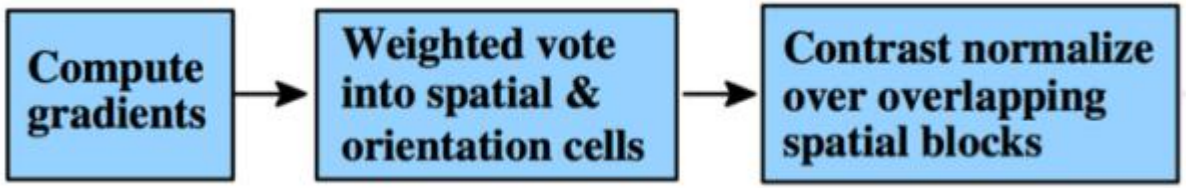


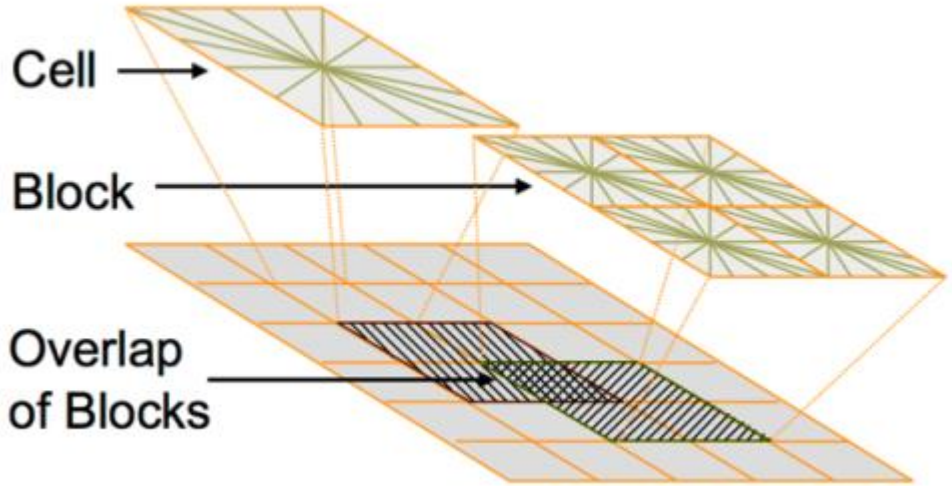


- HOG/SIFT

Sliding window

- General approach
  - Scan all possible locations
  - **Extract features**
  - Classify features
  - Post-processing





- Extract features

Sliding window

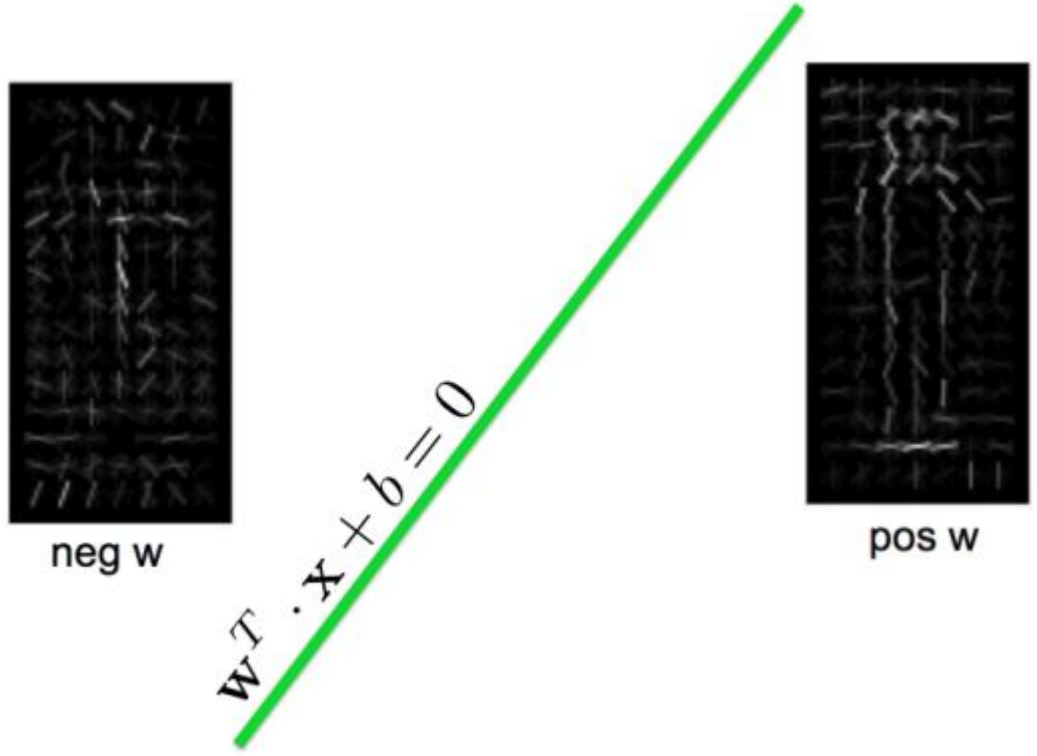
- General approach
  - Scan all possible locations
  - Extract features
  - **Classify features**
  - Post-processing



- Linear/SVM/NN

Sliding window

- General approach
  - Scan all possible locations
  - Extract features
  - **Classify features**
  - Post-processing
- Linear/SVM/NN



Sliding window

- General approach
  - Scan all possible locations
  - Extract features
  - **Classify features**
  - Post-processing



- Run classifier at all scales
- Use a threshold
- Detect all positives

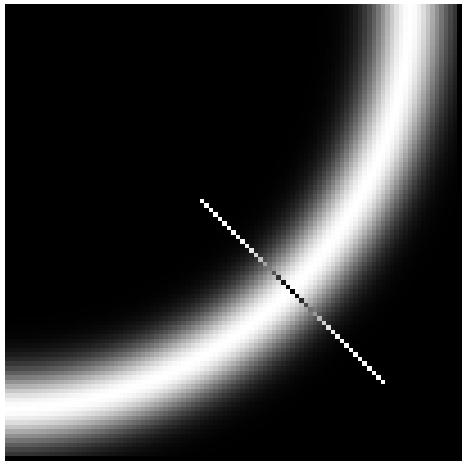
Sliding window

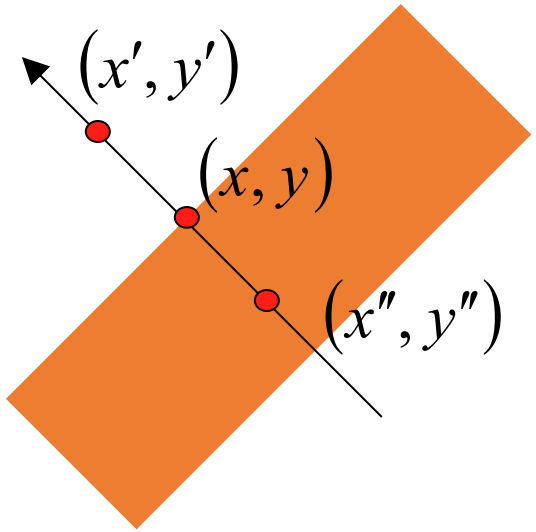
- General approach
  - Scan all possible locations
  - Extract features
  - Classify features
  - **Post-processing**

- Multiple detections
- Non-maxima suppression



Non-maximum suppression





$$M(x,y) = \begin{cases} |\nabla S|(x,y) & \text{if } |\nabla S|(x,y) > |\Delta S|(x',y') \\ & \& |\Delta S|(x,y) > |\Delta S|(x'',y'') \\ 0 & \text{otherwise} \end{cases}$$

x' and **x**'' are the neighbors of **x** along normal direction to an edge

Non-maxima suppression (NMS)

- Iterate over all detections
  - Pick the highest scoring box
- Find overlap
  - with all other boxes
- Remove boxes with high overlap
  - Threshold, usually 0.5

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$



Non-maxima suppression (NMS)

- Iterate over all detections
  - Pick the highest scoring box
- Find overlap
  - with all other boxes
- Remove boxes with high overlap
  - Threshold, usually 0.5



Non-maxima suppression (NMS)

- Iterate over all detections
  - Pick the highest scoring box
- Find overlap
  - with all other boxes
- Remove boxes with high overlap
  - Threshold, usually 0.5



Non-maxima suppression (NMS)

- Iterate over all detections
  - Pick the highest scoring box
- Find overlap
  - with all other boxes
- Remove boxes with high overlap
  - Threshold, usually 0.5



Non-maxima suppression (NMS)

- Iterate over all detections
  - Pick the highest scoring box
- Find overlap
  - with all other boxes
- Remove boxes with high overlap
  - Threshold, usually 0.5



Non-maxima suppression (NMS)

- Iterate over all detections
  - Pick the highest scoring box
- Find overlap
  - with all other boxes
- Remove boxes with high overlap
  - Threshold, usually 0.5



Questions?

Object Detection

Lecture 13

Evaluation

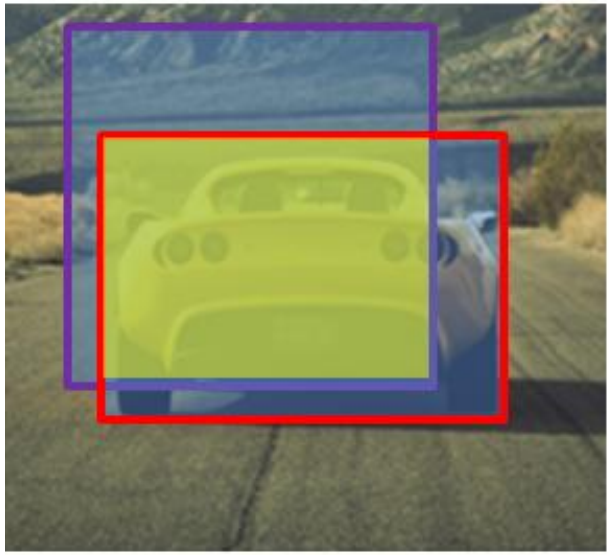
Evaluation

- How do we know if our approach is doing well?



Evaluation

- Intersection over union

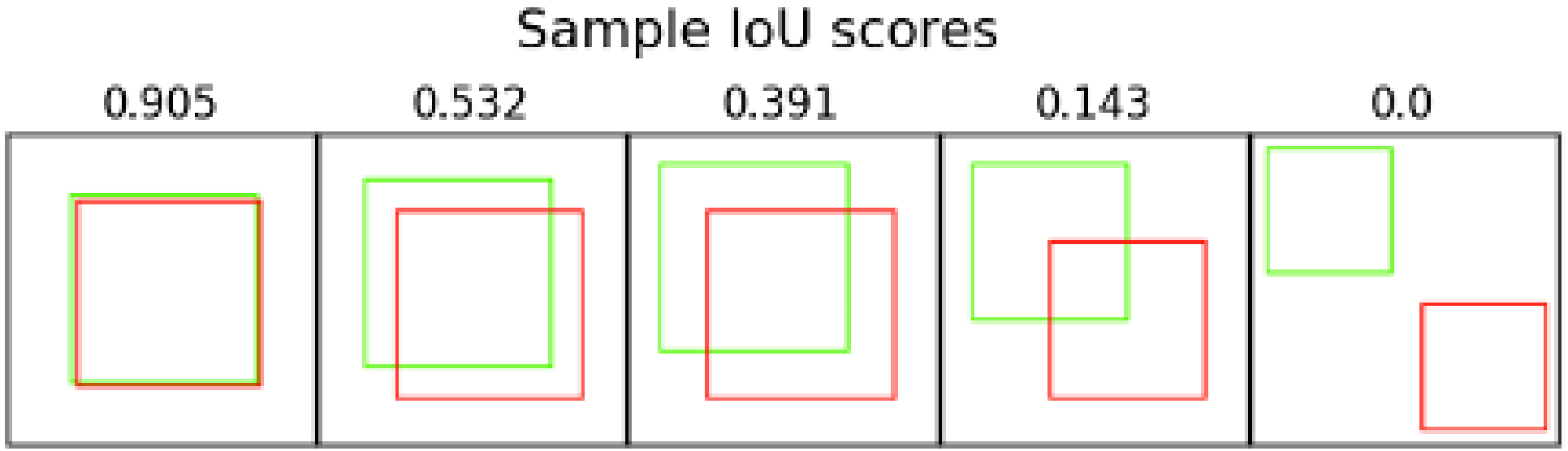


Intersection over union (IoU)

$$= \frac{\text{size of } \square}{\text{size of } \square}$$

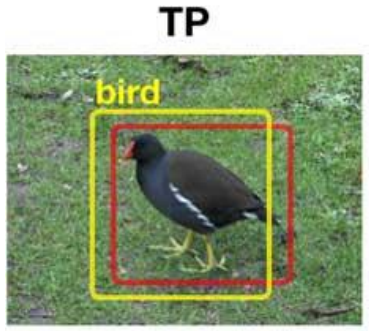
"Correct" if IoU $\geq 0,5$

Sample IoU



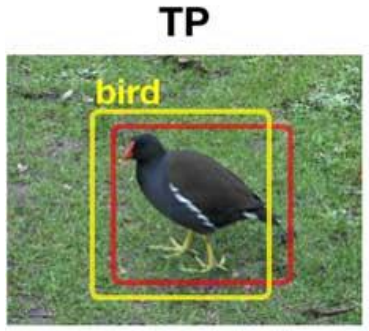
Evaluation

- True positives



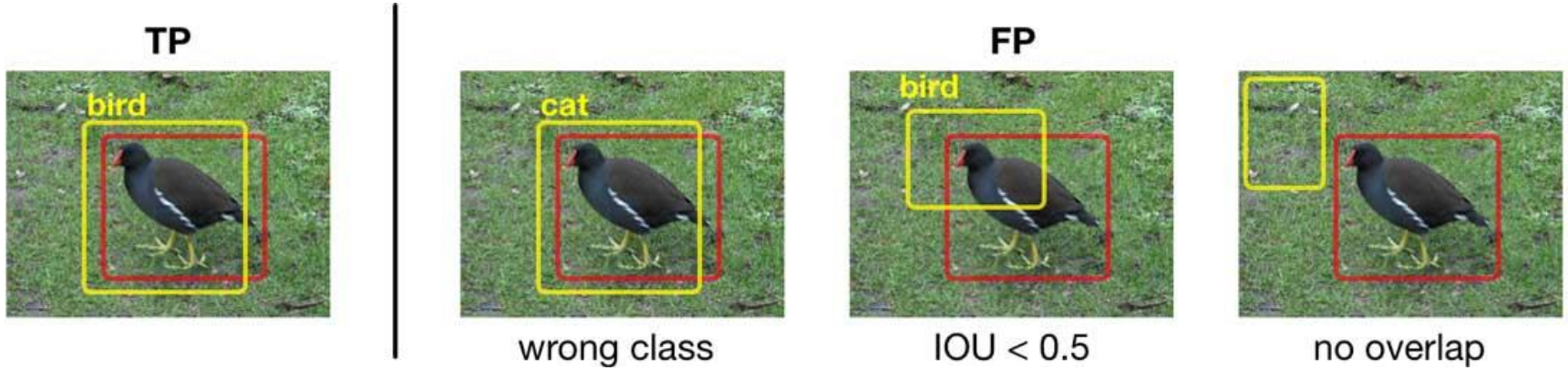
Evaluation

- True positives
- False positives



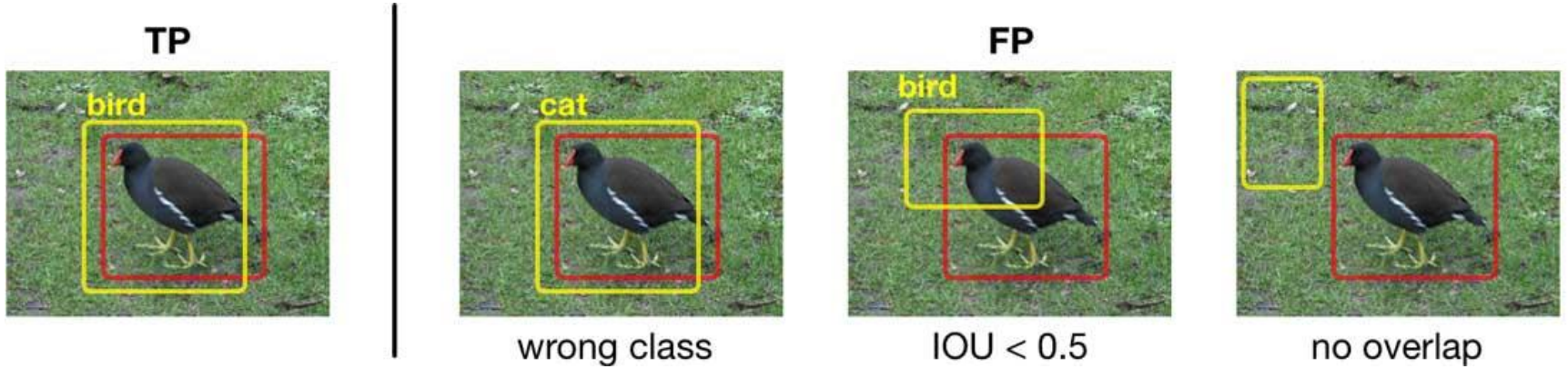
Evaluation

- True positives
- False positives



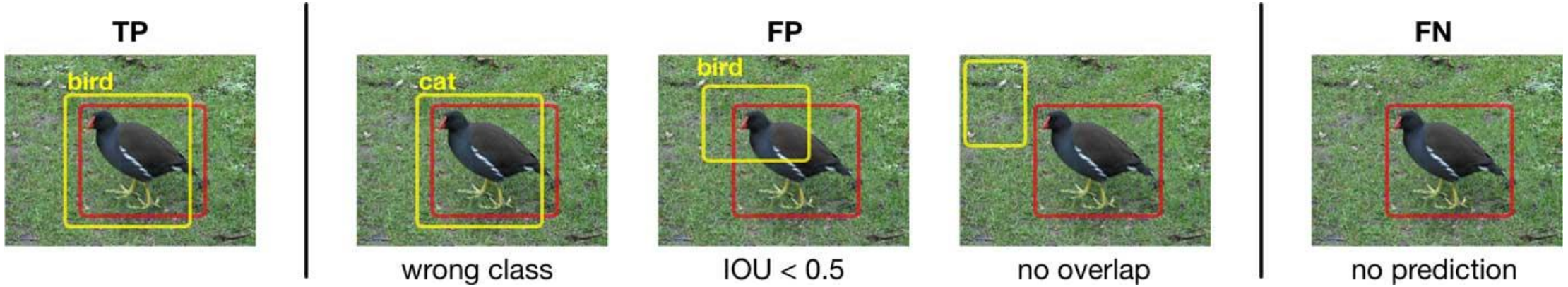
Evaluation

- True positives
- False positives
- False negatives



Evaluation

- True positives
- False positives
- False negatives



Evaluation

- Only one is correct



Evaluation

- Precision
  - Precision is the ability of a model to identify **only** the relevant objects.
  - It is the percentage of correct positive predictions and is given by:

$$precision = \frac{true\ positives}{true\ positives + false\ positives}$$

- Recall
  - Recall is the ability of a model to find all the relevant cases
  - It is the percentage of true positive detected among all relevant ground truths and is given by:

