Object detection

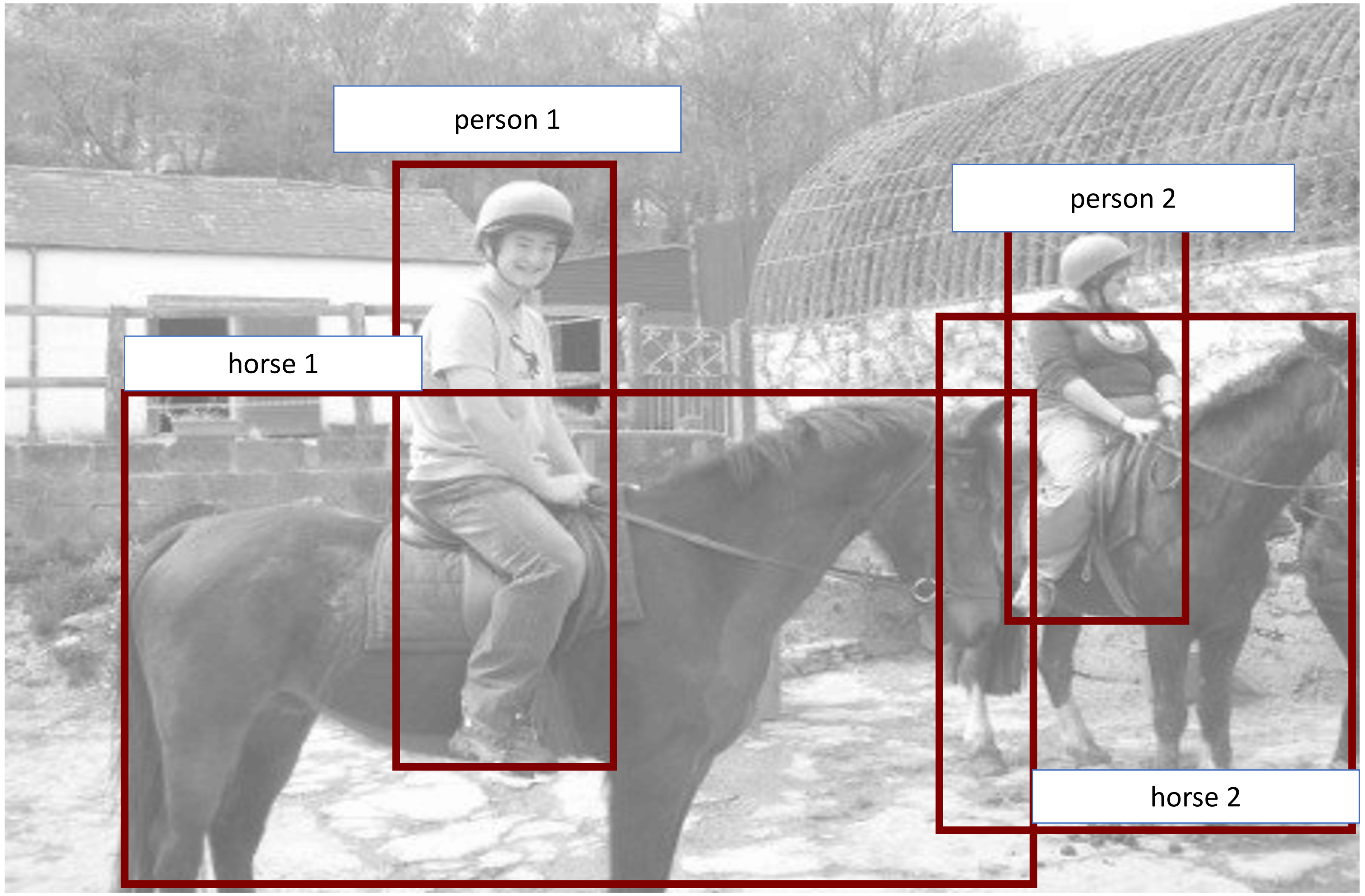
Image classification is a made-up problem

Horse? Person? House? Horse-riding?

Image classification is a made-up problem

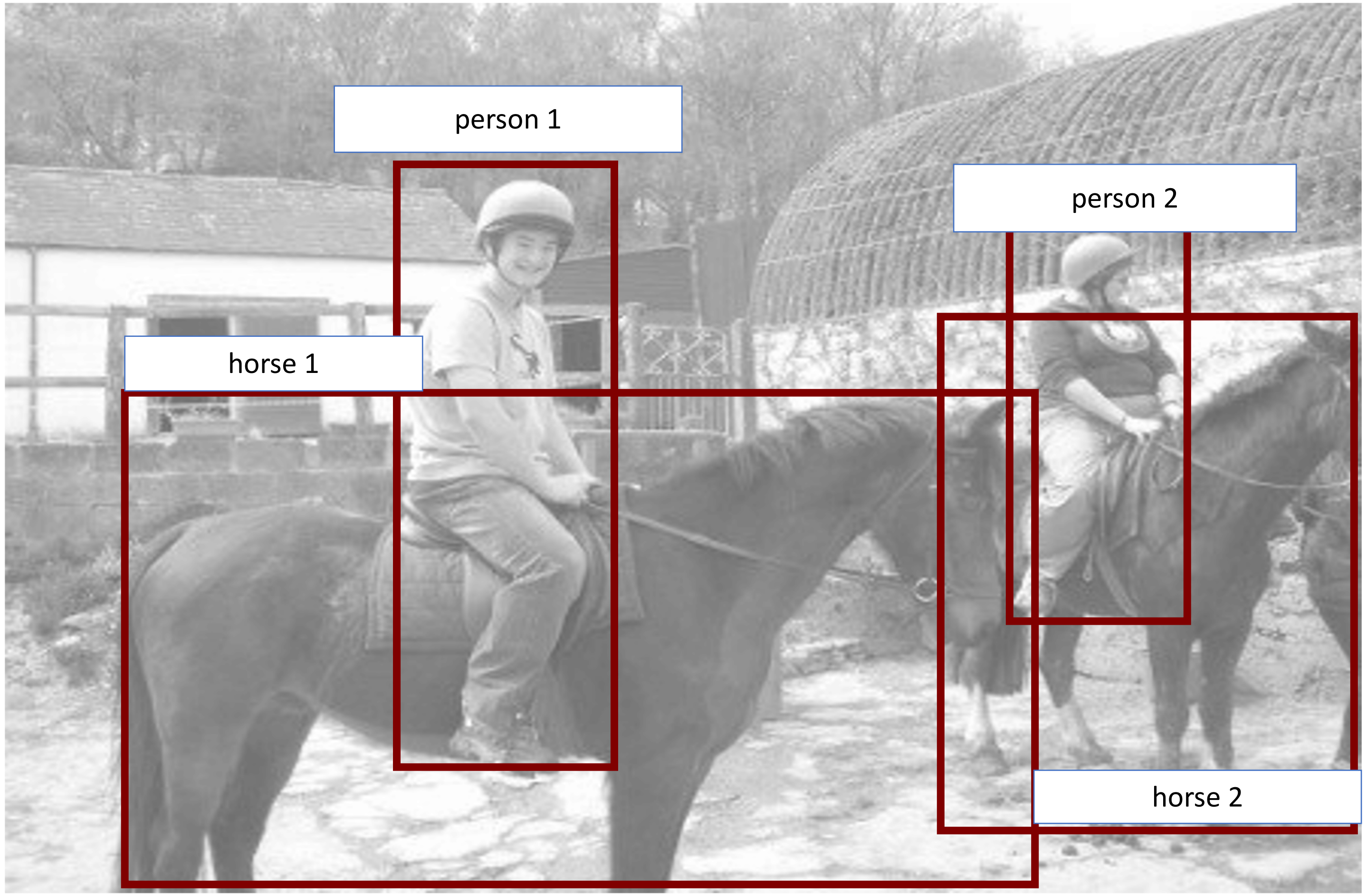
- Most images in most domains have lots of things going on
  - One label does not cut it
- There is also spatial information we want to recover
  - Where are certain objects in the scene
- Next level of recognition: *Object detection*