$$recall = \frac{true\ positives}{true\ positives\ + false\ negatives}$$

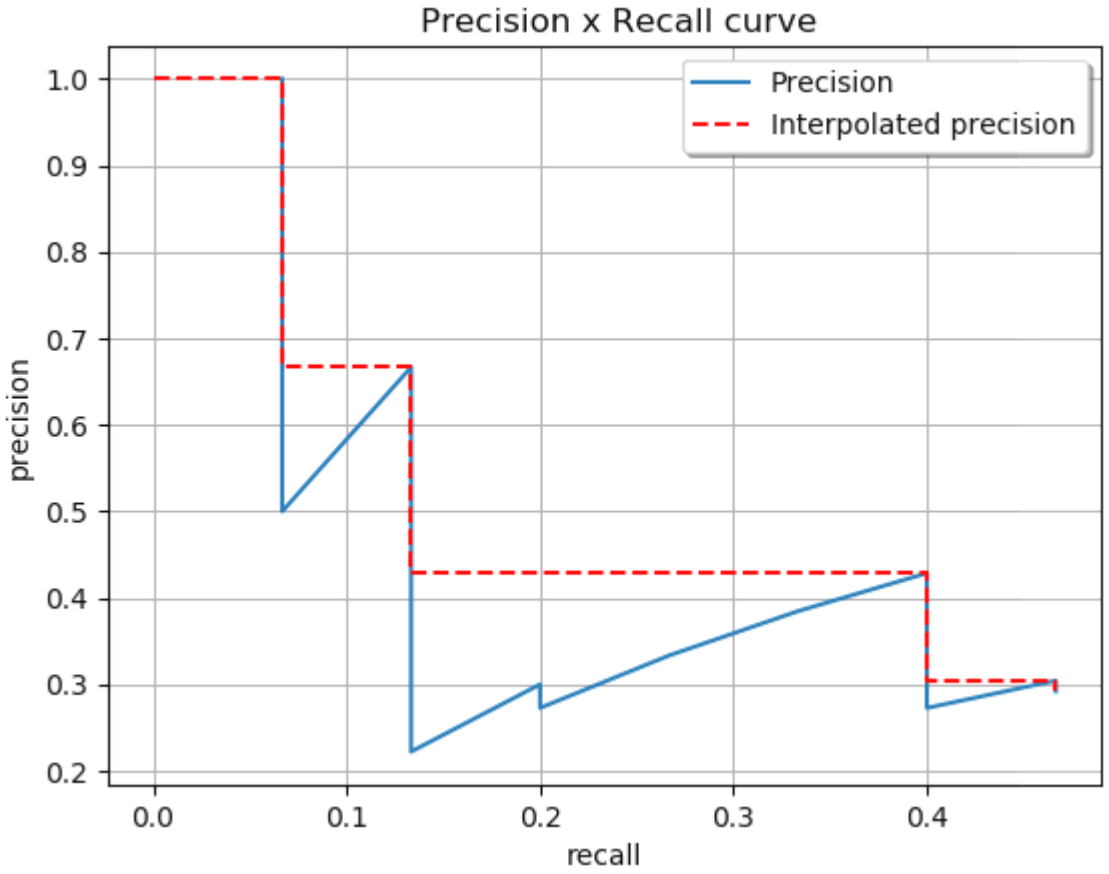
Evaluation

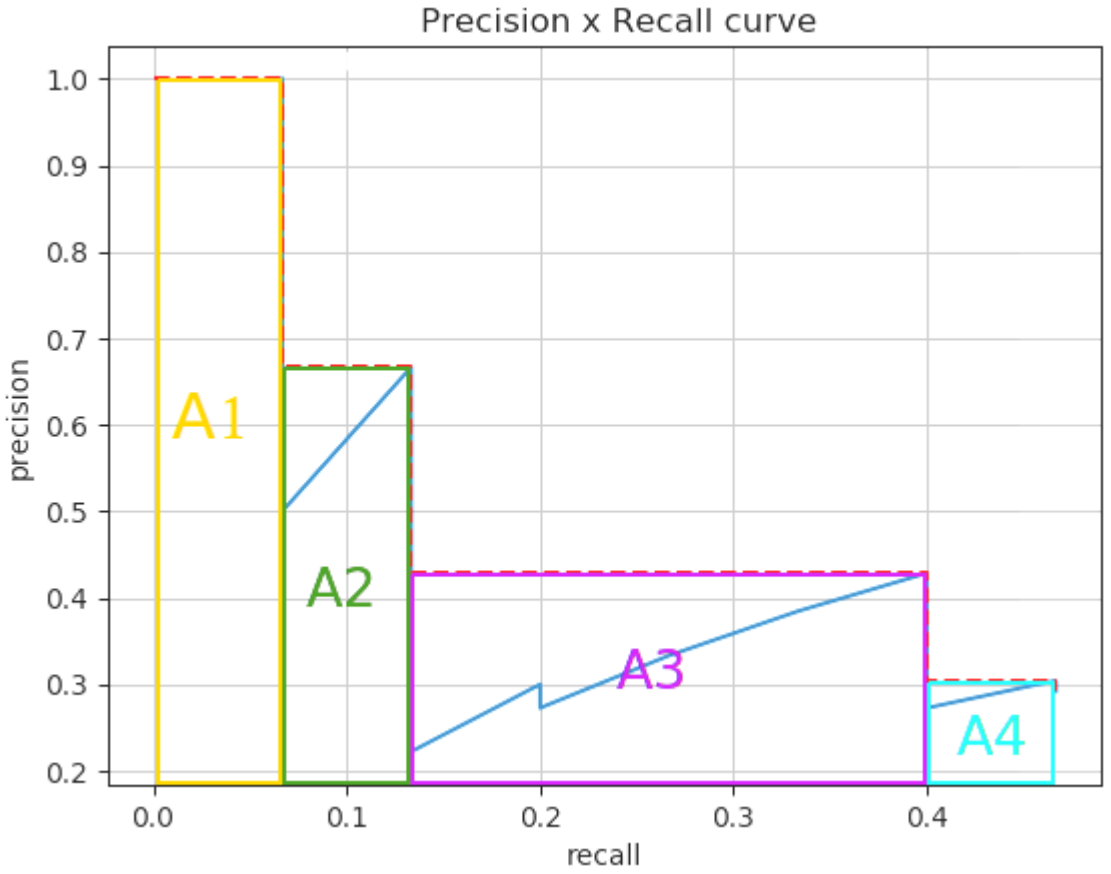
- Sort all predicted boxes (for all images)
  - According to scores
- For each k (location) in the list
  - Compute recall and precision

Images	Detections	Confidences	TP	FP	Acc TP	Acc FP	Precision	Recall
Image 5	R	95%	1	0	1	0	1	0.0666
Image 7	Y	95%	0	1	1	1	0.5	0.0666
Image 3	J	91%	1	0	2	1	0.6666	0.1333
Image 1	A	88%	0	1	2	2	0.5	0.1333
Image 6	U	84%	0	1	2	3	0.4	0.1333
Image 1	C	80%	0	1	2	4	0.3333	0.1333
Image 4	M	78%	0	1	2	5	0.2857	0.1333

https://github.com/rafaelpadilla/Object-Detection-Metrics

Average precision (AP)





mAP: average AP over multiple classes

Questions?

Pop Quiz?

CAP5415 Computer Vision

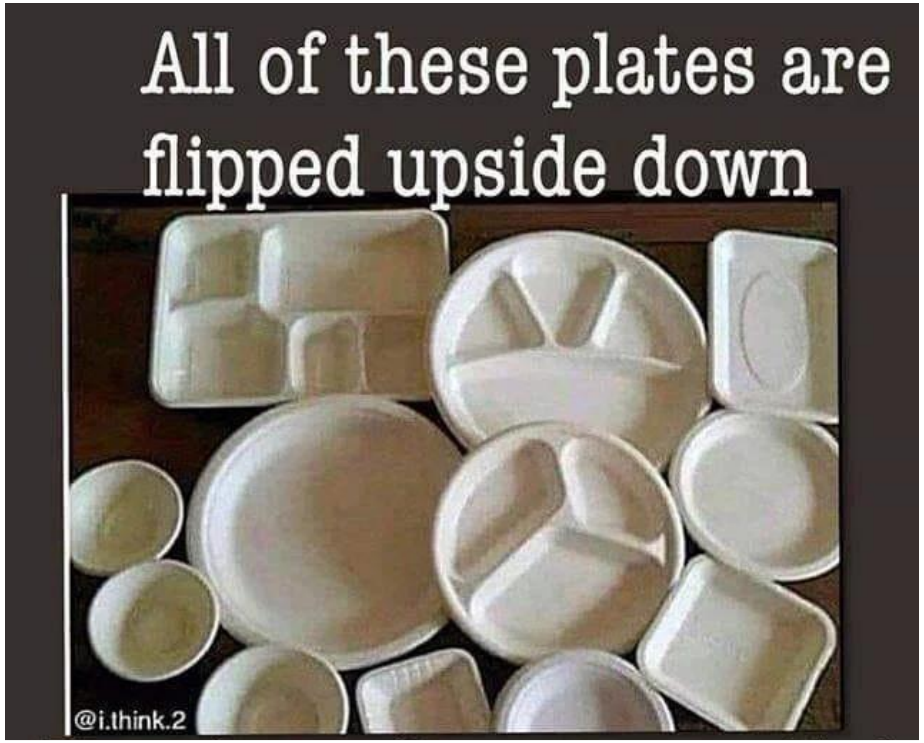
Yogesh S Rawat

yogesh@ucf.edu

HEC-241

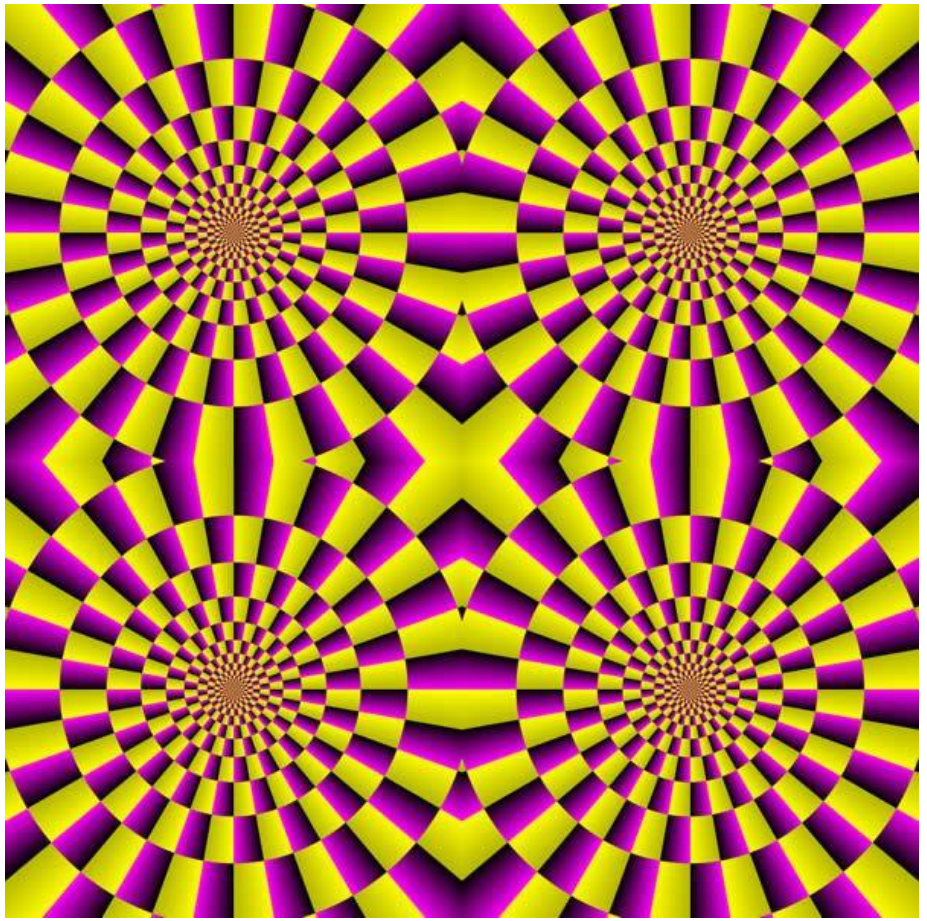
Questions?

Human vision vs machine vision



Source reddit

Human vision vs machine vision



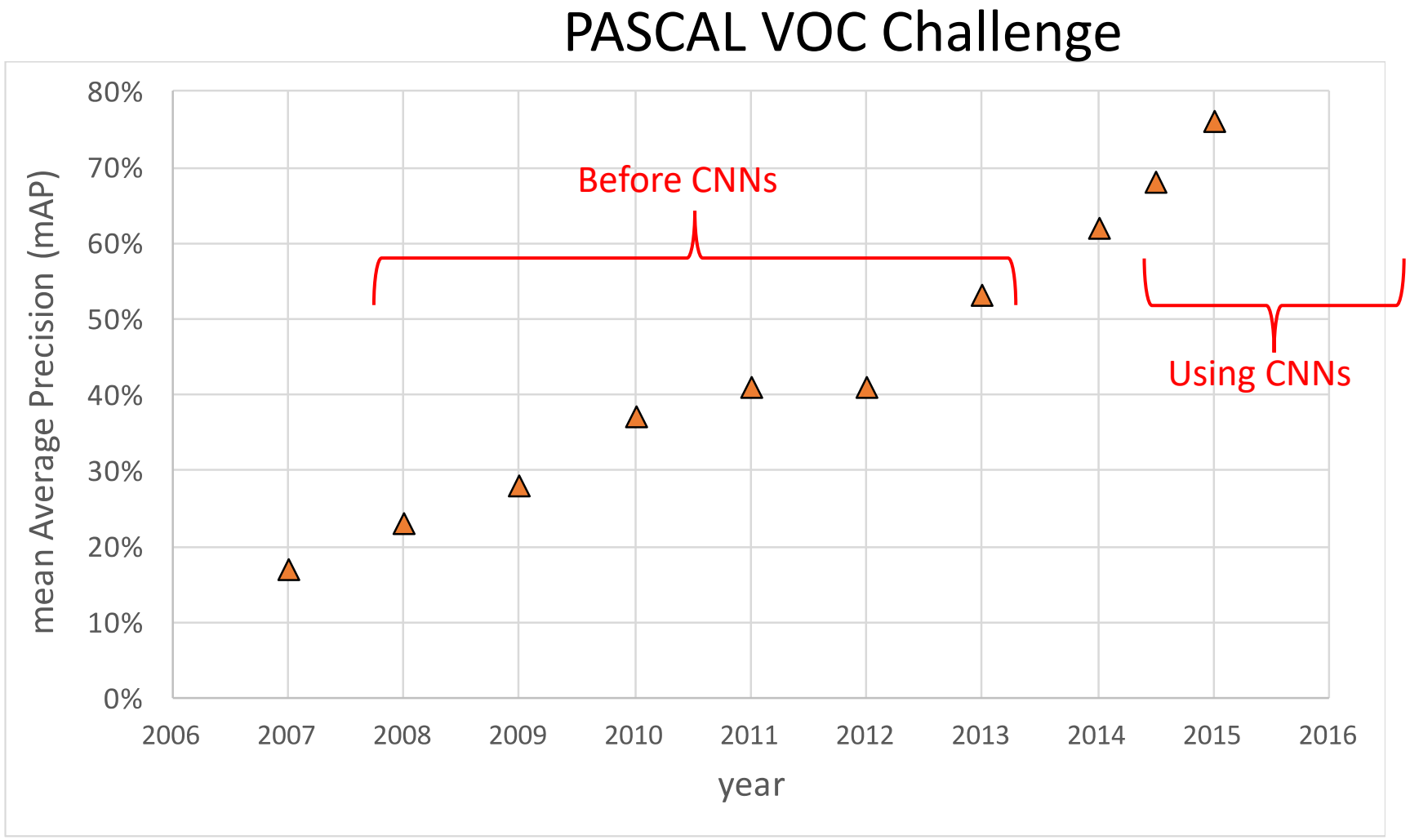
Source: https://www.pinterest.com/pin/illusionary-motion--138133913558846406/

Object Detection

Lecture 13

CNN based approach
A simple solution

Object detection progress over years



Deformable Parts Models (before the CNN era)

Source: R. Girshick

Object Detection: Datasets

	PASCAL VOC (2010)	ImageNet Detection (ILSVRC 2014)	MS-COCO (2014)
Number of classes	20	**200**	80
Number of images (train + val)	~20k	**~470k**	~120k
Mean objects per image	2.4	1.1	**7.2**

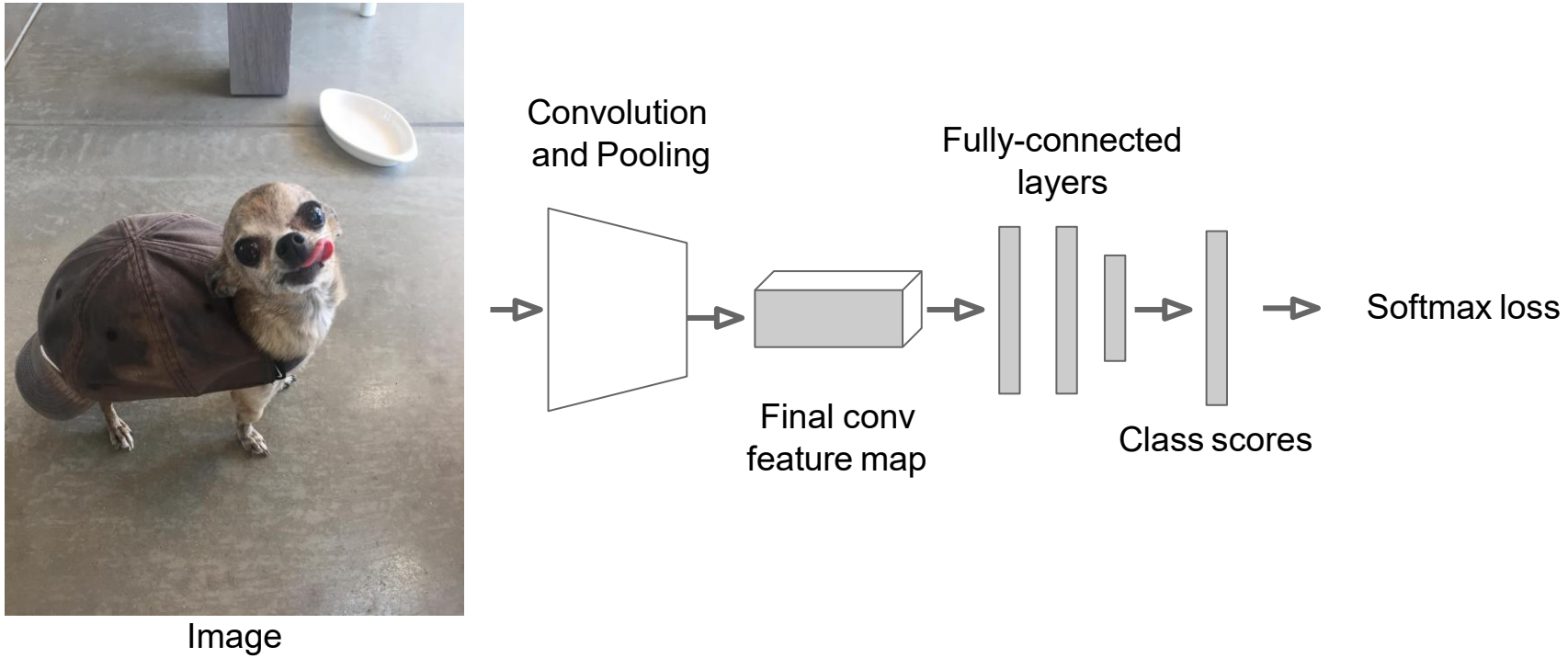
Object detection

Questions:

- What is the purpose of the detection task?
- What do we compute for detection at the output?
- Does it differ from classification?
- If so, how?