The Task

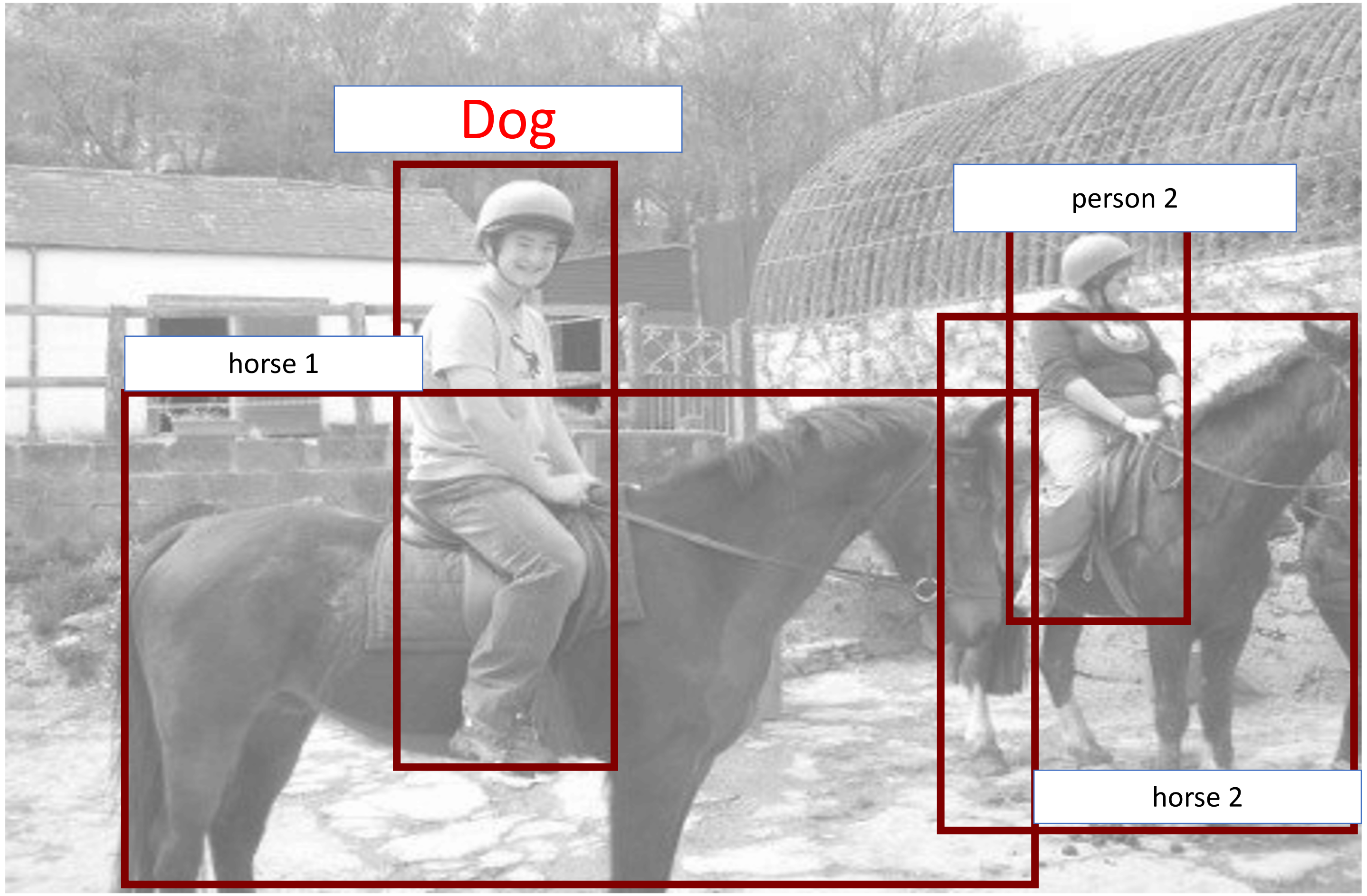


What applications can you do?

- Graphics? (cut and paste?)
- Robotics? (Navigation? Manipulation?)

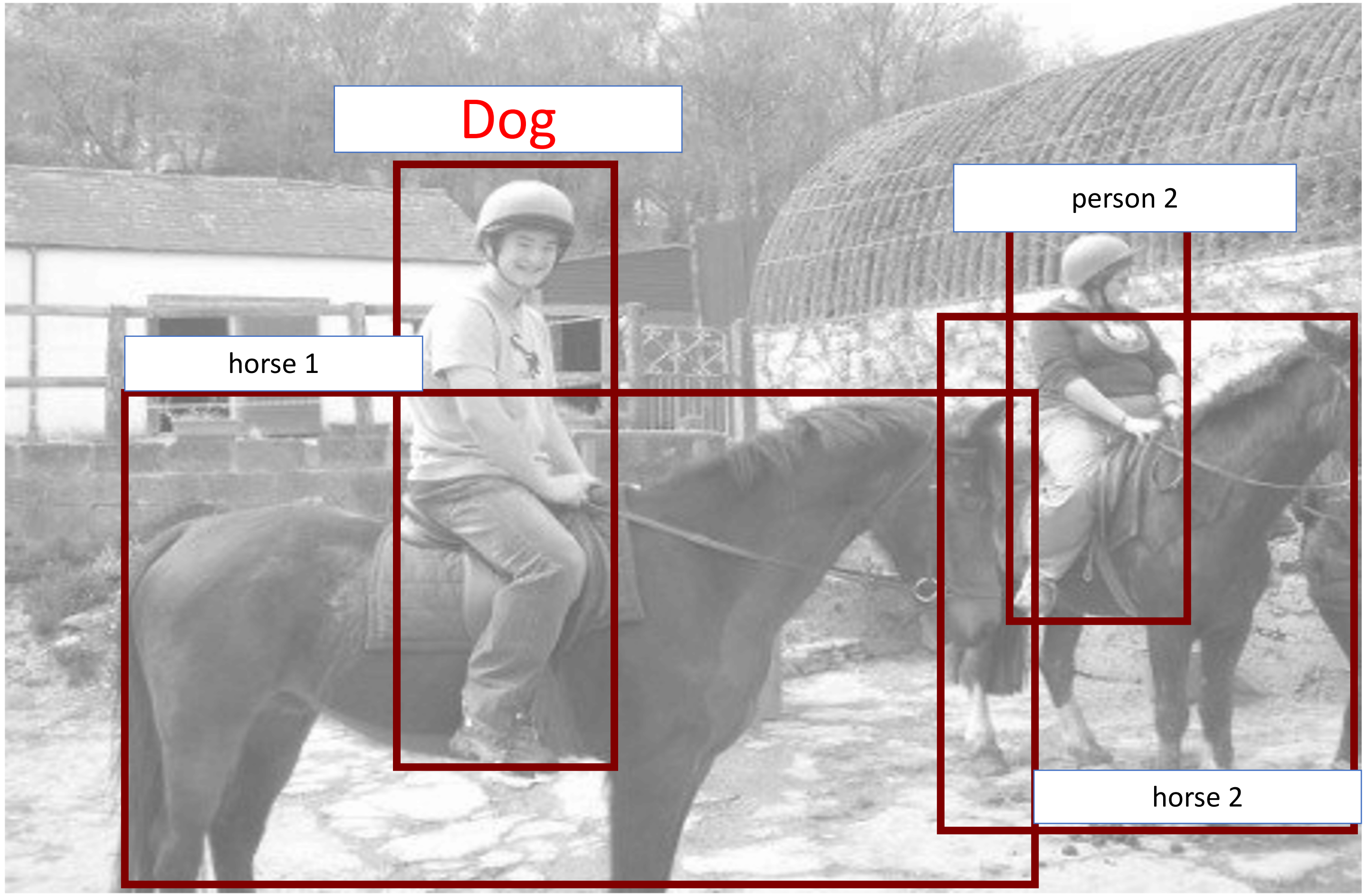


How should we measure performance?



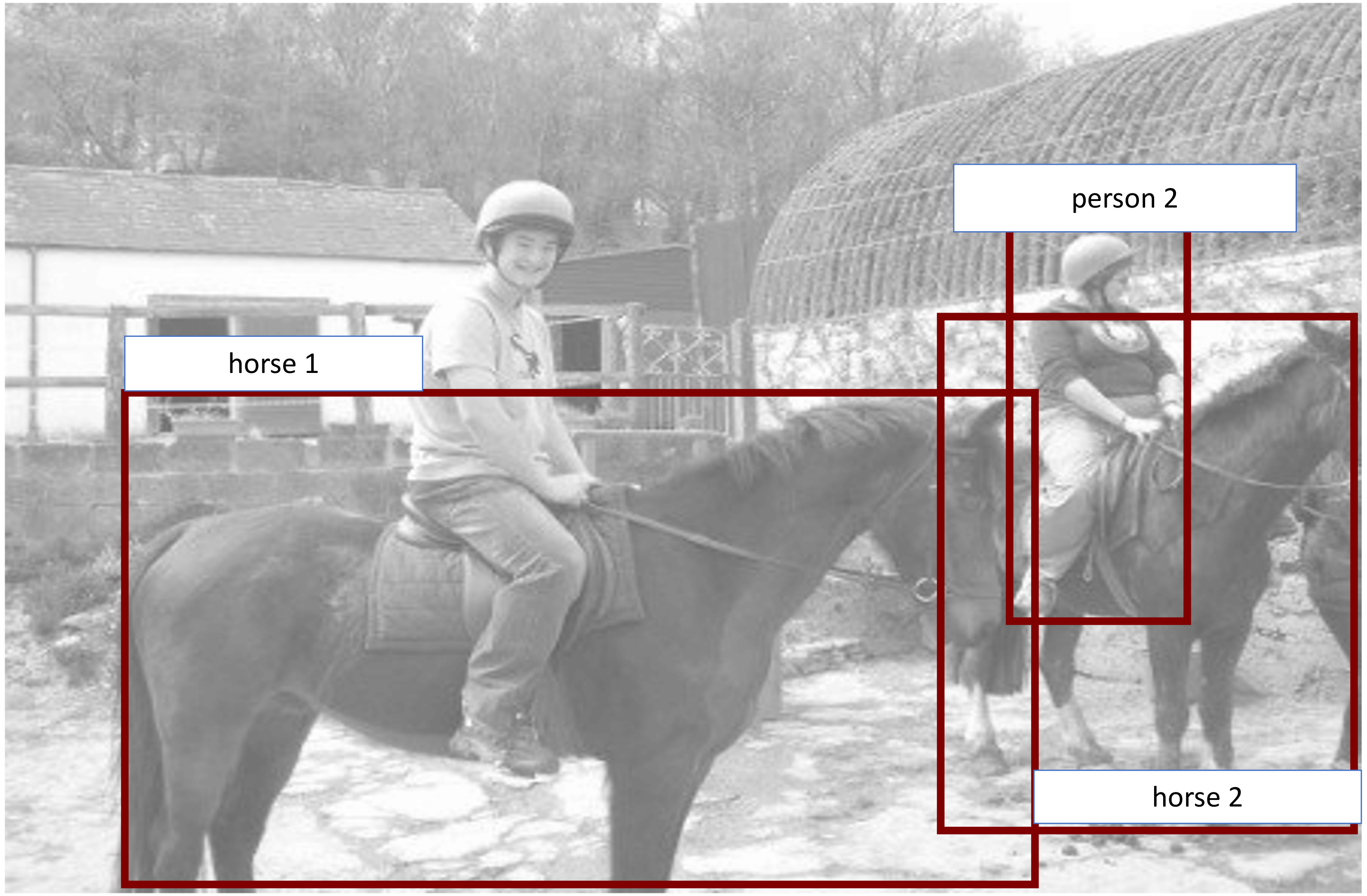
What kinds of errors?

- Incorrect labels?



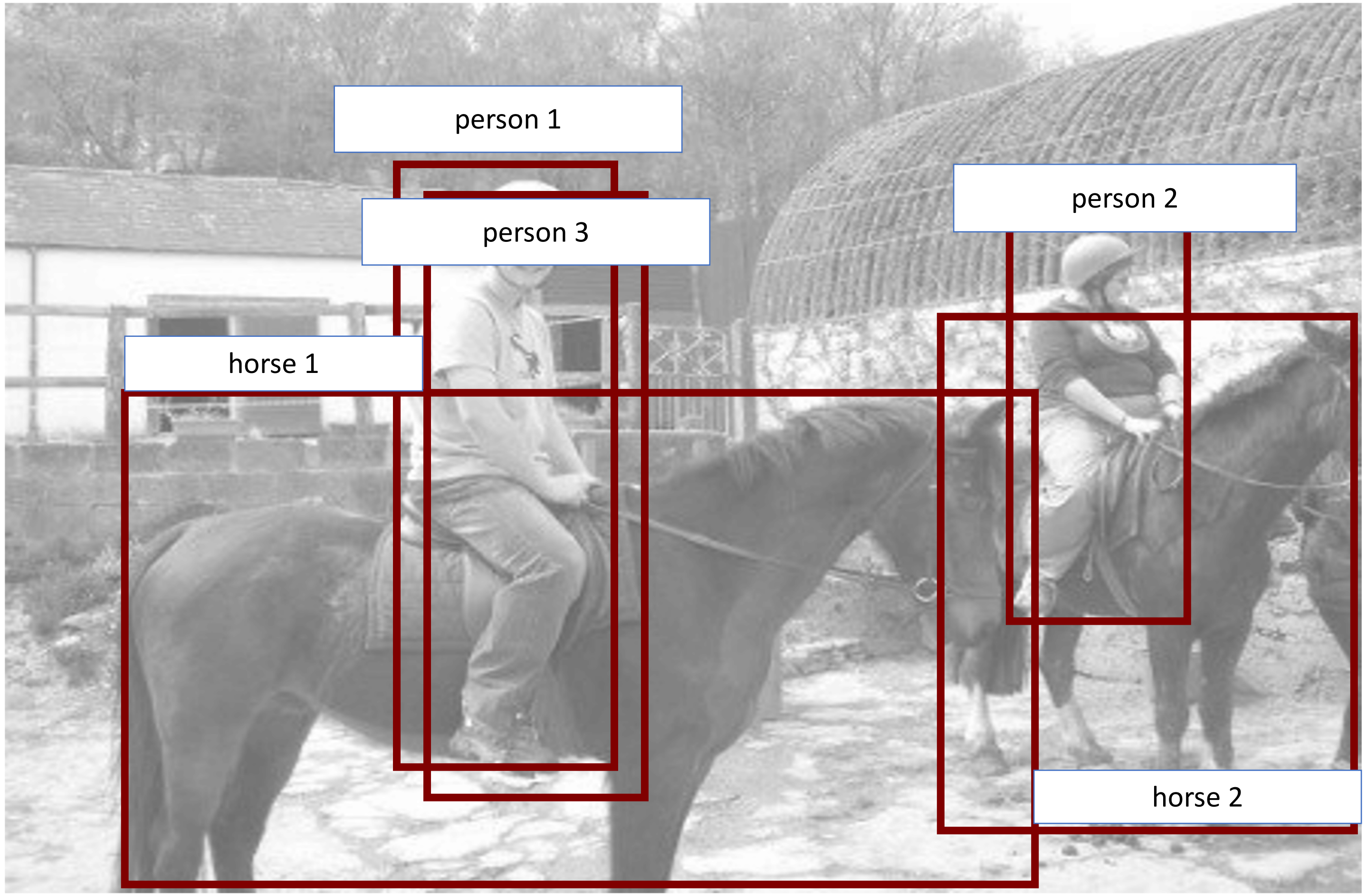
What kind of errors?