Simple Recipe for Object Detection

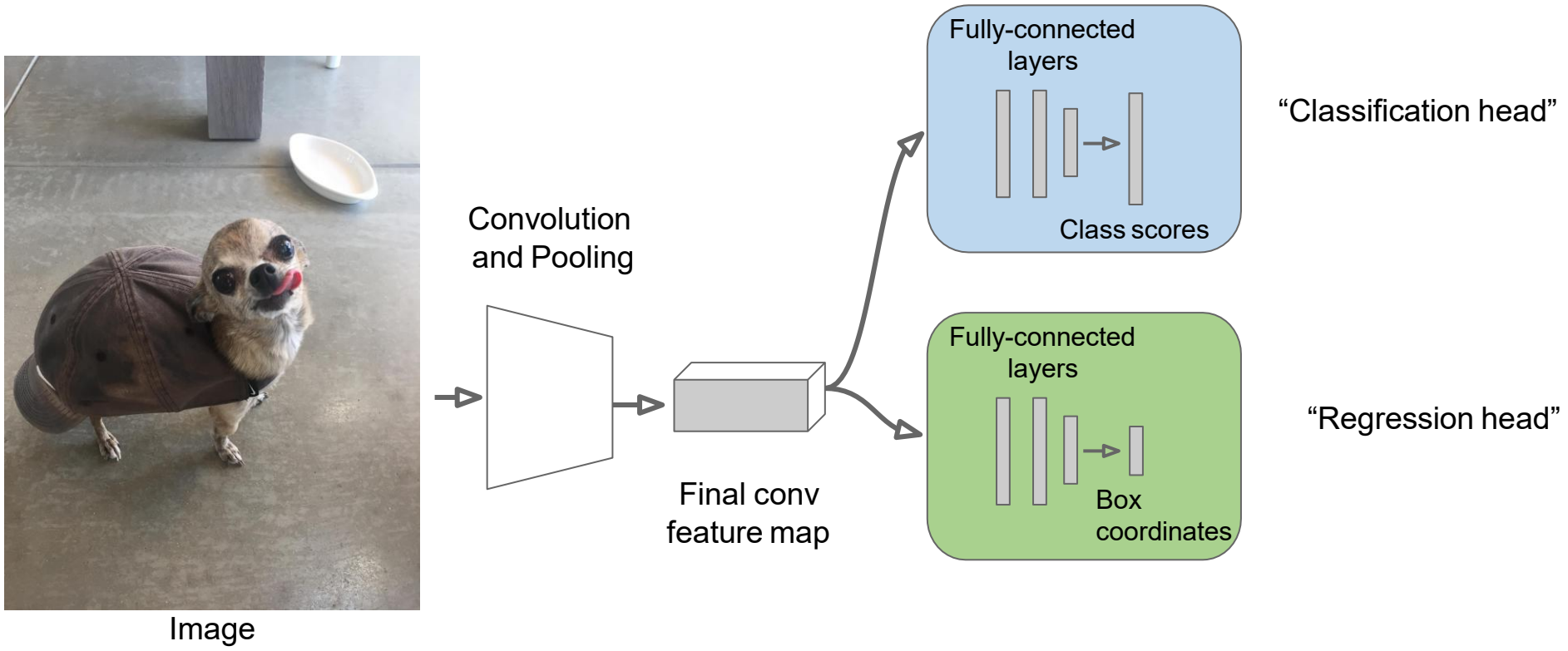
Step 1: Train (or download) a classification model (AlexNet, VGG, GoogLeNet)



Problem: Is there a **turtle** in this picture? If yes, localize!

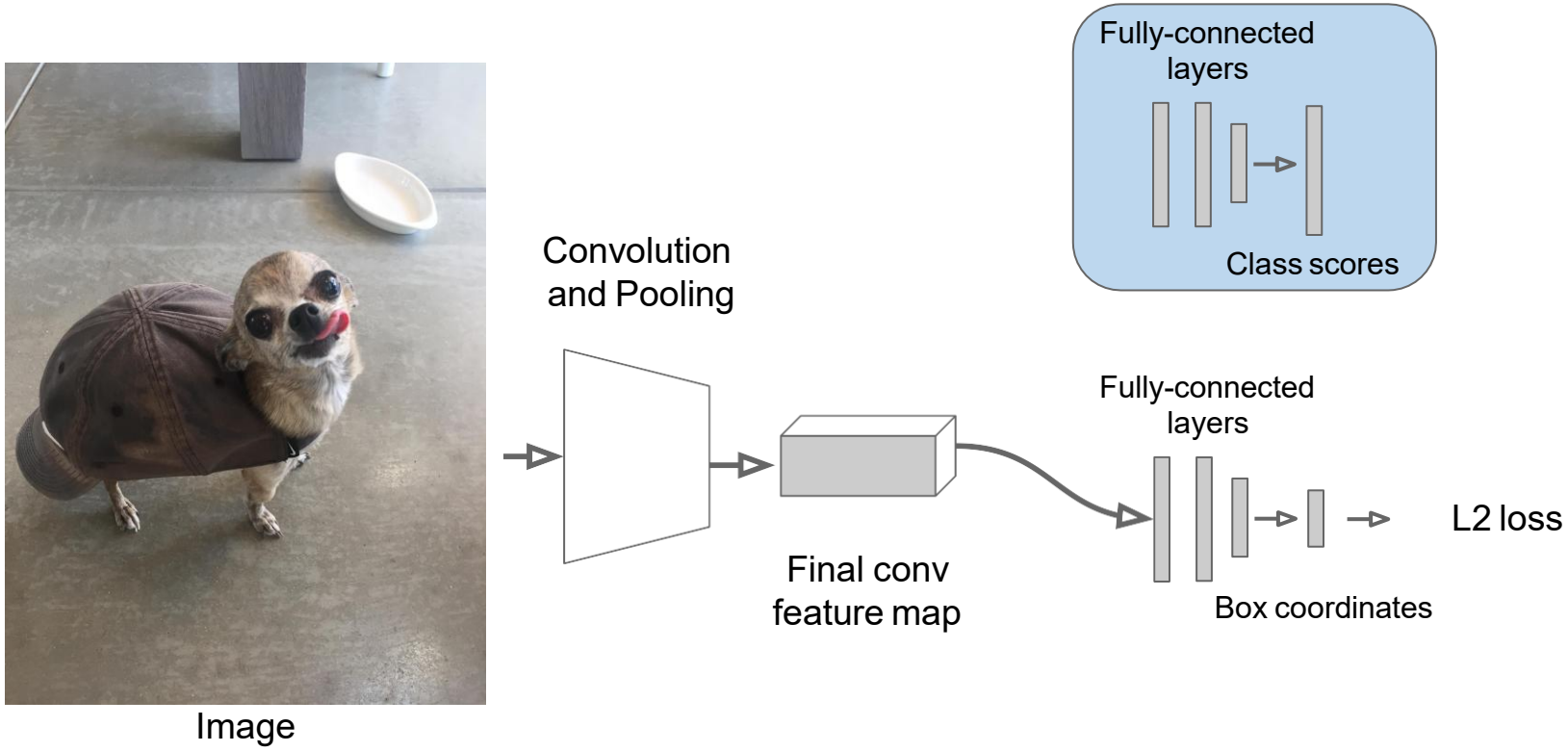
Simple Recipe for Object Detection

Step 2: Attach new fully-connected "regression head" to the network.



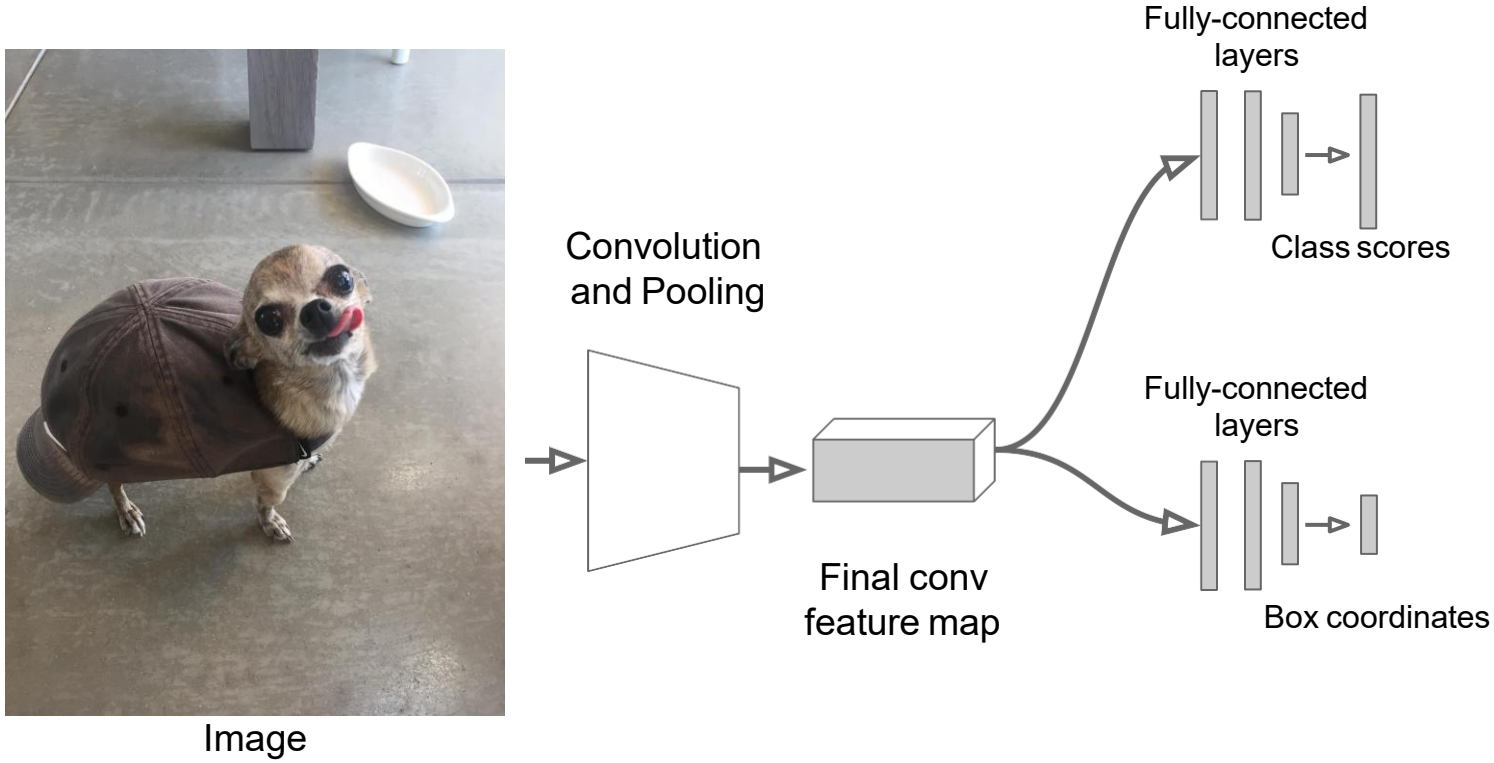
Simple Recipe for Object Detection

Step 3: Train the regression head only using L2 loss



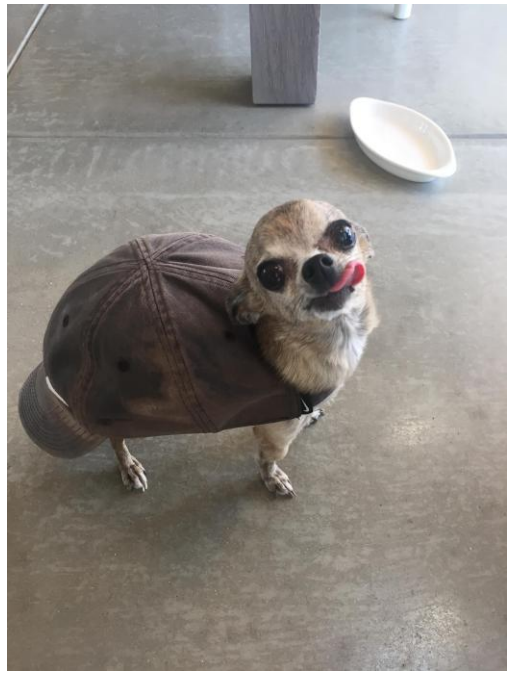
Simple Recipe for Object Detection

Step 4: At test time use both heads.

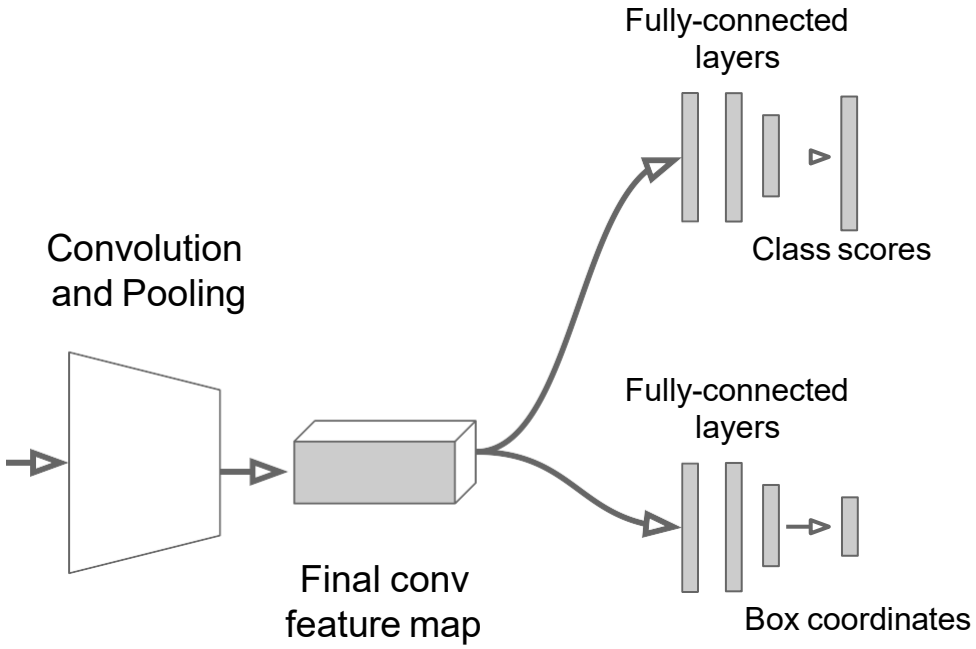


“Per-class" or “class agnostic" regression

Assume classification over C classes:



Image



Classification head:
C numbers
(one per class)

Class agnostic:
4 numbers
(one box)

Class specific:
C x 4 numbers
(one box per class)

Questions?

Object Detection

Lecture 13

CNN based approach
RCNN

Detection as Regression?



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

= 16 numbers

Detection as Regression?



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

= 8 numbers

Detection as Regression?



CAT, (x, y, w, h)
CAT, (x, y, w, h)
....
CAT (x, y, w, h)

= many numbers

Need variable sized outputs

Detection as Classification



CAT? NO

DOG? NO

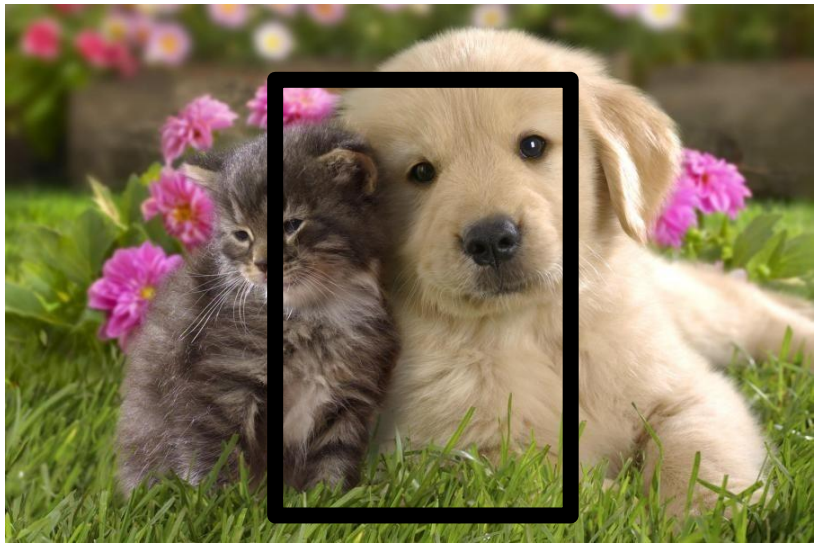
Detection as Classification



CAT? YES!

DOG? NO

Detection as Classification



CAT? NO

DOG? NO

Remember sliding window?

Detection as Classification

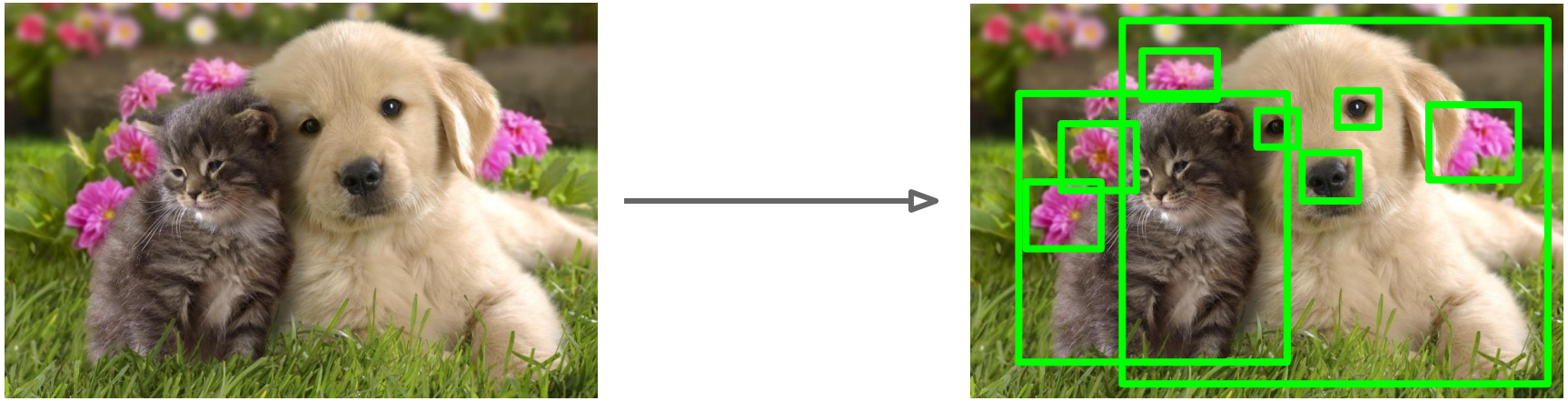
Problem:

- Need to test many positions and scales
- Use a computationally demanding classifier (CNN)
- Search at different scales
- Search at different positions

Solution: Only look at a tiny subset of possible positions

Region Proposals

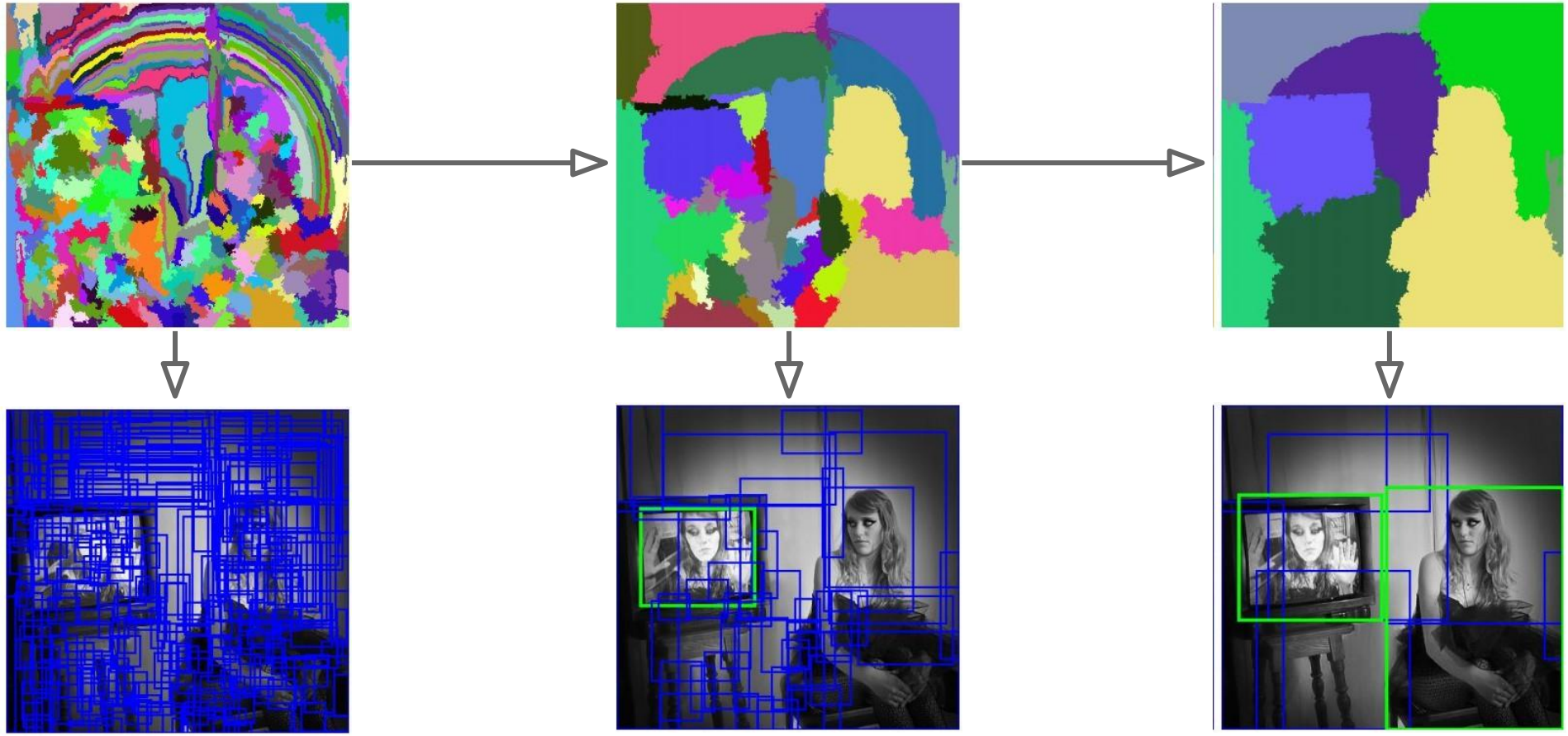
- Find “blobby” image regions that are likely to contain objects
- “Class-agnostic” object detector
- Look for “blob-like” regions



Region Proposals: Selective Search

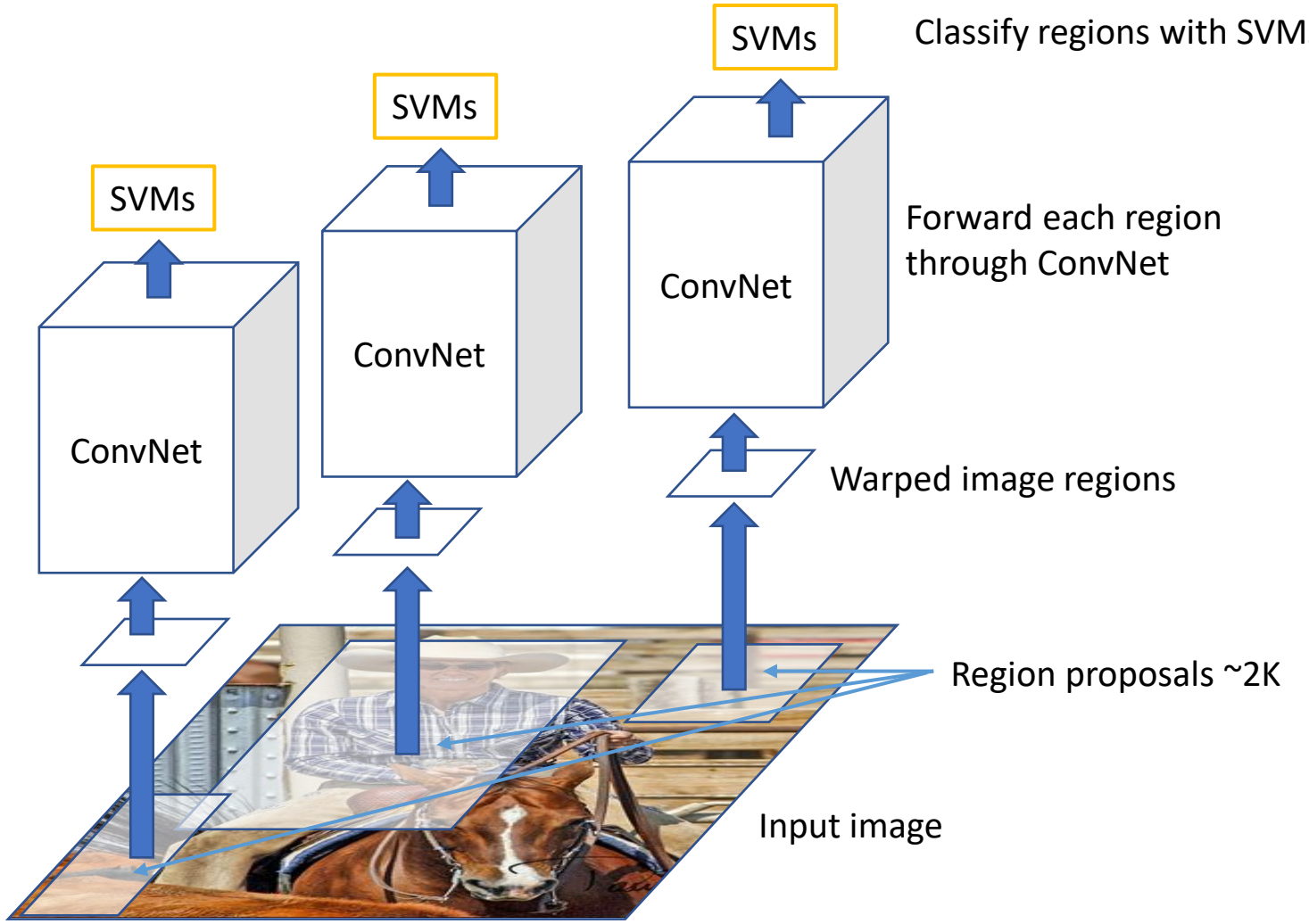
Bottom-up segmentation, merging regions at multiple scales

Convert regions to boxes



Uijlings et al, “Selective Search for Object Recognition”, IJCV 2013

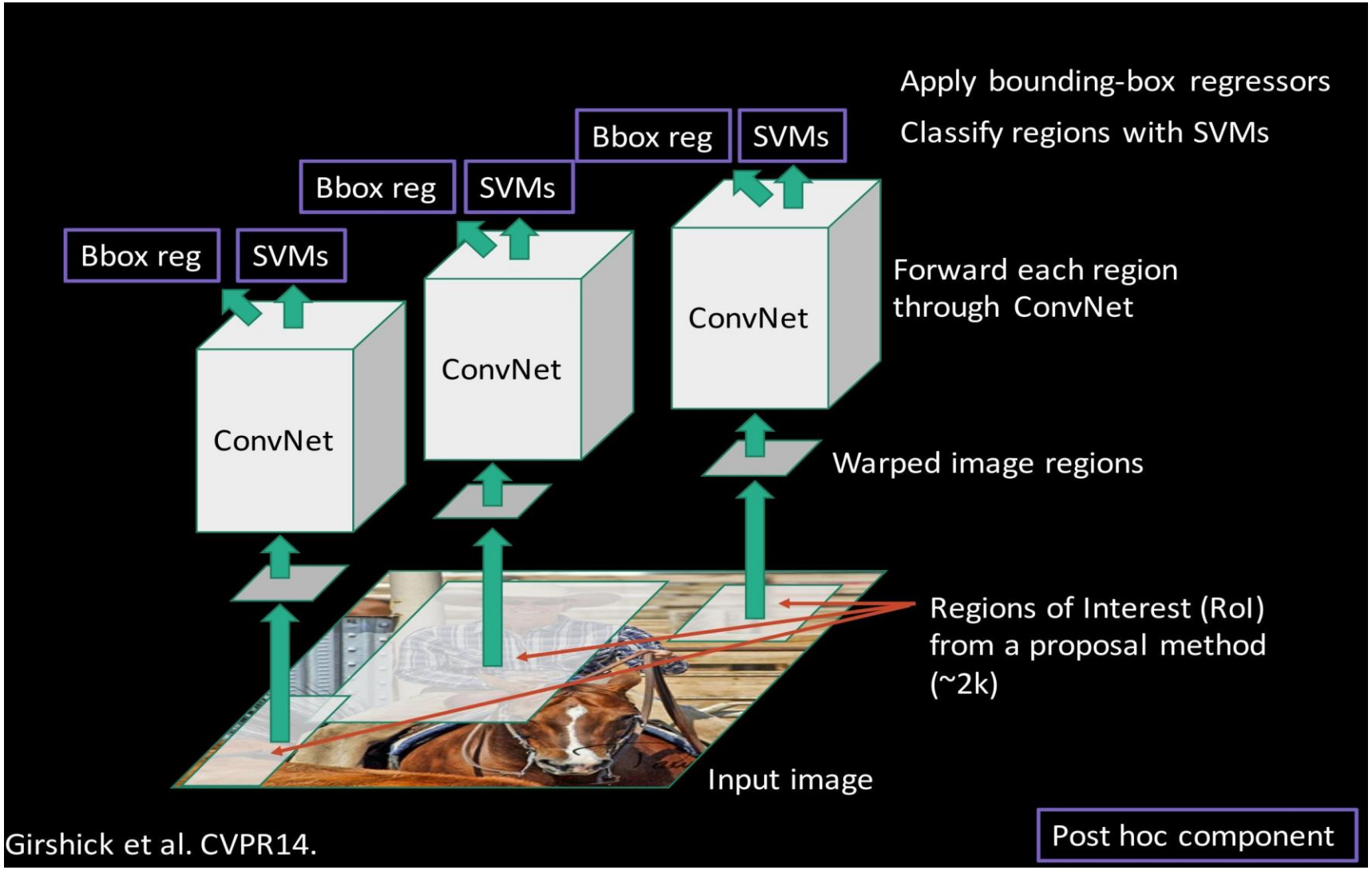
R-CNN: Region proposals + CNN features



Source: R. Girshick

R. Girshick, J. Donahue, T. Darrell, and J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, CVPR 2014.

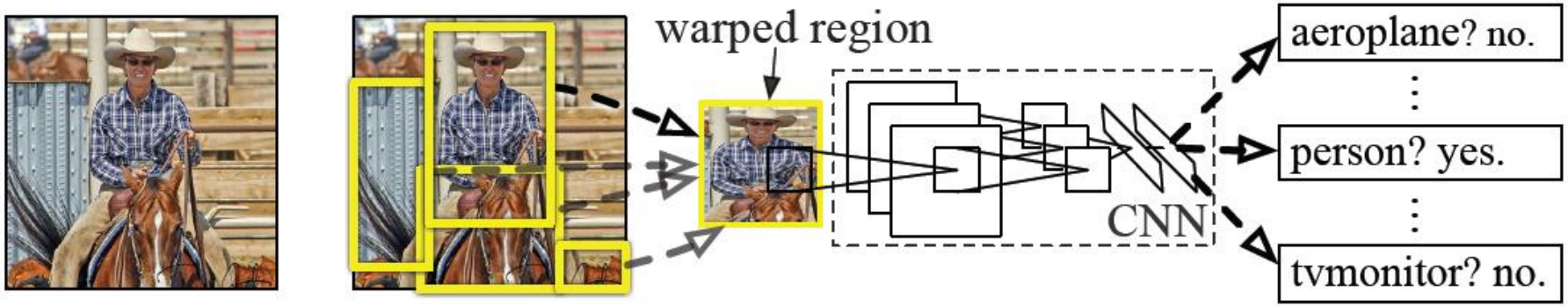
R-CNN: Region proposals + CNN features



Source: R. Girshick

R. Girshick, J. Donahue, T. Darrell, and J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, CVPR 2014.

R-CNN details

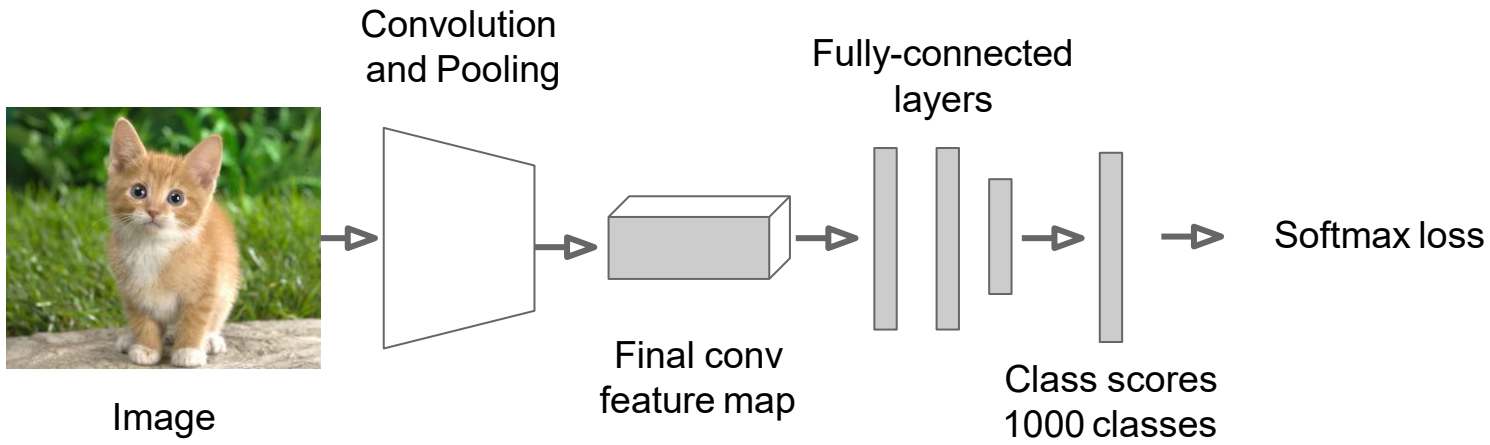


- **Regions**: uses ~2000 Selective Search proposals
- **Network**: uses AlexNet *pre-trained* on ImageNet (1000 classes), *fine-tuned* on PASCAL (21 classes)
- **Final detector**:
  - first warp proposal regions,
  - then extract fc7 network activations (4096 dimensions),
  - Finally, classify with linear SVM
- **Bounding box regression** is also used to refine box locations

R. Girshick, J. Donahue, T. Darrell, and J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, CVPR 2014.

R-CNN Training (initialization)

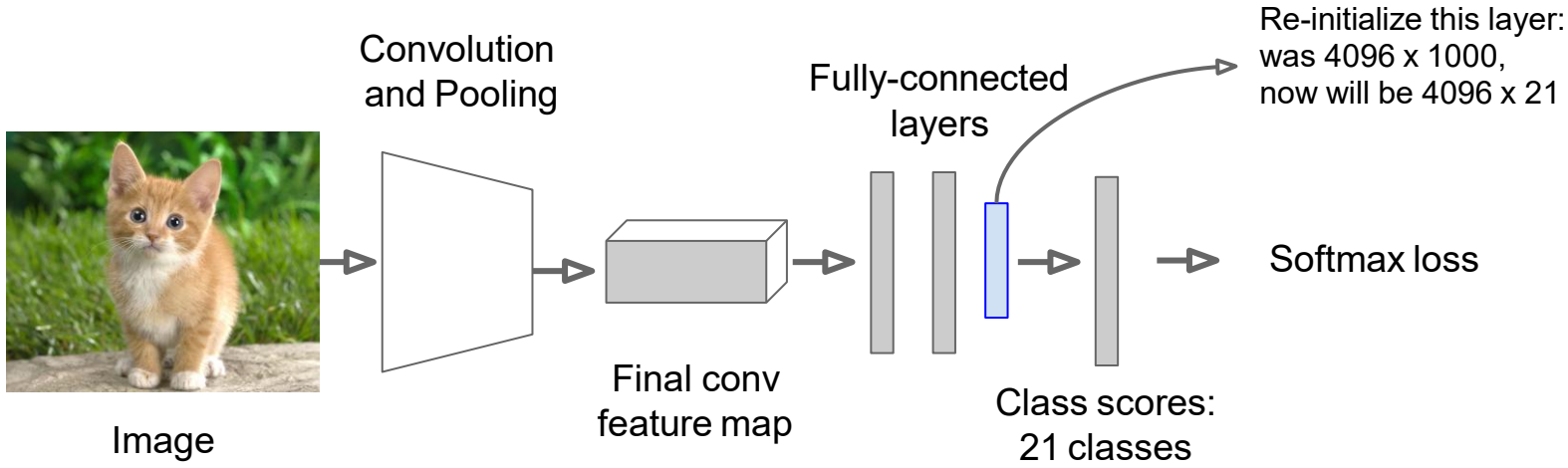
Step 1: Train (or download) a classification model for ImageNet (AlexNet)



R-CNN Training (Fine-tuning)

Step 2: "**Fine-tune**" model for detection

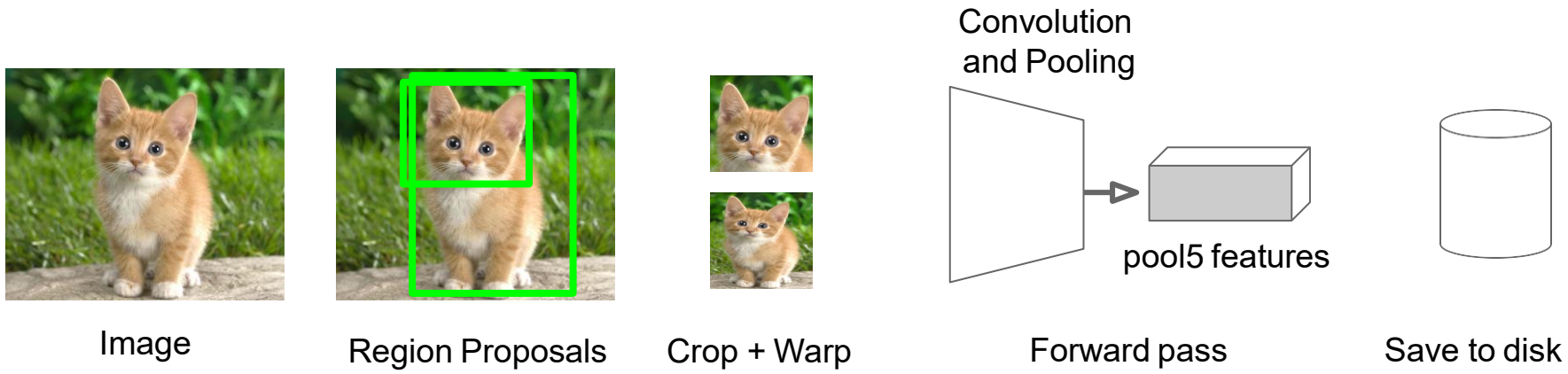
- Instead of 1000 ImageNet classes, want 20 object classes + background
- Throw away final fully-connected layer, reinitialize from scratch
- Keep training model using positive / negative regions from detection images



R-CNN Training (feature extraction)

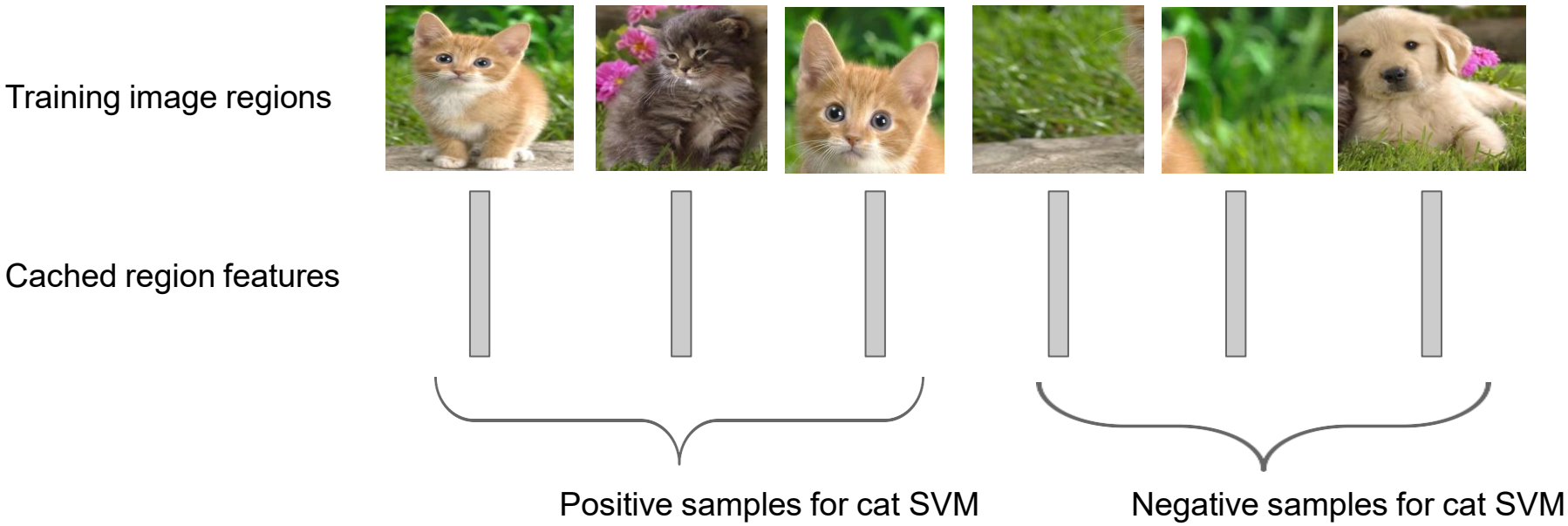
Step 3: Extract features

- Extract region proposals for all images
- For each region: warp to CNN input size, run forward through CNN, save pool5 features to disk
- Have a big hard drive: features are ~200GB for PASCAL dataset!