- Incorrect labels?
- Missed detections?



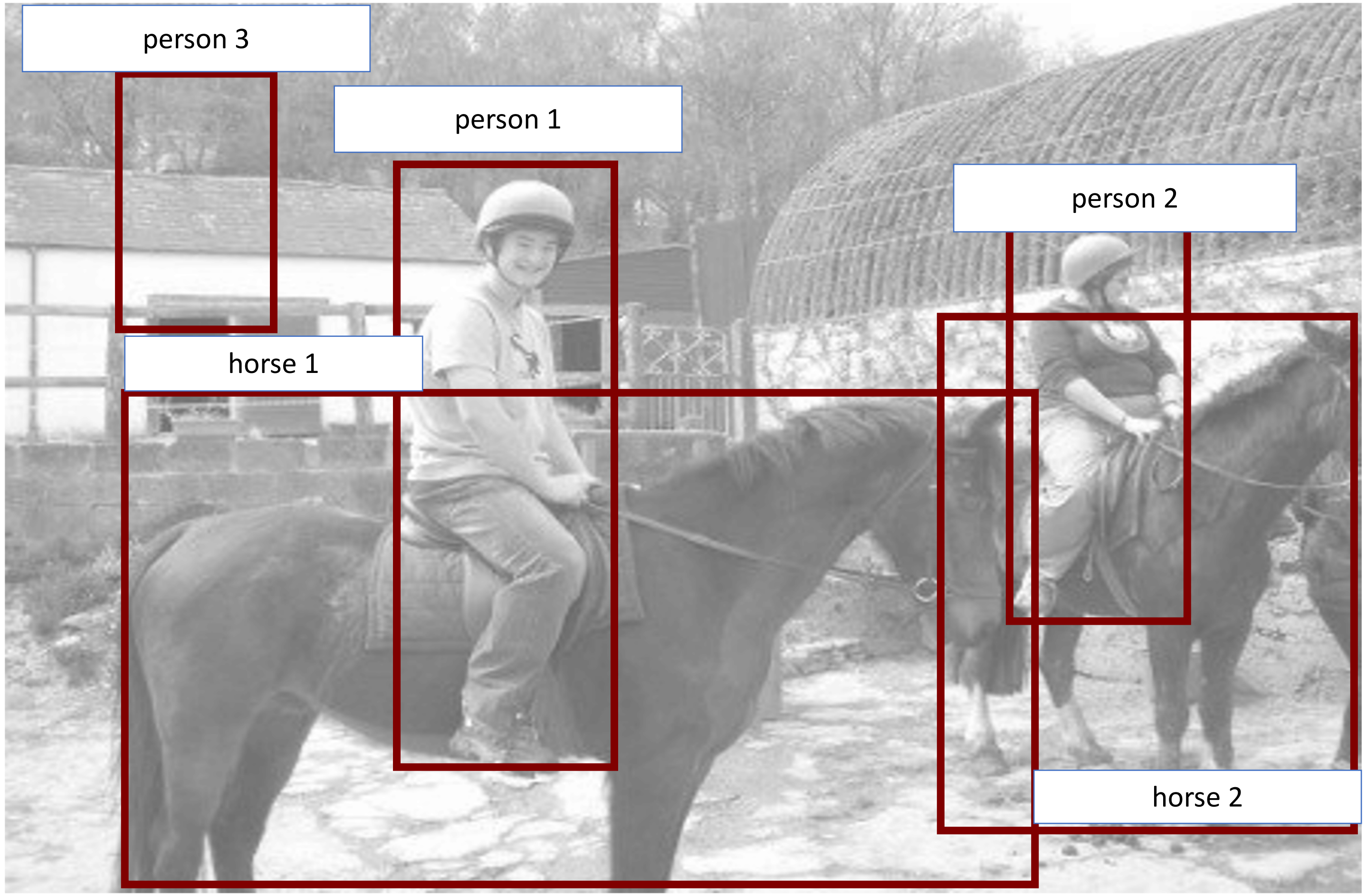
What kind of errors?

- Incorrect labels?
- Missed detections?
- Multiple detections per object?



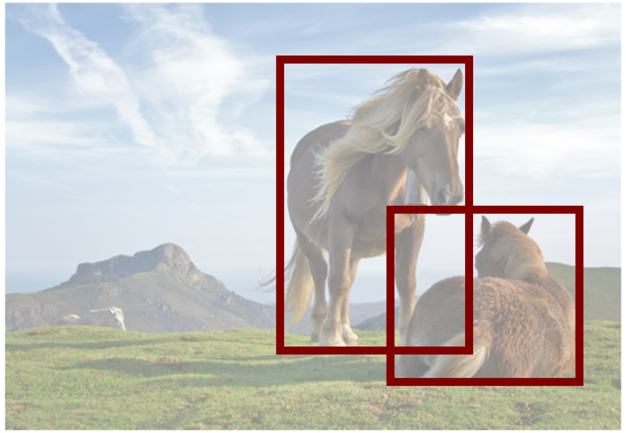
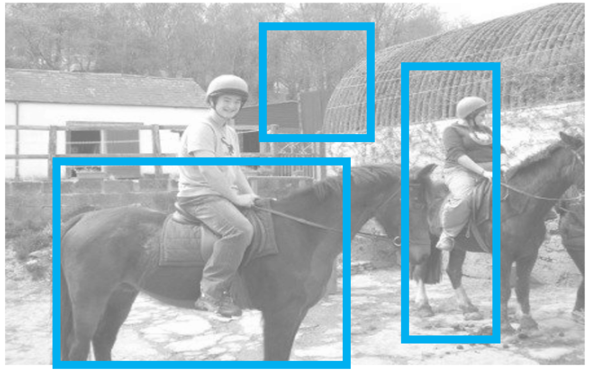
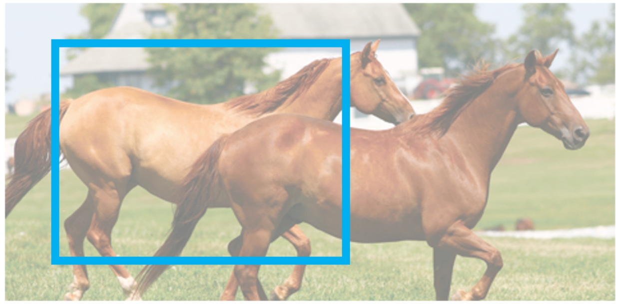
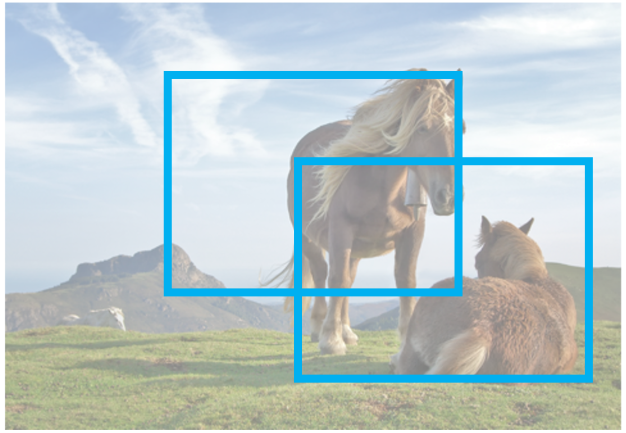
What kind of errors?

- Incorrect labels?
- Missed detections?
- Multiple detections per object?
- False positives?

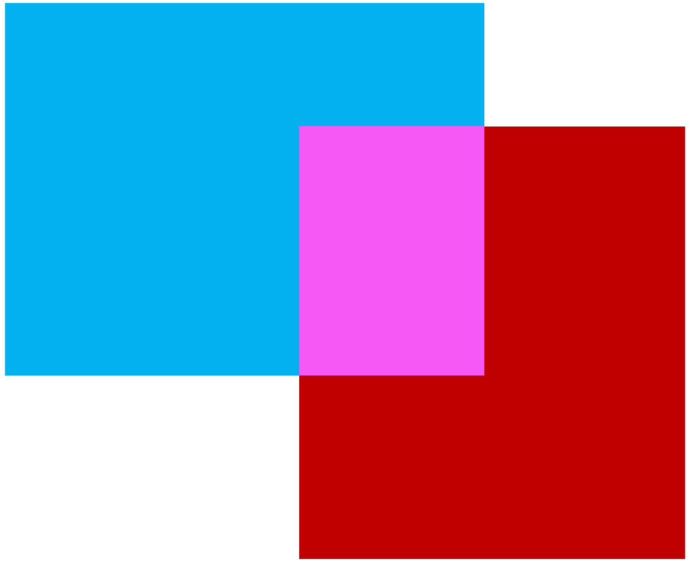


Evaluation metric



Matching detections to ground truth



$$IoU(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

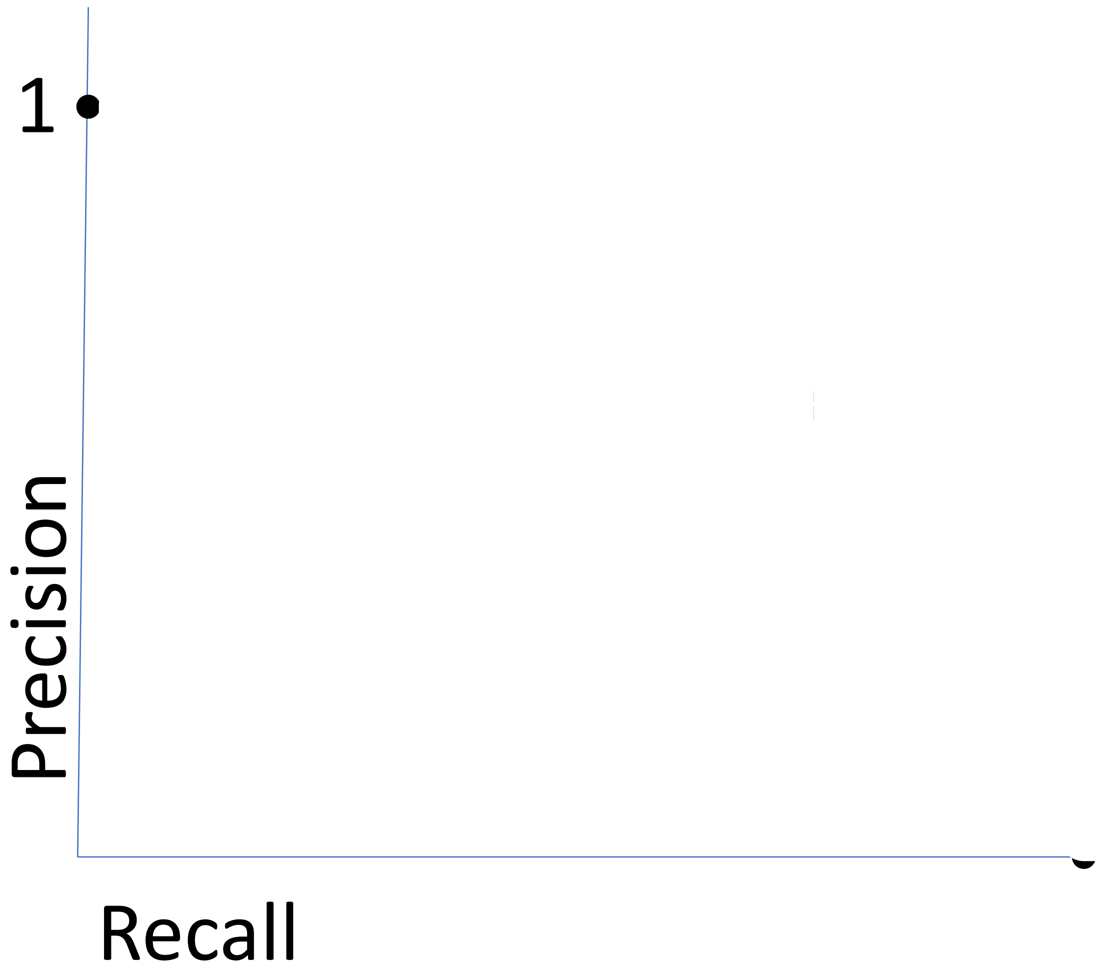
Matching detections to ground truth

- Match detection to most similar ground truth
  - highest IoU
- If IoU > 50%, mark as correct
- If multiple detections map to same ground truth, mark only one as correct
- **Precision** = #correct detections / total detections
- **Recall** = #ground truth with matched detections / total ground truth

Tradeoff between precision and recall

- ML usually gives scores or probabilities, so threshold
- Too low threshold → too many detections → low precision, high recall
- Too high threshold → too few detections → high precision, low recall
- Right tradeoff depends on application
  - Detecting cancer cells in tissue: need high recall
  - Detecting edible mushrooms in forest: need high precision

Average precision



Average precision

Average average precision

- AP marks detections with overlap > 50% as correct
- But may need better localization
- *Average* AP across multiple overlap thresholds
- Confusingly, still called average precision
- Introduced in COCO

Mean and category-wise AP

- Every category evaluated independently
- Typically report mean AP averaged over all categories
- Confusingly called “mean Average Precision”, or “mAP”

Datasets

- Face detection
- One category: face
- Frontal faces
- Fairly rigid, unoccluded

Human Face Detection in Visual Scenes. H. Rowley, S. Baluja, T. Kanade. 1995.

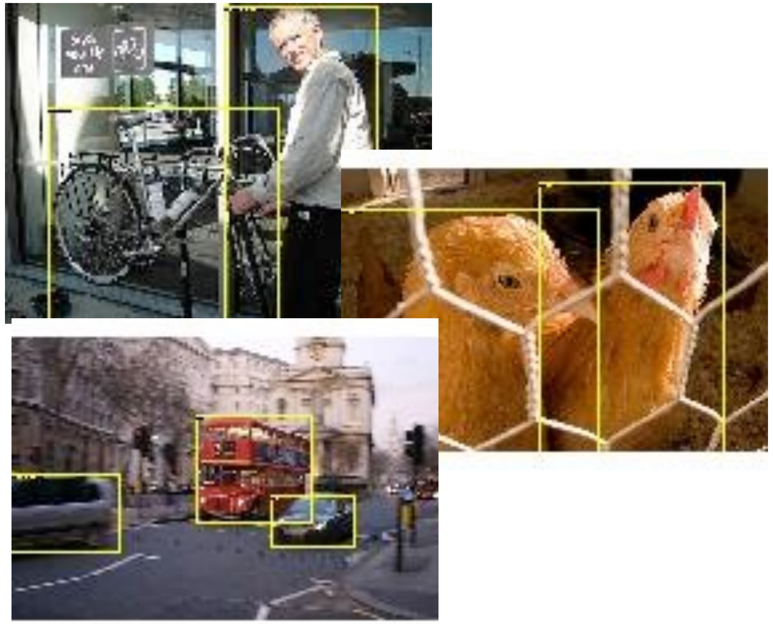
Pedestrians

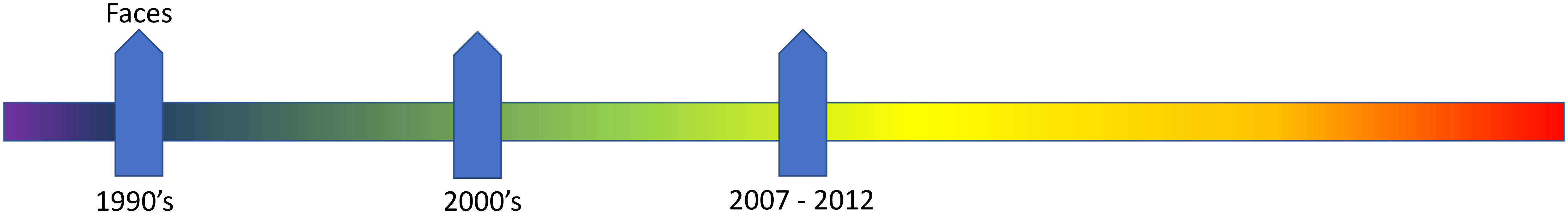
- One category: pedestrians
- Slight pose variations and small distortions
- Partial occlusions