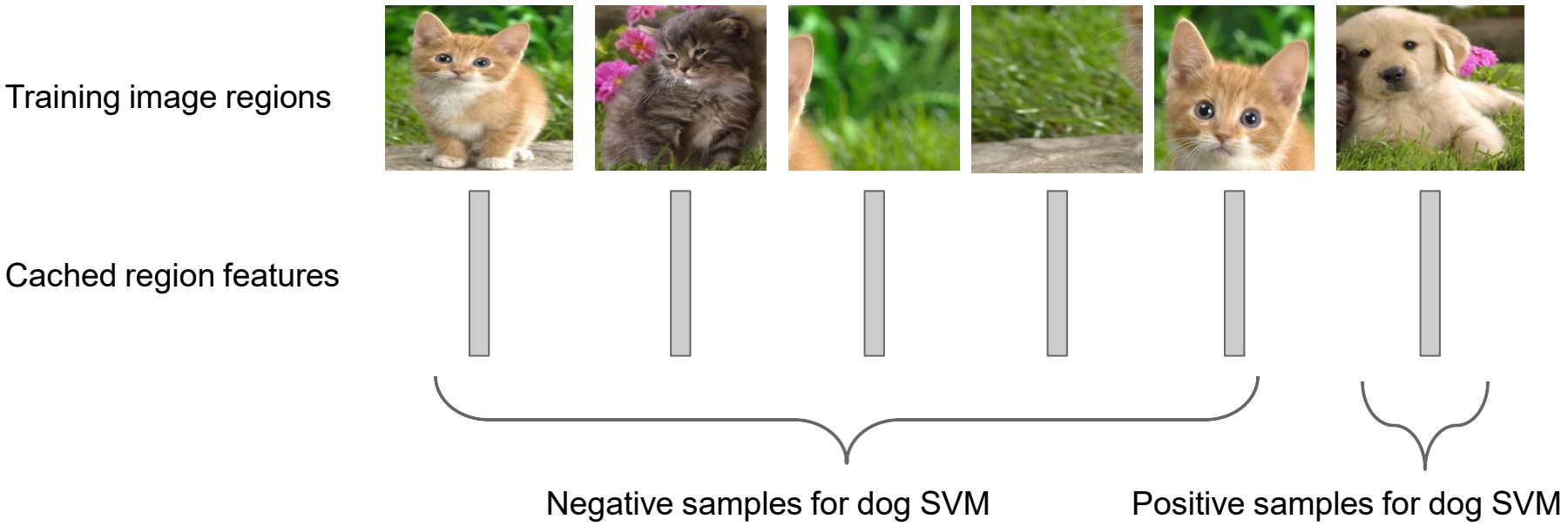
R-CNN Training (train classifier)

Step 4: Train one binary SVM per class to classify region features



R-CNN Training (train classifier)

Step 4: Train one binary SVM per class to classify region features



R-CNN Training (bounding box regression/prediction)

Step 5 (bbox regression): **For each class**, train a linear **regression** model to map from cached features to offsets to GT boxes to make up for “slightly wrong” proposals

Training image regions:







Cached region features







Regression **targets:**
(dx, dy, dw, dh)
Normalized coordinates

(0, 0, 0, 0)
Proposal is good

(.25, 0, 0, 0)
Proposal too far to left

(0, 0, -0.125, 0)
Proposal too wide

Questions?

How can CV help?

- https://youtu.be/w8sAnw9ecm8?si=vZT4lhMdb2L1BS-J

Questions?

SPI

- Student evaluation of course instruction (SPI)
  - Will be available on Webcourse
  - Will be available for around 15 days
- Online questionnaire includes
  - ONLY eleven (11) questions
  - 9 multiple choice and 2 open comment questions

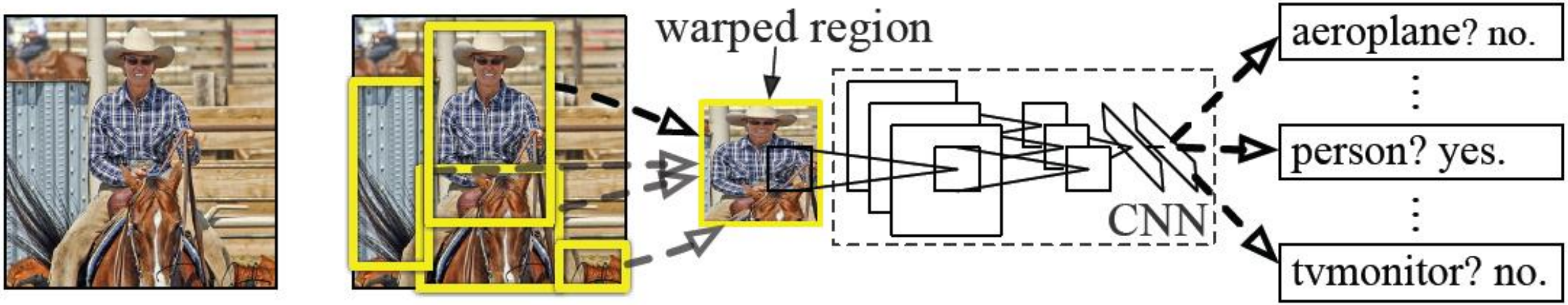
Object Detection

Lecture 13

CNN based approach
Fast-RCNN

Issue #1 with R-CNN

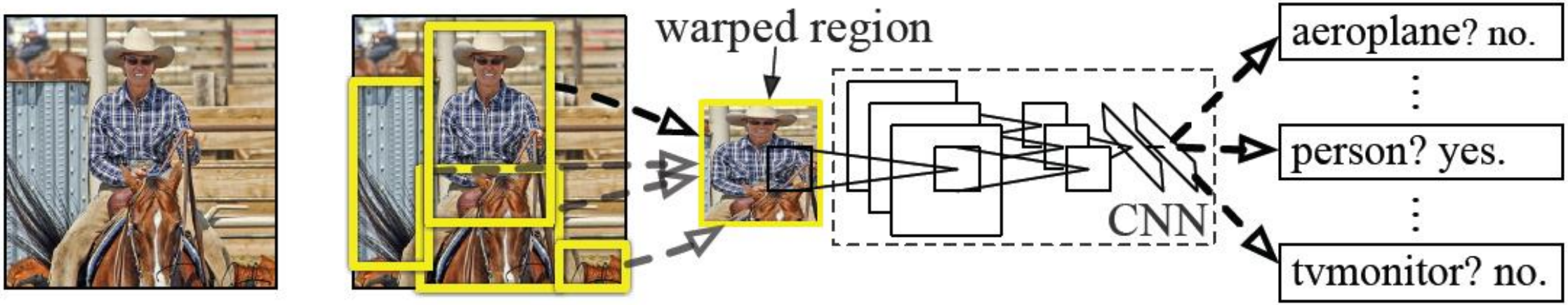
- Slow in run-time
  - Multiple forward passes for each proposal
  - There are thousands of proposals



- Solution
  - Single forward pass for each image?

Issue #2 with R-CNN

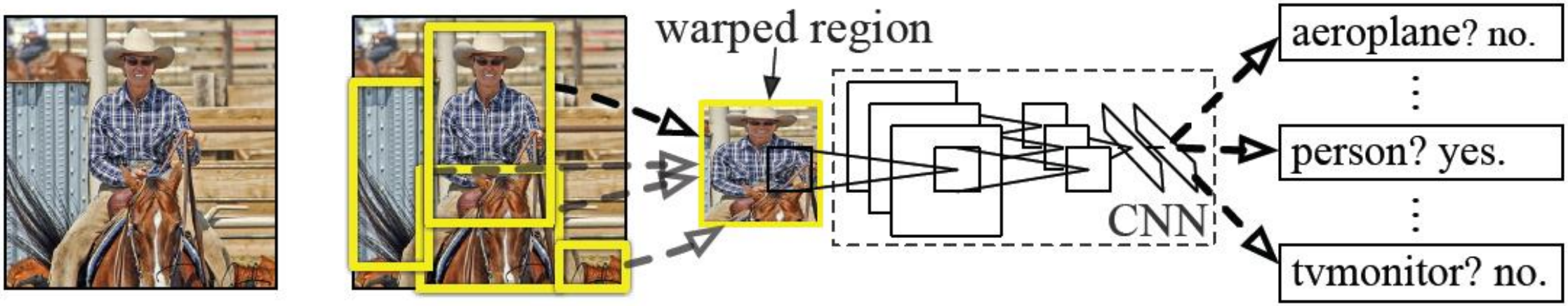
- Separate classifier training
  - CNN feature extractor is not trained with classifier and regressor



- Solution
  - End-to-end training?

Issue #3 with R-CNN

- Complex training pipeline
  - Proposals
  - Feature extraction
  - Classification

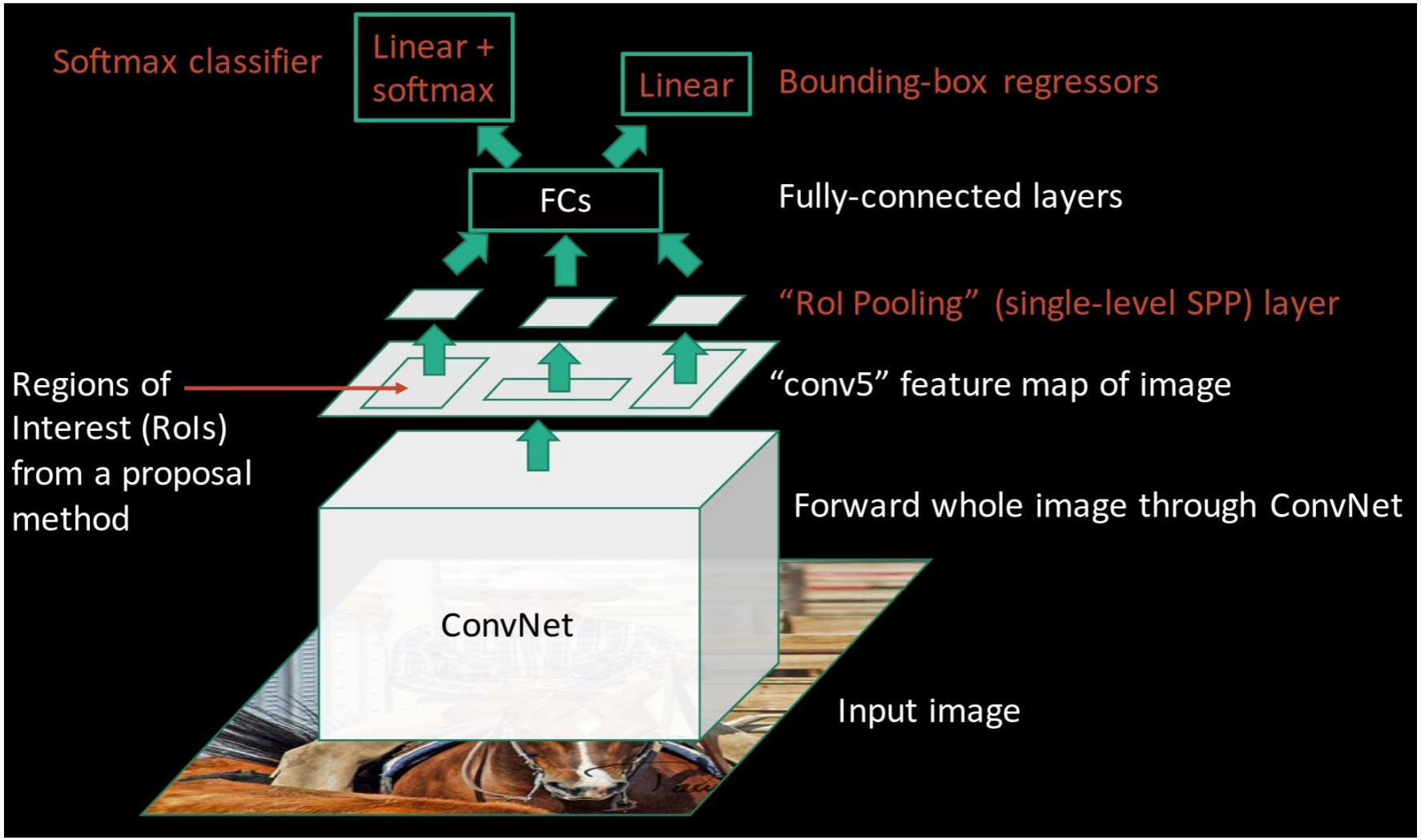


- Solution
  - Single forward pass for each image?

Solution

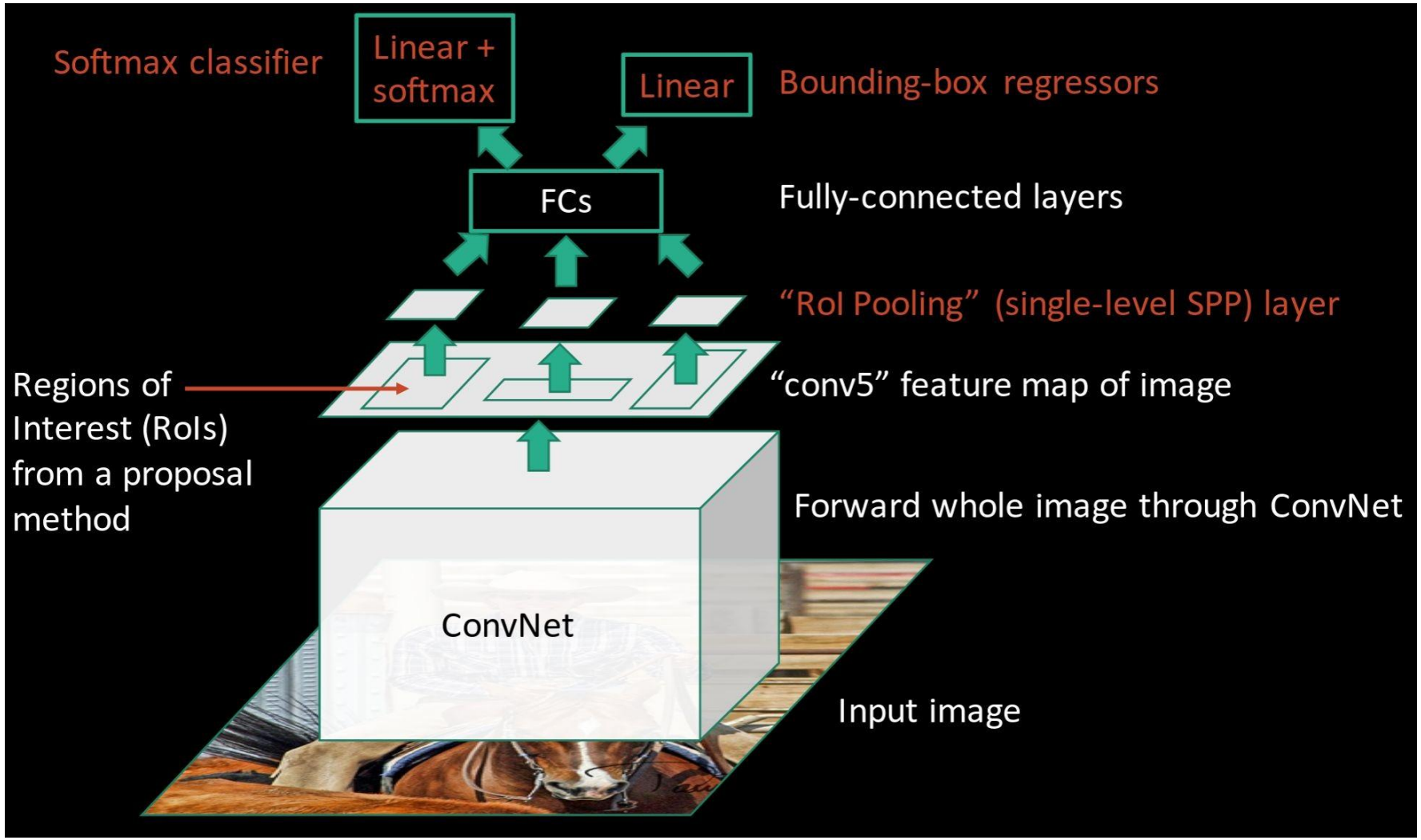
- Fast R-CNN
  - Single forward pass for each image
  - No separate classifier
  - End-to-end training

Fast R-CNN



Slide credit: Ross Girschick, Girschick, "Fast R-CNN", ICCV 2015

Fast R-CNN

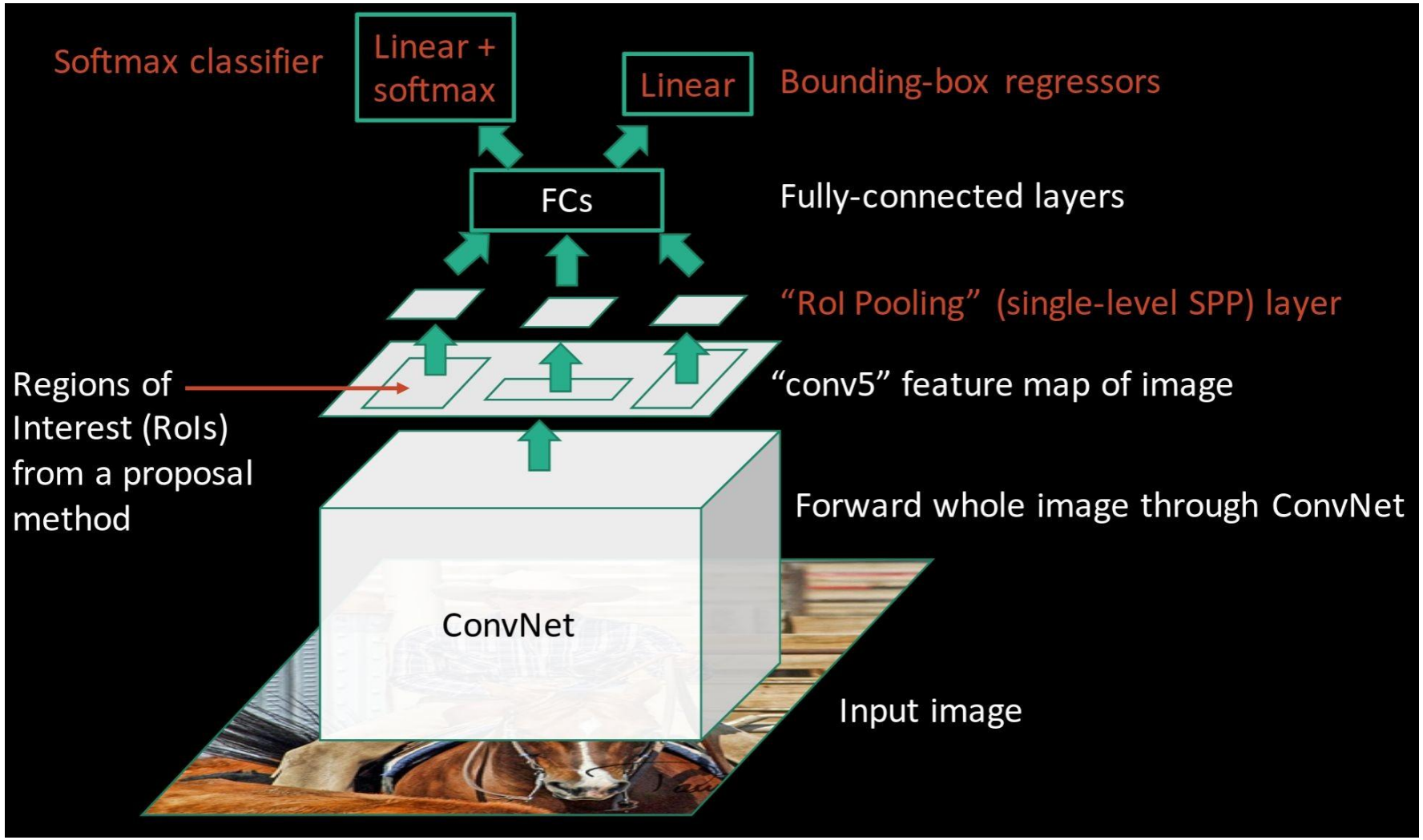


Solution #1: Share computation of convolutional layers between proposals for an image

Slide credit: Ross Girschick, Girschick, "Fast R-CNN", ICCV 2015

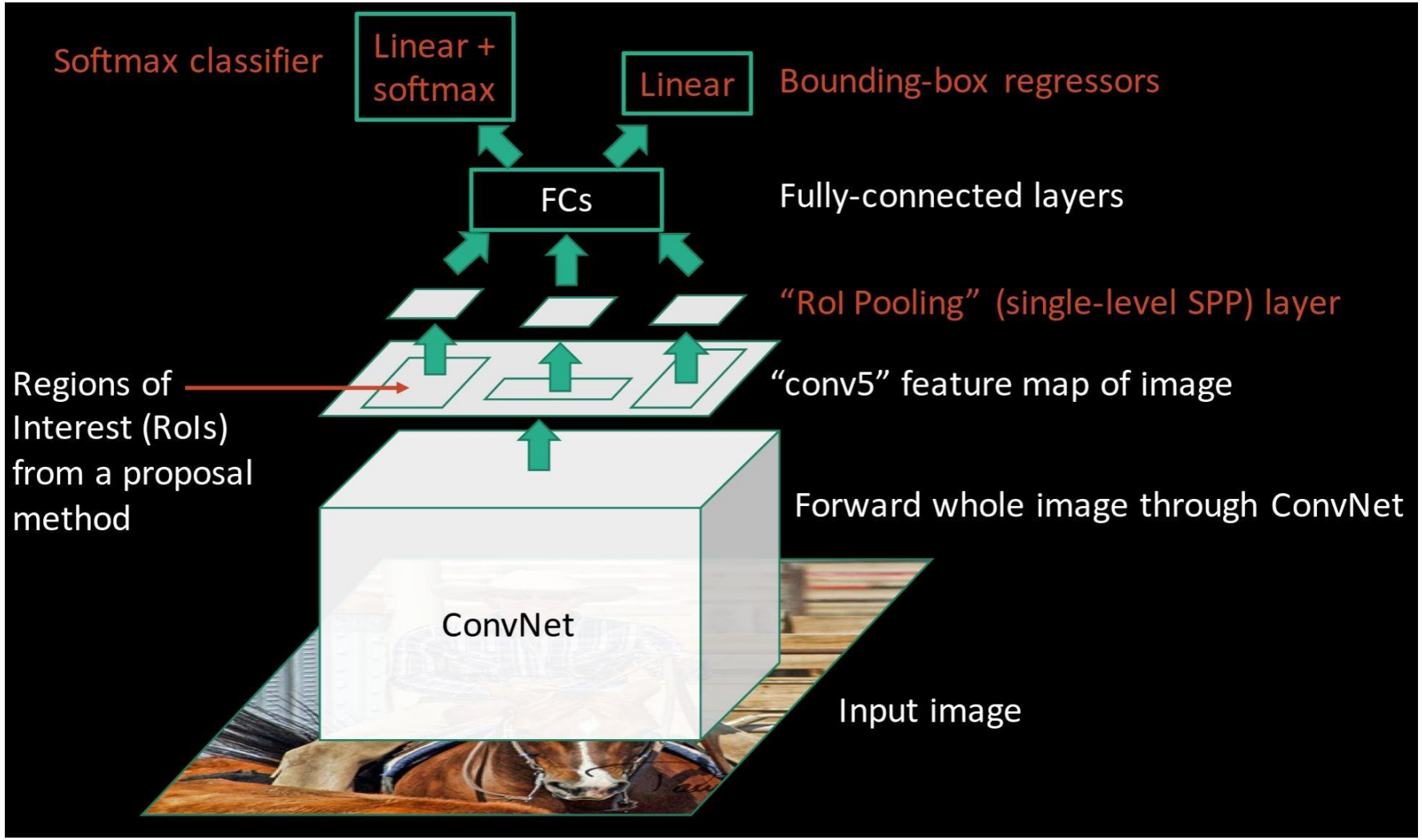
Fast R-CNN

Solution #2:
Train softmax classifier jointly



Slide credit: Ross Girschick, Girschick, "Fast R-CNN", ICCV 2015

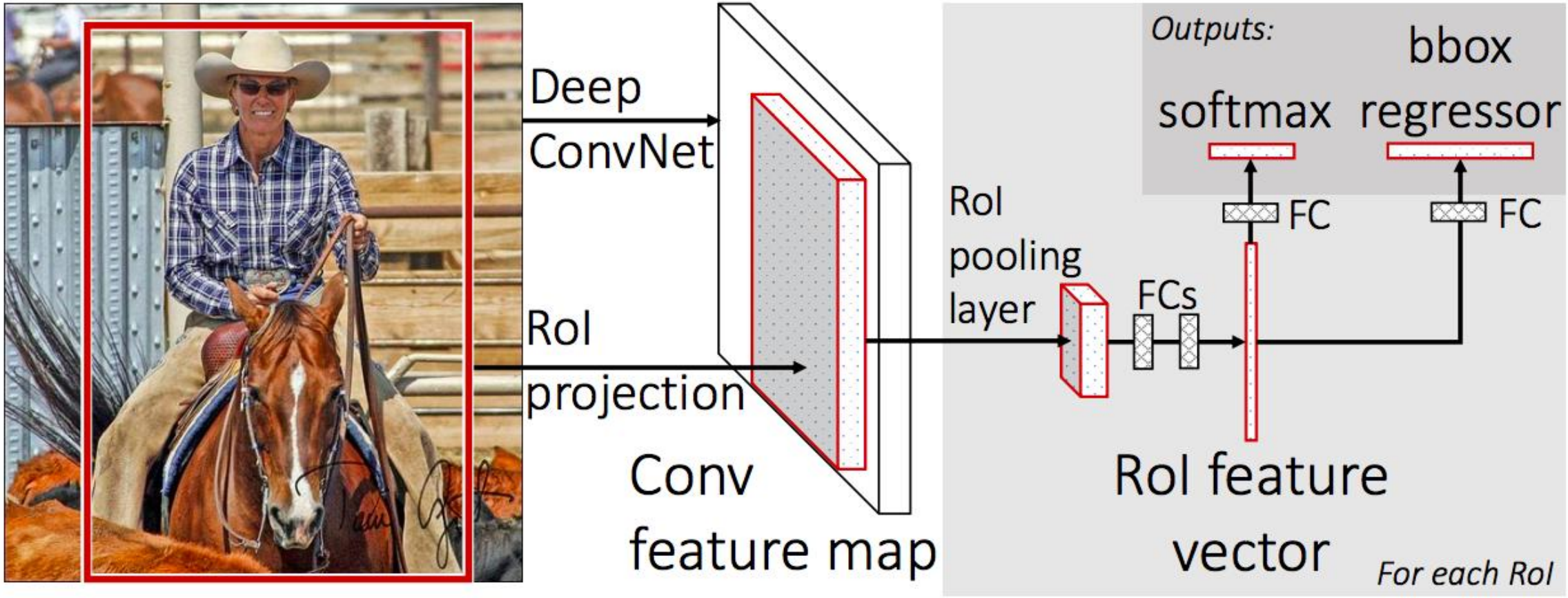
Fast R-CNN



Solution #3:
Just train the whole system end-to-end all at once!

Slide credit: Ross Girschick, Girschick, "Fast R-CNN", ICCV 2015

Fast R-CNN: Another view

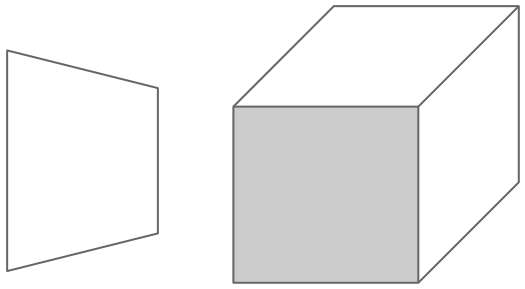


R. Girshick, Fast R-CNN, ICCV 2015

Fast R-CNN: Region of Interest Pooling

Convolution and Pooling

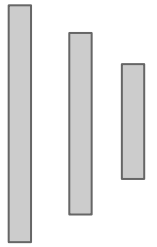




Hi-res input image: 3 x 800 x 600 with region proposal

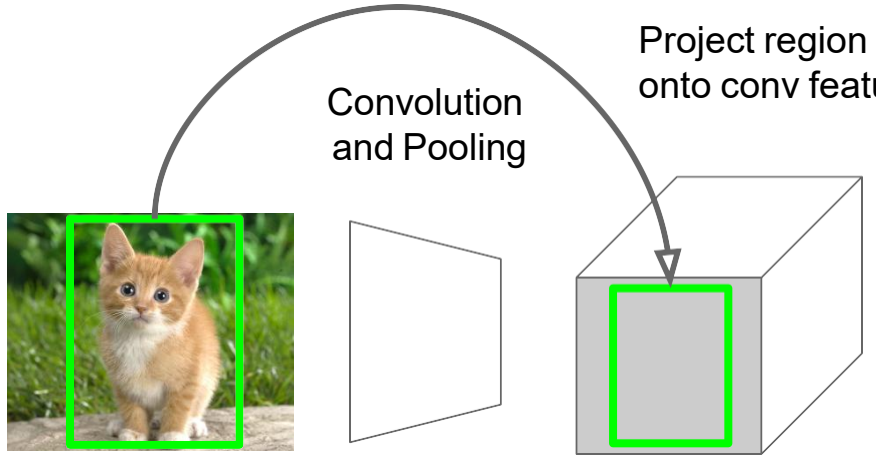
Hi-res conv features: C x H x W with region proposal

Fully-connected layers



Problem: Fully-connected layers expect low-res conv features: C x h x w

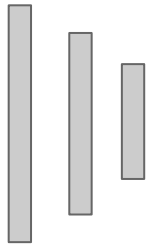
Fast R-CNN: Region of Interest Pooling



Hi-res input image:
3 x 800 x 600
with region
proposal

Hi-res conv features:
C x H x W
with region proposal

Fully-connected
layers



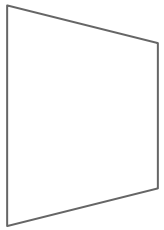
Problem: Fully-connected layers expect low-res conv features: C x h x w