Histograms of Oriented Gradients for Human Detection. N. Dalal and B. Triggs. CVPR 2005

PASCAL VOC

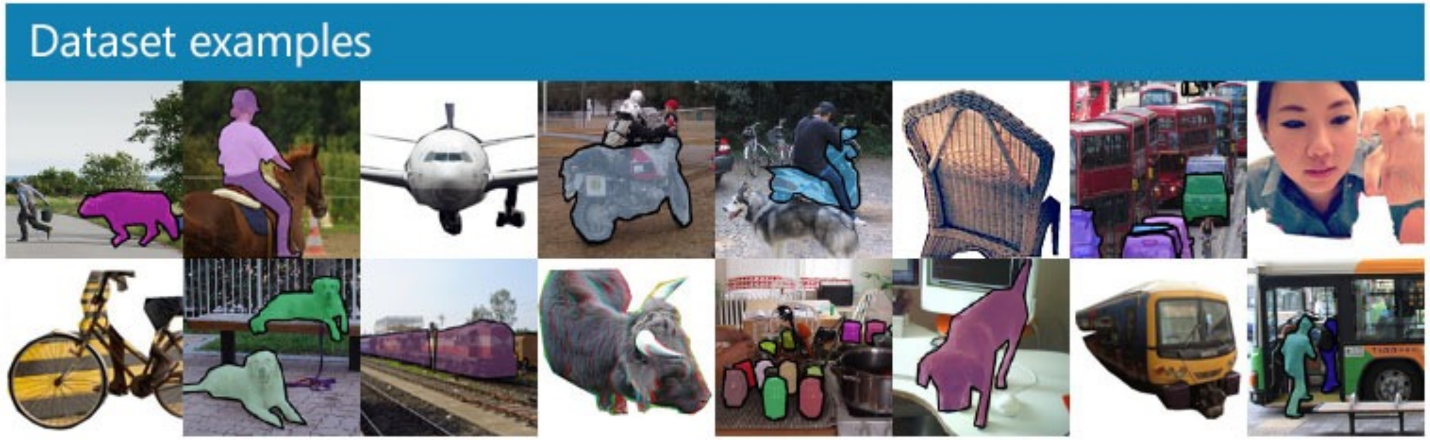
- 20 categories
- 10K images
- Large pose variations, heavy occlusions
- Generic scenes
- Cleaned up performance metric

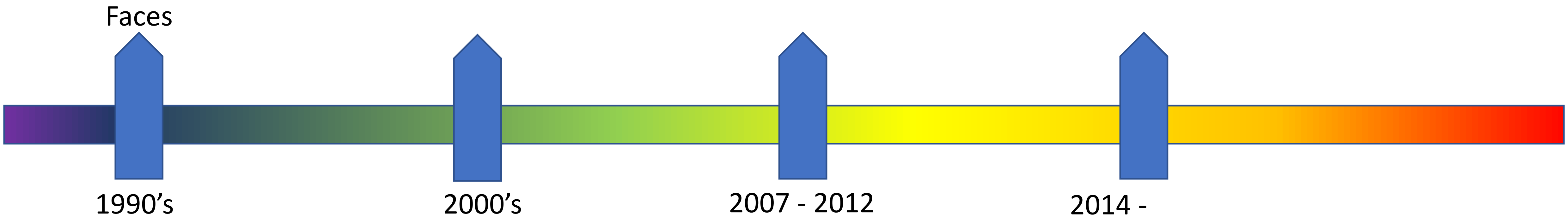




Coco

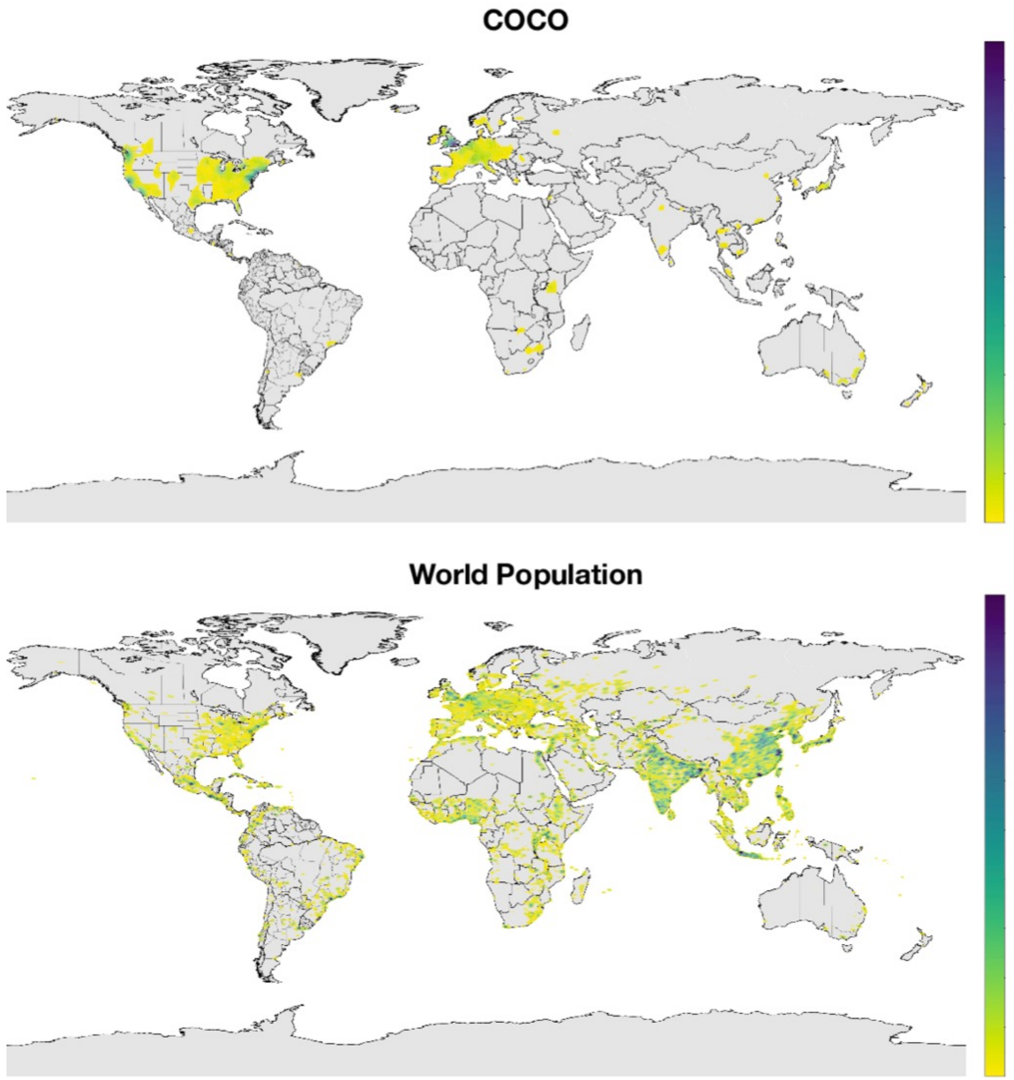
- 80 diverse categories
- 100K images
- Heavy occlusions, many objects per image, large scale variations





Are object detection datasets representative?

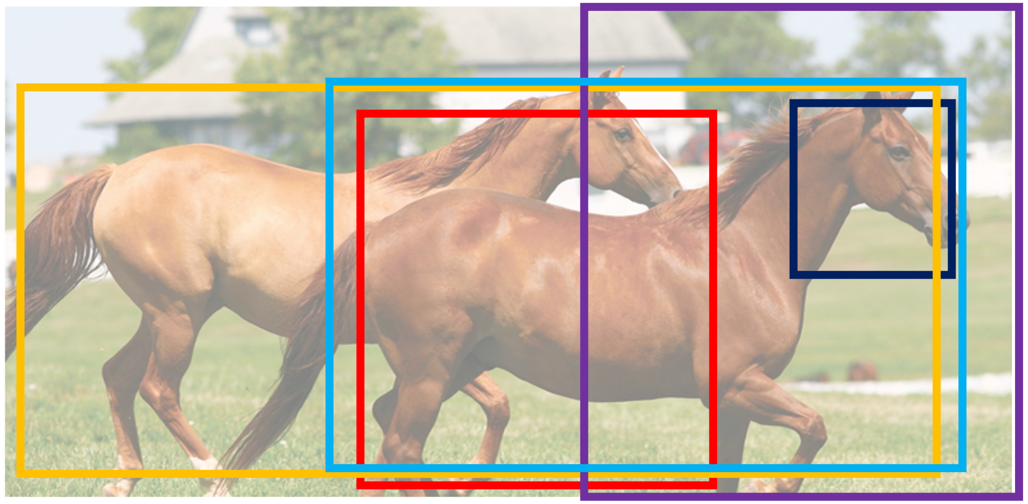
- Images typically from Flickr
  - Who uses Flickr?
- How can we get a more representative dataset?
  - Do we need one?



https://arxiv.org/abs/1906.02659

Why is detection hard(er)?

- Precise localization



Why is detection hard(er)?

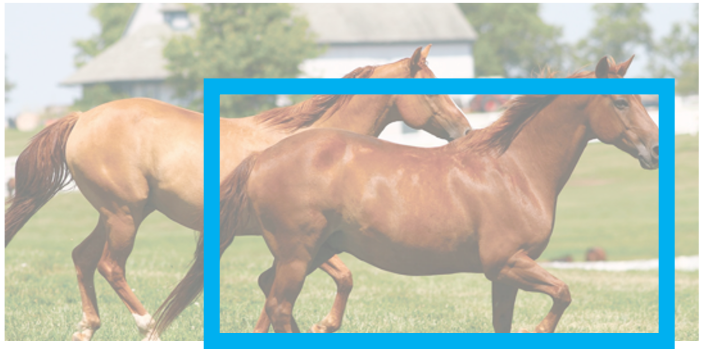
- Much larger impact of pose

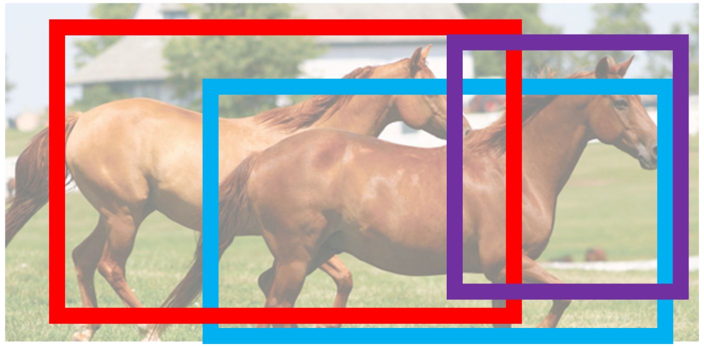
Why is detection hard(er)?

- Occlusion makes localization difficult

Why is detection hard(er)?

- Counting



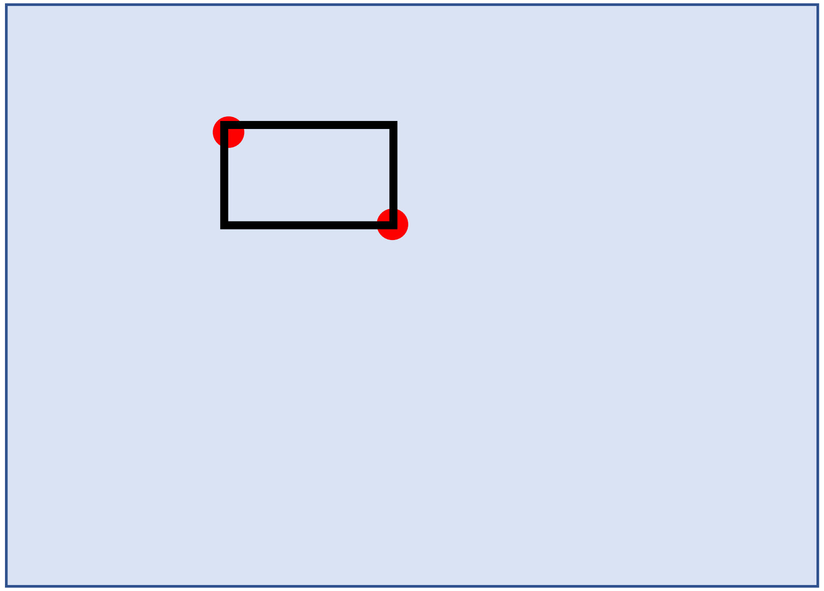


Why is detection hard(er)?

- Small objects

Detection as classification

- Run through every possible box and classify
  - Well-localized object of class k or not?
- How many boxes?
  - Every pair of pixels = 1 box
  - $\binom{N}{2}$ = O(N^2)
  - For 300 x 500 image, N = 150K
  - 2.25 x 10^{10} boxes!
- Related challenge: almost all boxes are negative!

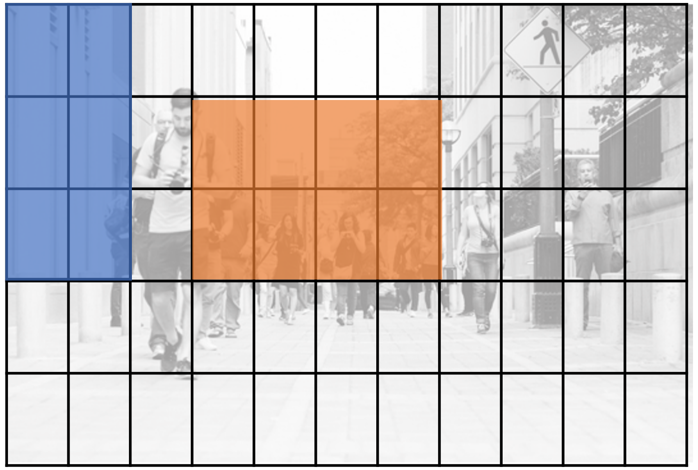


Idea 1: scanning window

- Fix size
- Fix stride
- Crop boxes and classify
- Alternatively
  - Compute collection of feature maps
  - Convolve with filter