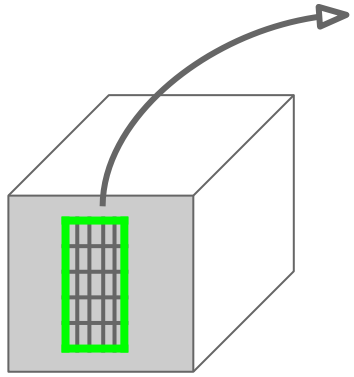
Fast R-CNN: Region of Interest Pooling



Hi-res input image:
3 x 800 x 600
with region
proposal

Convolution
and Pooling

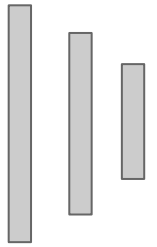




Hi-res conv features:
C x H x W
with region proposal

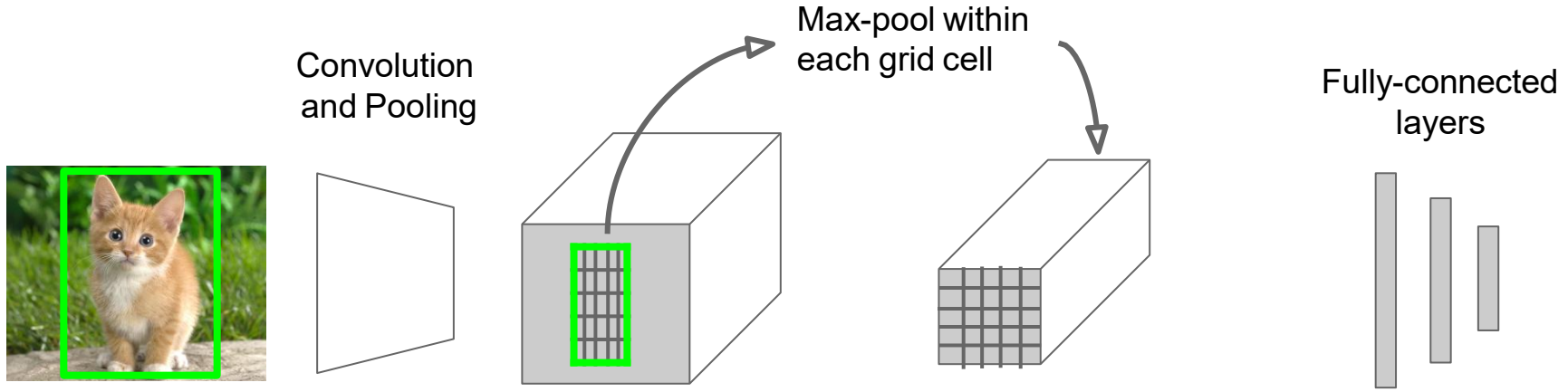
Divide projected
region into h x w grid

Fully-connected
layers

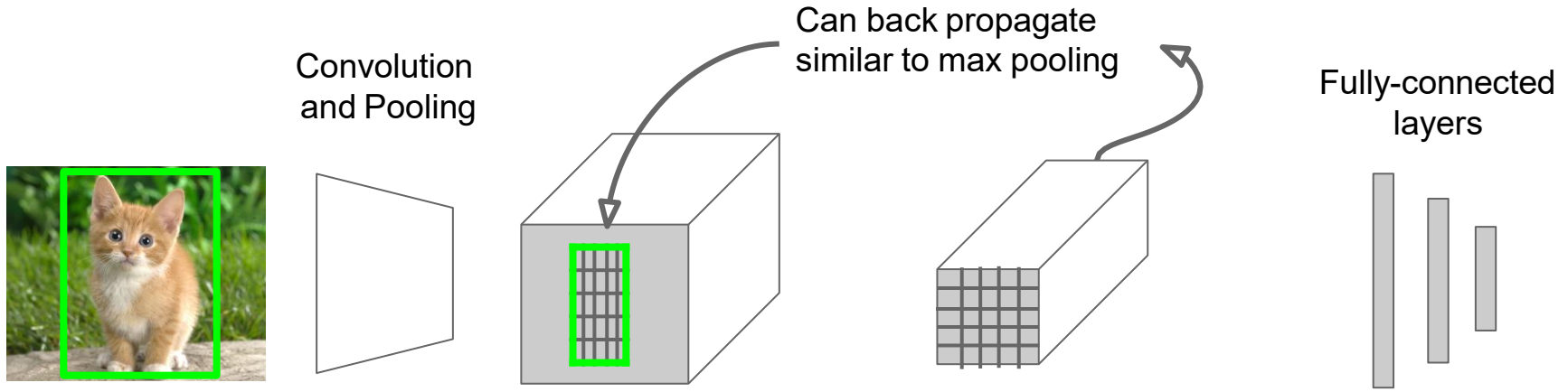


Problem: Fully-connected layers expect low-res conv features: C x h x w

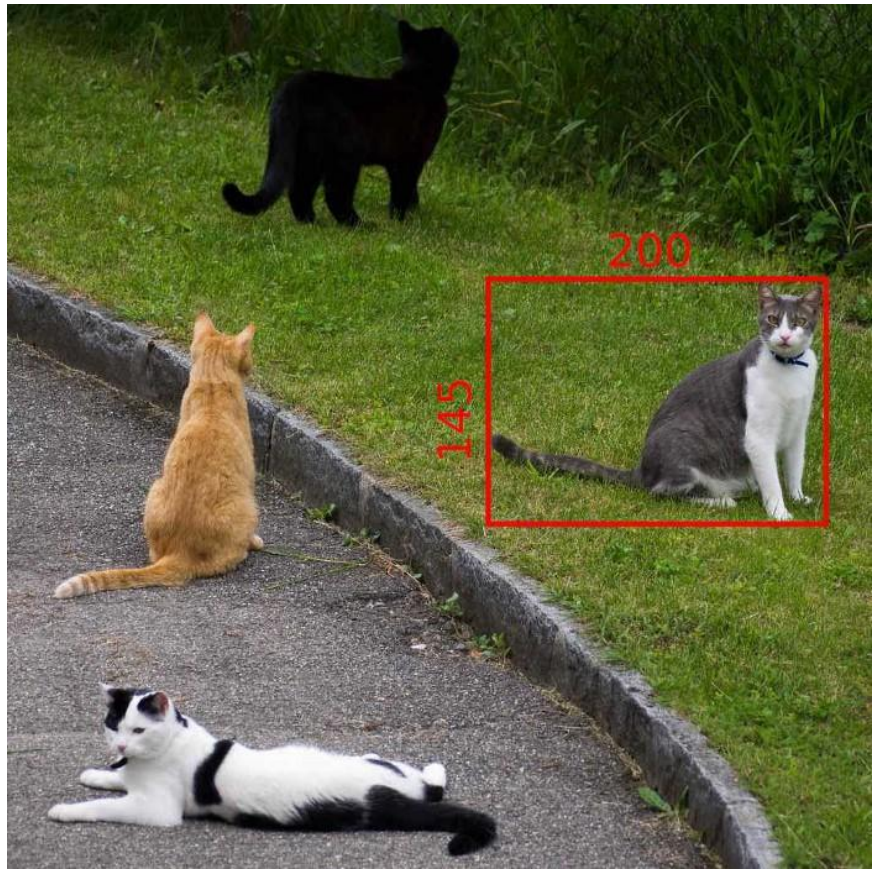
Fast R-CNN: Region of Interest Pooling

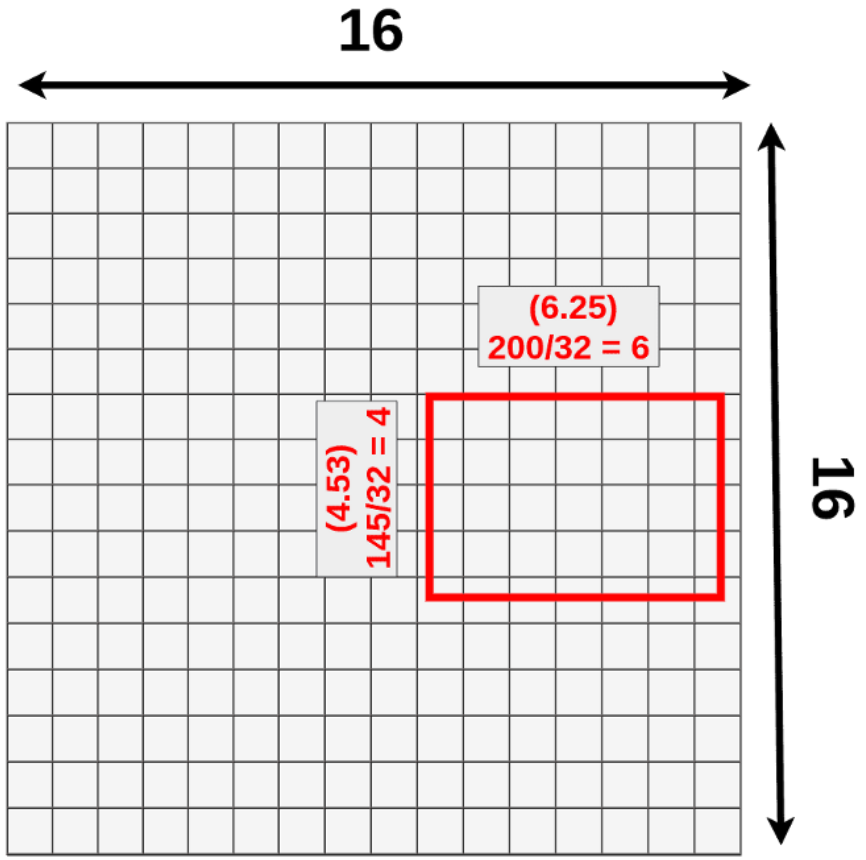


Fast R-CNN: Region of Interest Pooling



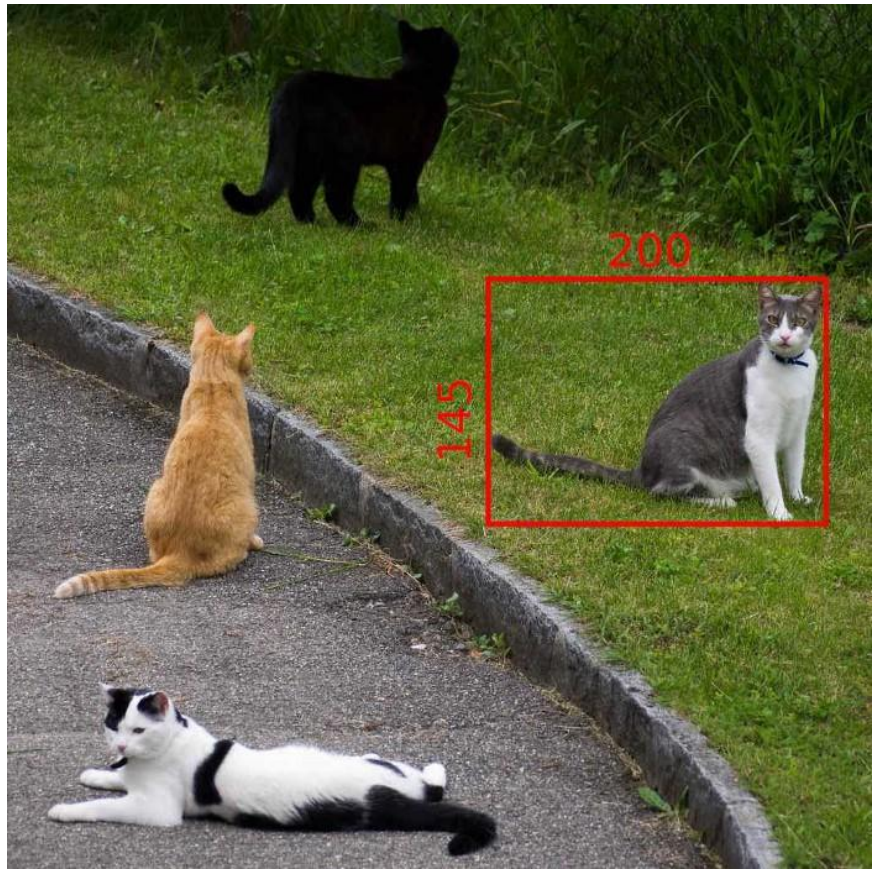
ROI-Pooling

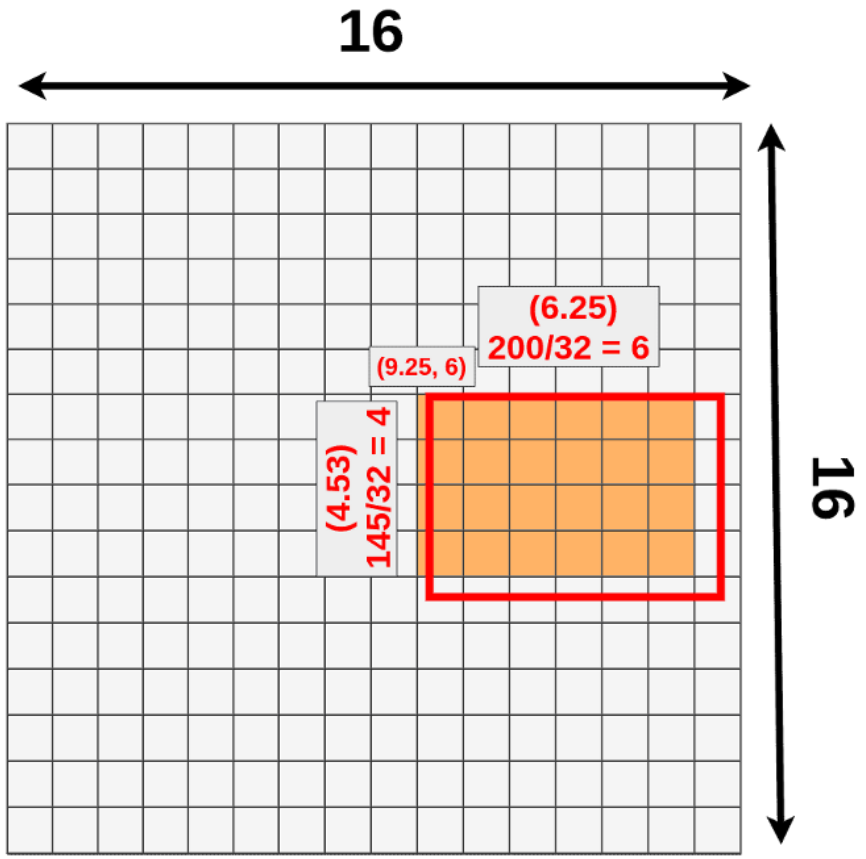




https://tinyurl.com/y6xpm24d

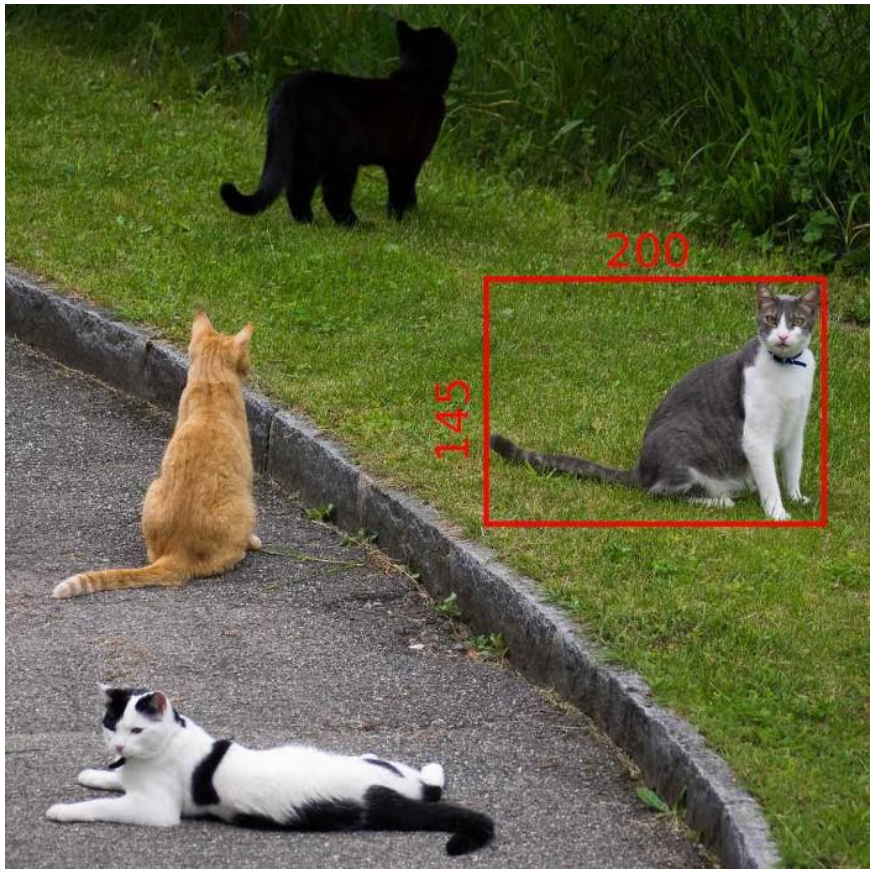
ROI-Pooling

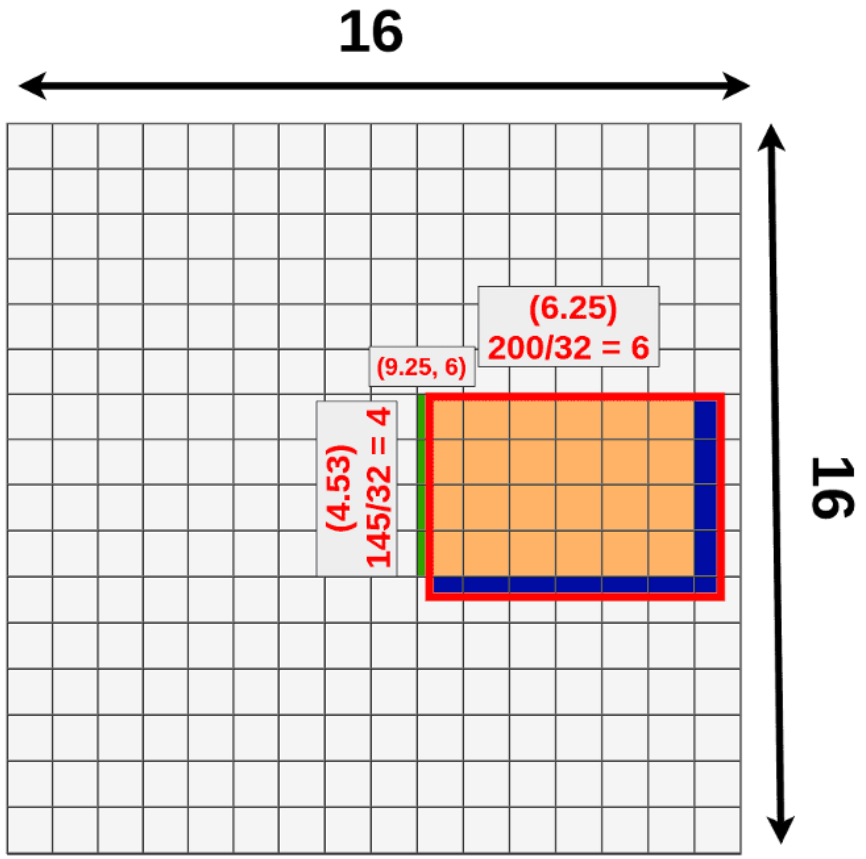




https://tinyurl.com/y6xpm24d

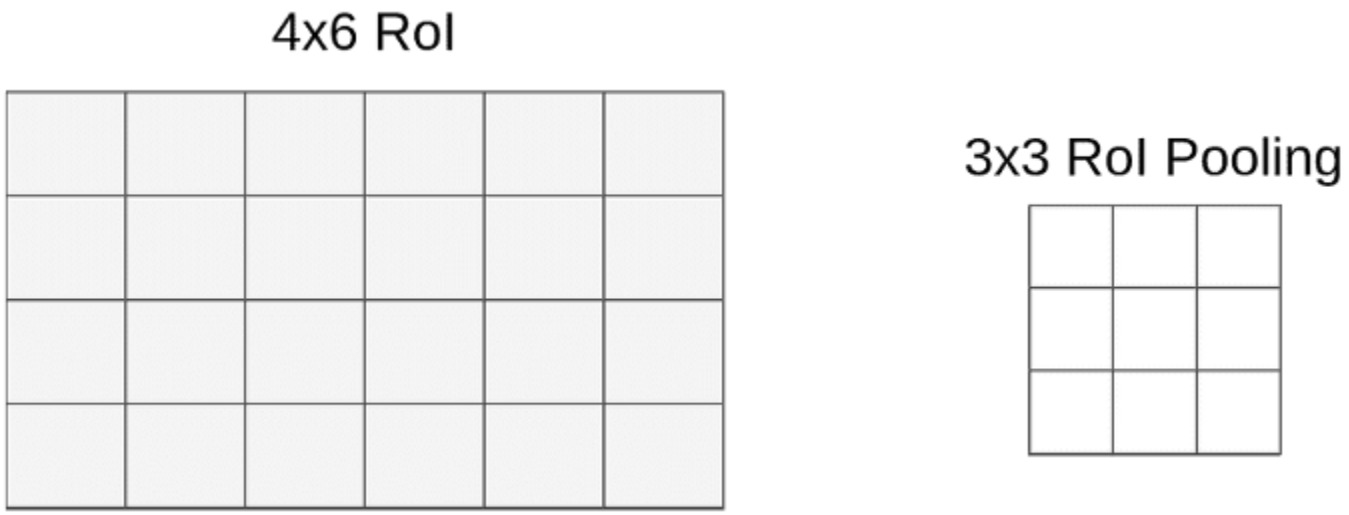
ROI-Pooling





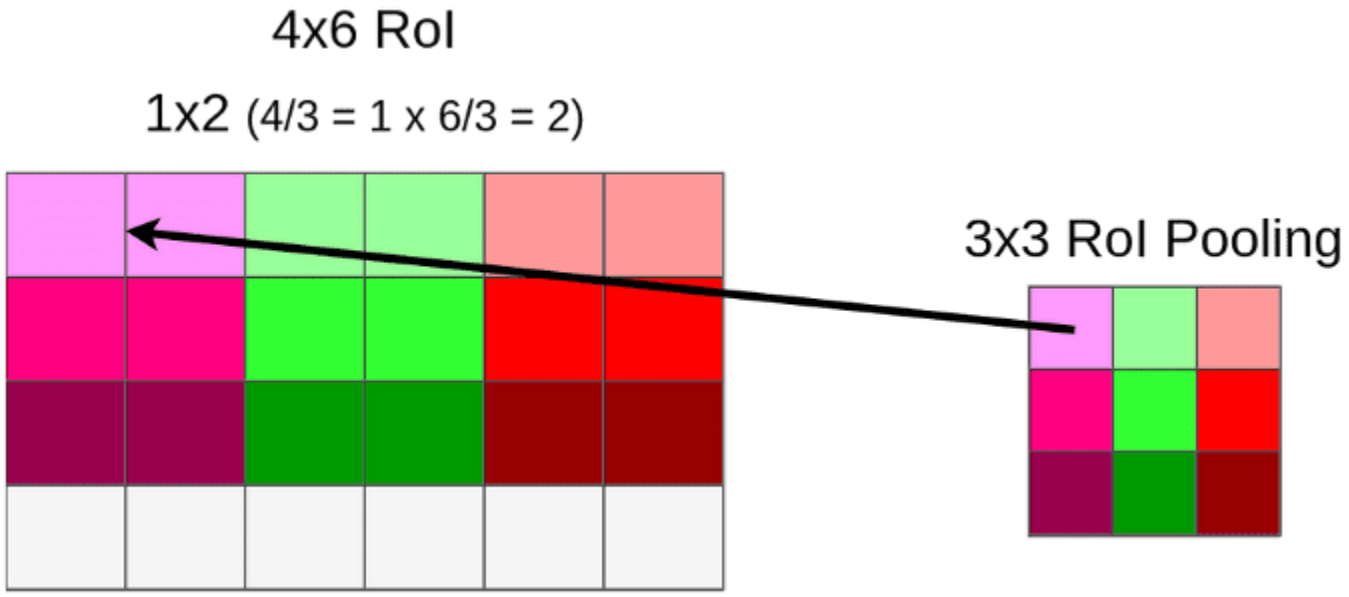
https://tinyurl.com/y6xpm24d

ROI-Pooling



https://tinyurl.com/y6xpm24d

ROI-Pooling



https://tinyurl.com/y6xpm24d

ROI-Pooling

4x6 RoI (Lost Data)

0.1	0.2	0.3	0.4	0.5	0.6
1	0.7	0.2	0.6	0.1	0.9
0.9	0.8	0.7	0.3	0.5	0.2
0.2	0.5	1	0.7	0.1	0.1

https://tinyurl.com/y6xpm24d

Fast R-CNN Results

Fast!

FASTER!

Better!

	R-CNN	**Fast R-CNN**
Training Time:	84 hours	**9.5 hours**
(Speedup)	1x	**8.8x**
Test time per image	47 seconds	**0.32 seconds**
(Speedup)	1x	**146x**
mAP (VOC 2007)	66.0	**66.9**

Using VGG-16 CNN on Pascal VOC 2007 dataset

Questions?

Object Detection

Lecture 13

CNN based approach
Faster-RCNN

Problem of Fast R-CNN?

- Test-time speeds don't include region proposals
- SOLUTION: ?

Problem of Fast R-CNN?

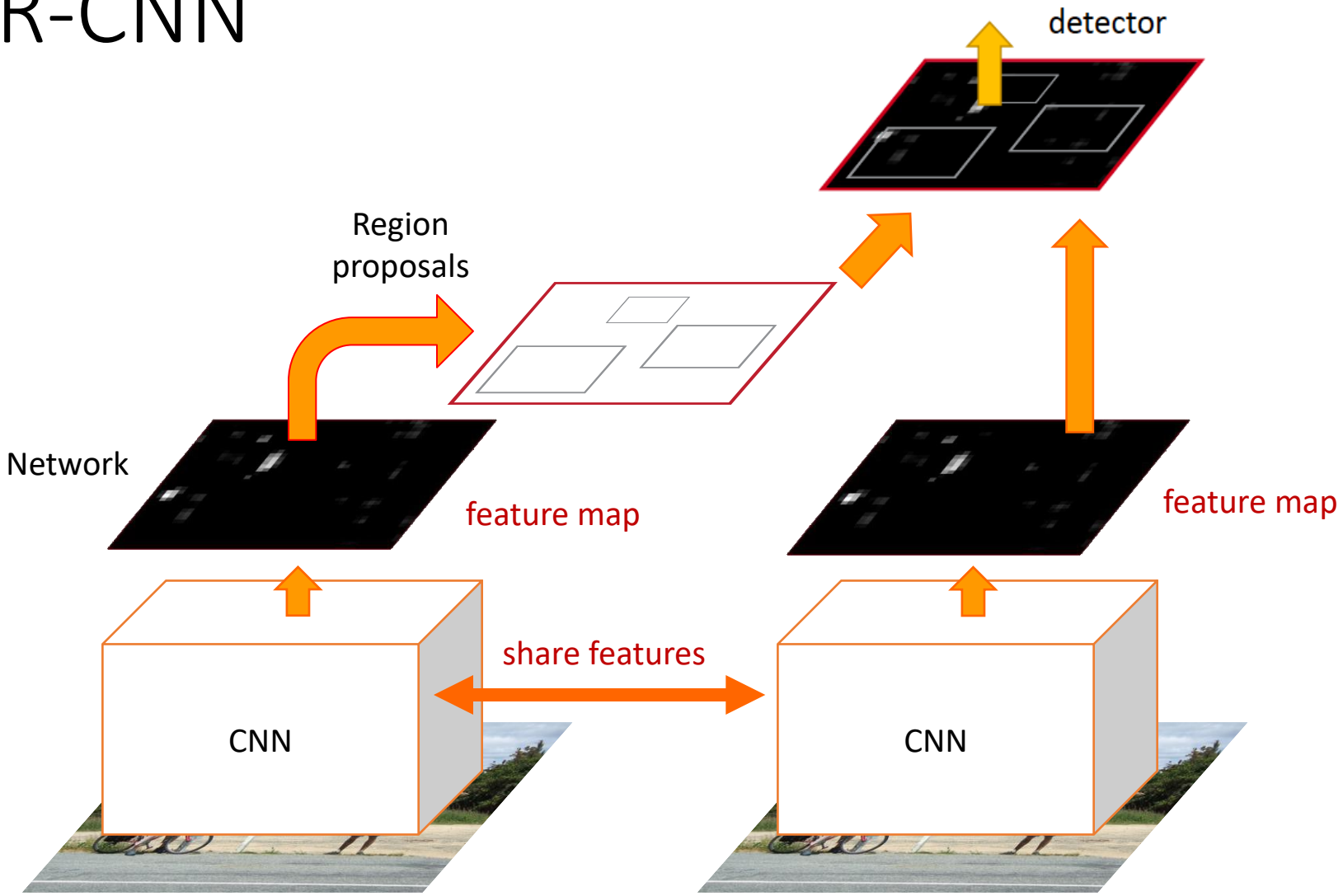
- Test-time speeds don't include region proposals
- SOLUTION: make the CNN do region proposals too!

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

Slide credit: Ross Girschick

Faster R-CNN

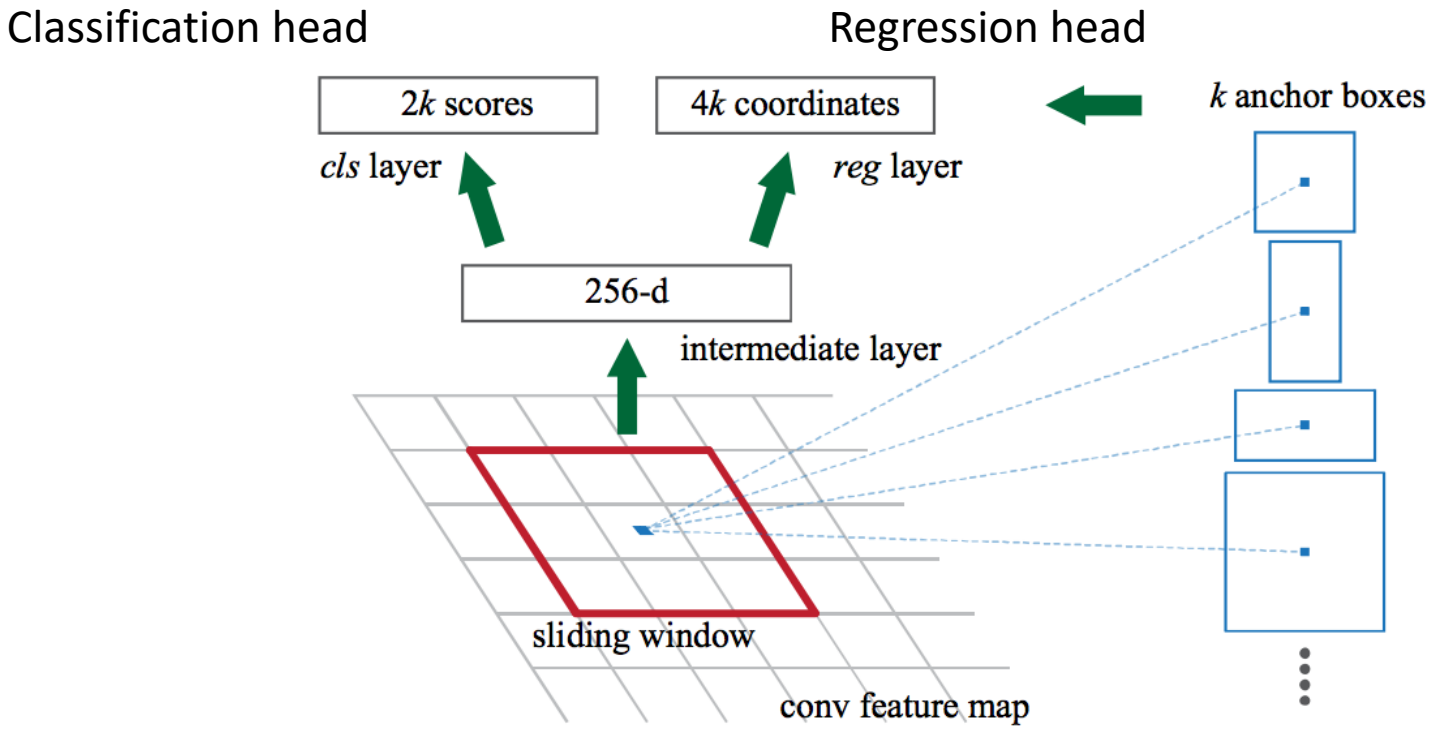
New: Region Proposal Network



S. Ren, K. He, R. Girshick, and J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, NIPS 2015

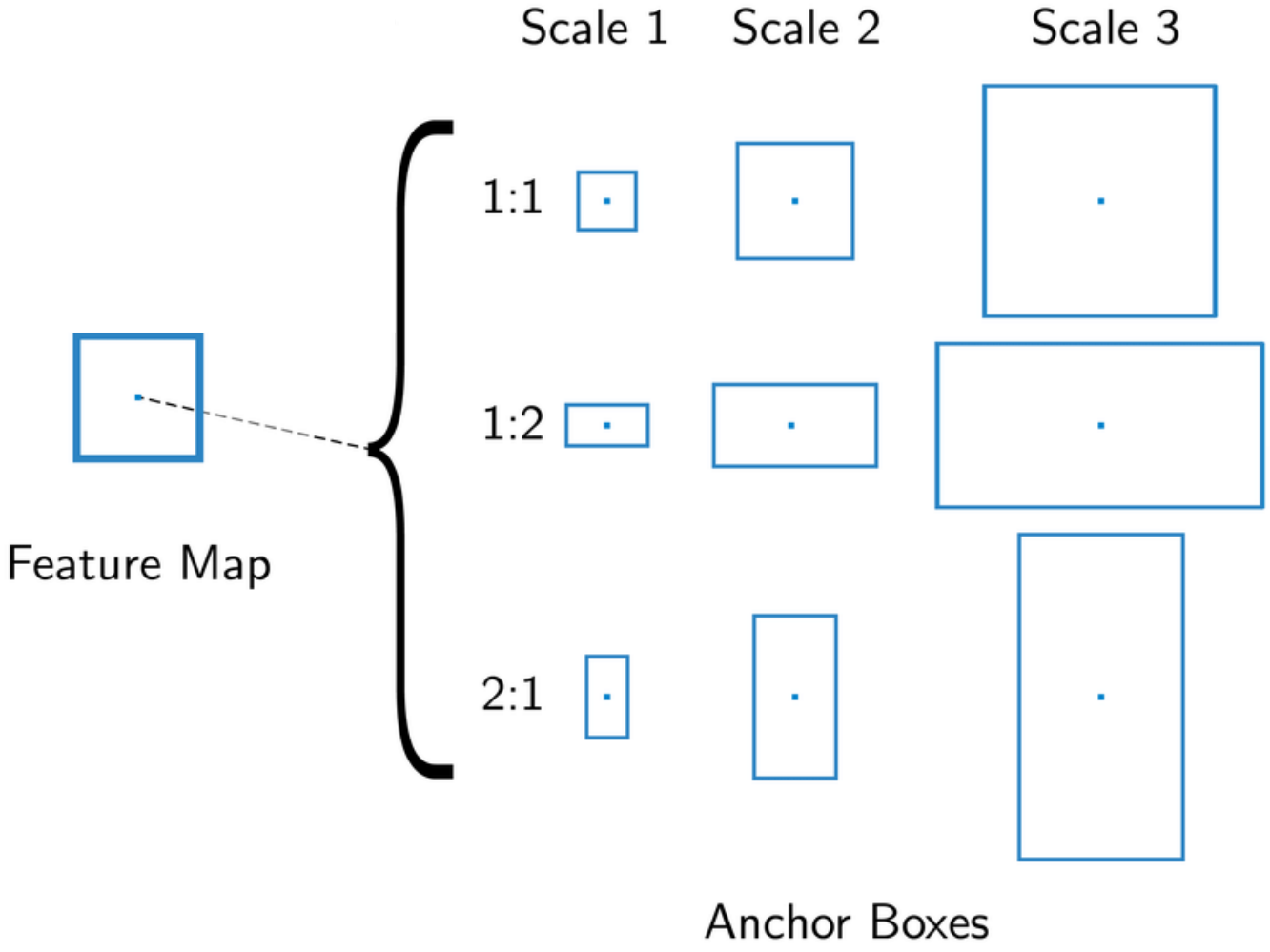
Region proposal network (RPN)

- Slide a small window over the conv5 layer
  - Predict object/no object
  - Regress bounding box coordinates
  - Box regression is with reference to *anchors* (3 scales x 3 aspect ratios)

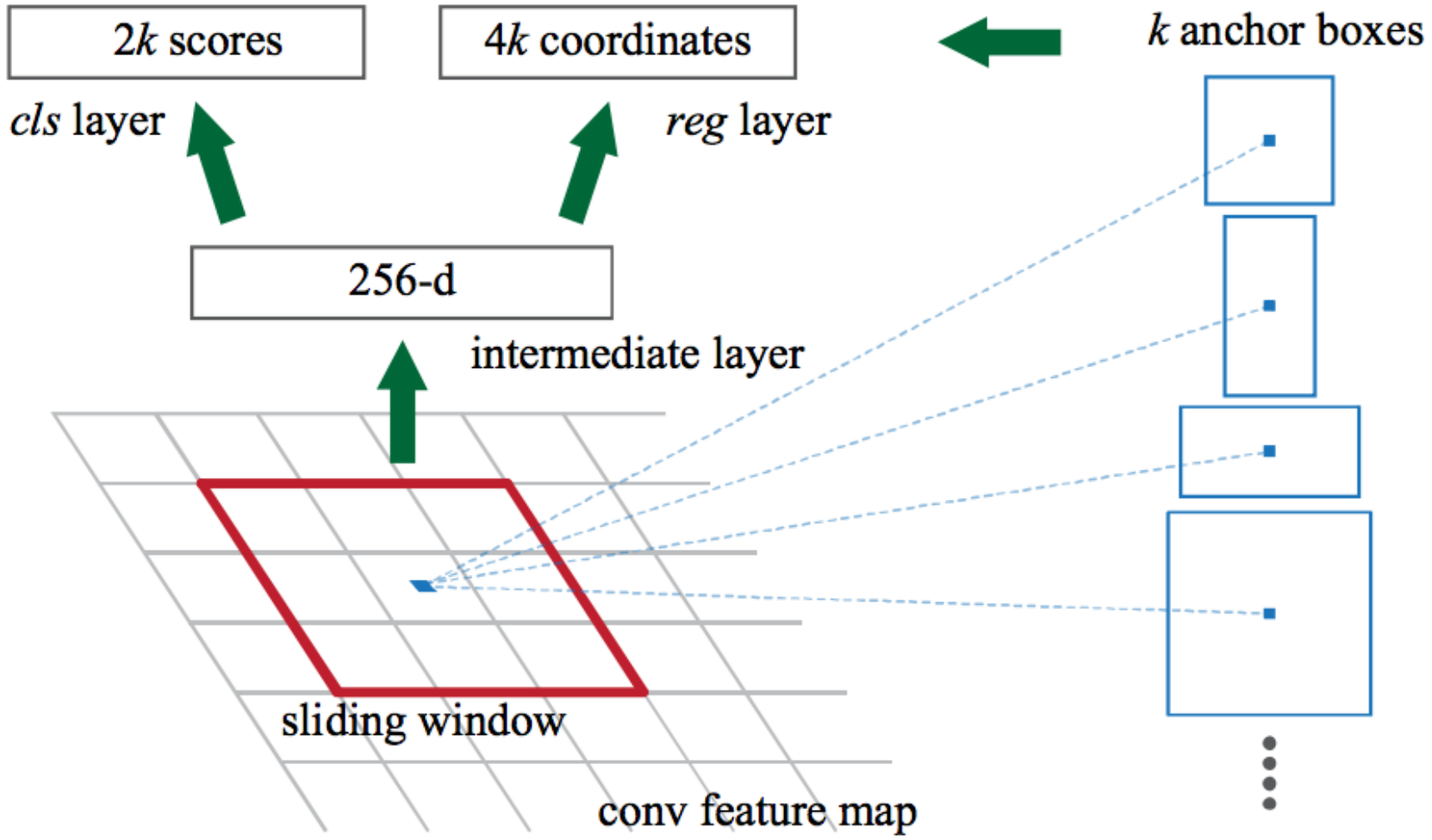


Source:

Anchor boxes

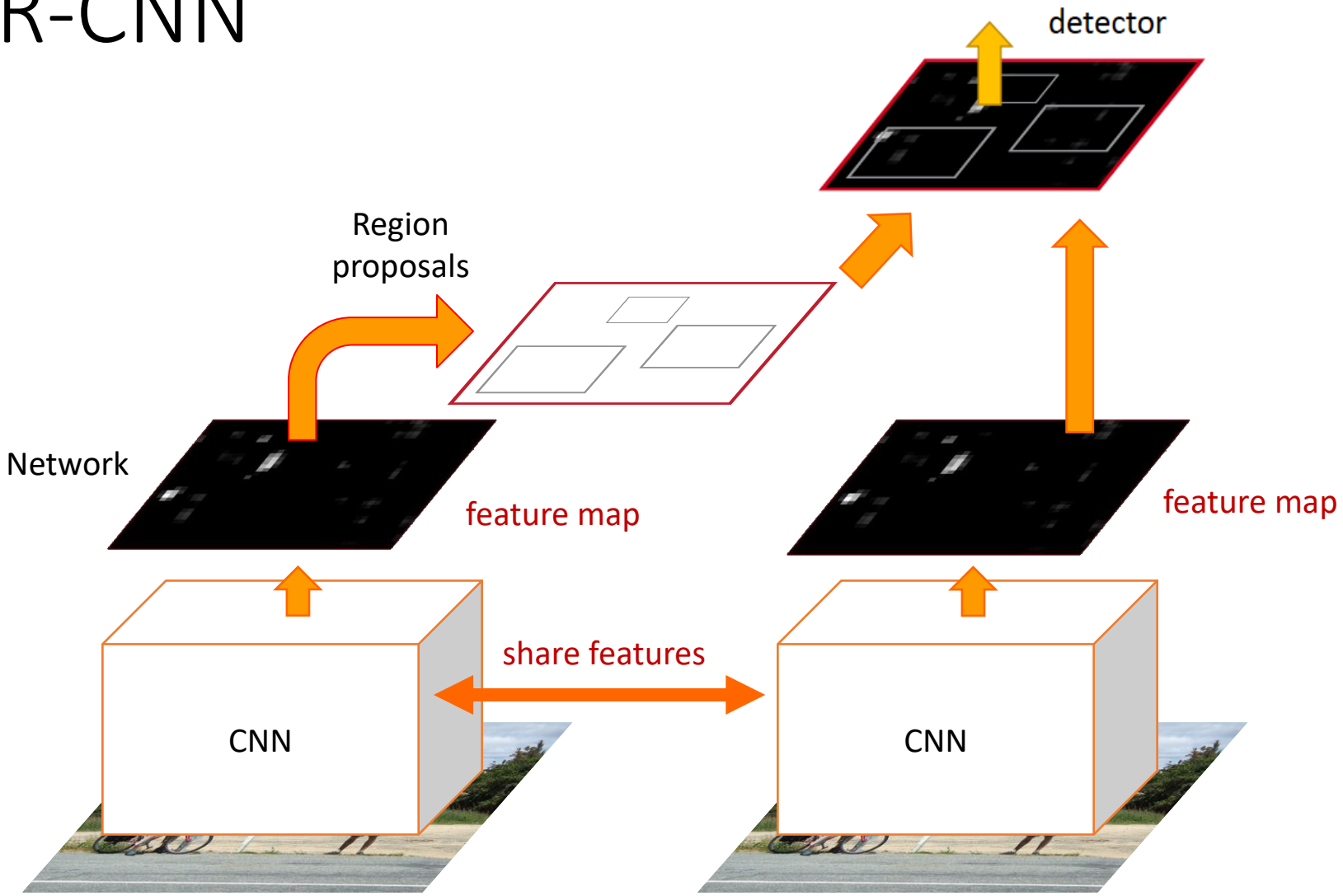


Region proposal network (RPN)



Faster R-CNN

New: Region Proposal Network



S. Ren, K. He, R. Girshick, and J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, NIPS 2015

R-CNN Summary:

R-CNN:	Propose regions first. Classify proposed regions one at a time. Output contains: label + bounding box.
Fast R-CNN:	Propose regions after the convolutional net. Use convolution implementation of sliding windows to classify all the proposed regions. End-to-end.
Faster R-CNN:	Use ConvNet to propose regions. End-to-end.

[Girshik et. al, 2013. Rich feature hierarchies for accurate object detection and semantic segmentation]
[Girshik, 2015. Fast R-CNN]
[Ren et. al, 2016. Faster R-CNN: Towards real-time object detection with region proposal networks]

Questions?

Sources for this lecture include materials from works by Sedat Ozer, Fei-Fei Li & Andrej Karpathy & Justin Johnson