Multiple object sizes

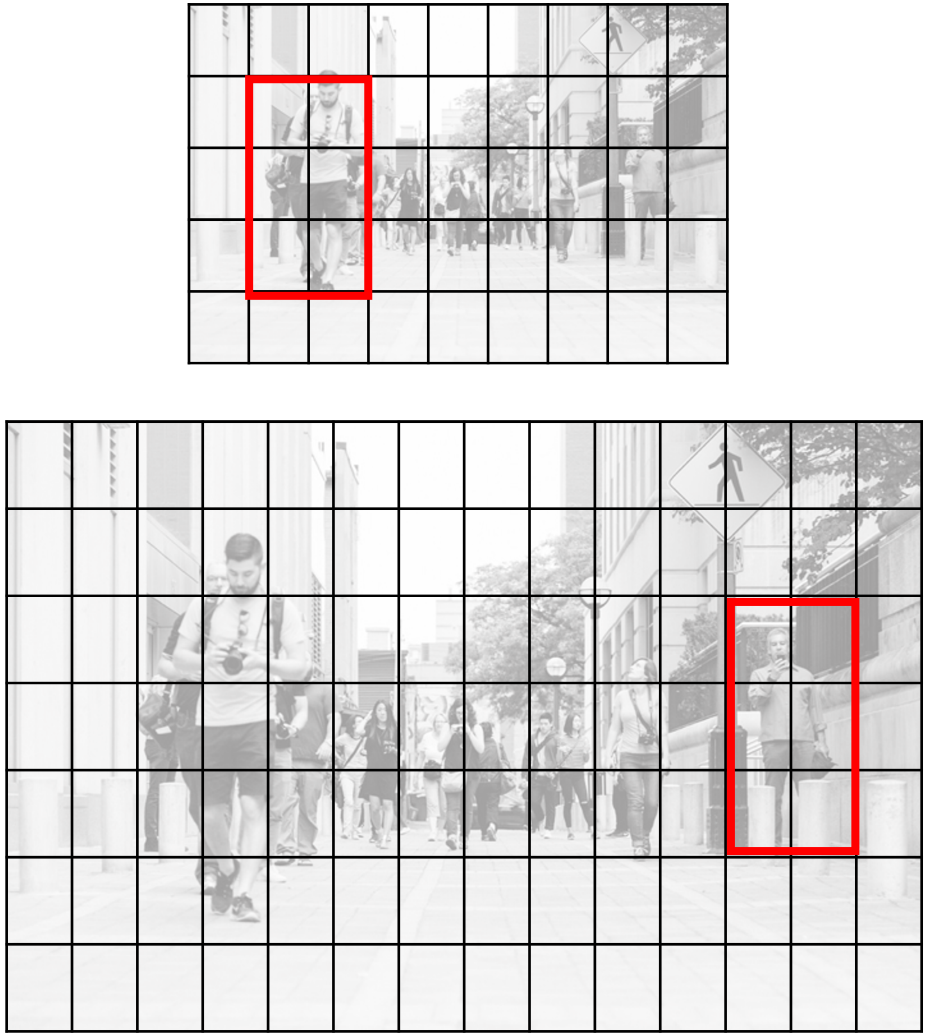
- Objects can appear at any size
- *Discretize* set of sizes into a few different sizes
    - Sometimes called “anchors”
- Train separate classifier for each size



Dealing with large scale changes

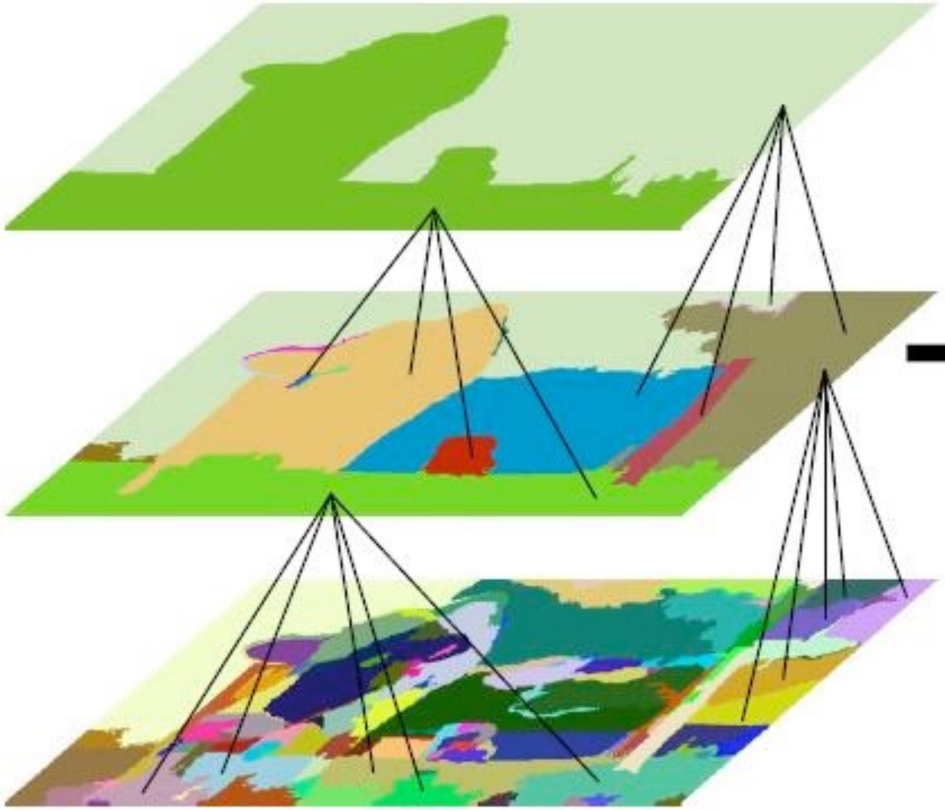
Dealing with large scale changes

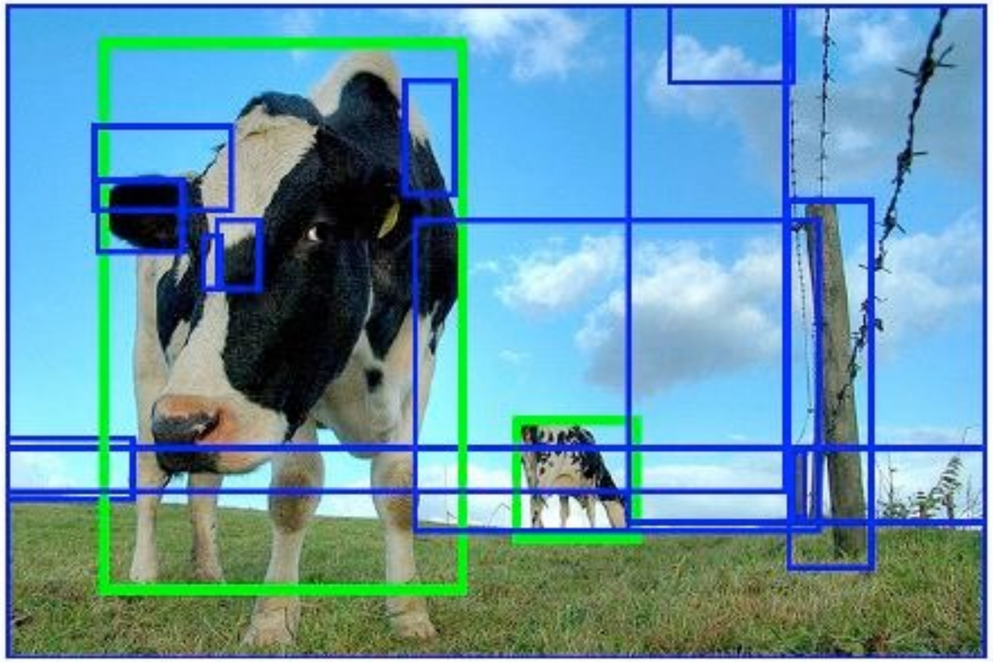
- Use an image pyramid
- Run same detector at multiple scales
- Take union of results



Idea 2: Object proposals

- Use segmentation to produce ~5K "candidates"





Selective Search for Object Recognition
J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, A. W. M. Smeulders
In International Journal of Computer Vision 2013.

Object proposals

- Basic idea: use grouping cues to identify segments that are likely to be objects
- Multiple versions
  - Do graph cuts with different seeds
  - Oversegment and then combinatorially group nearby objects

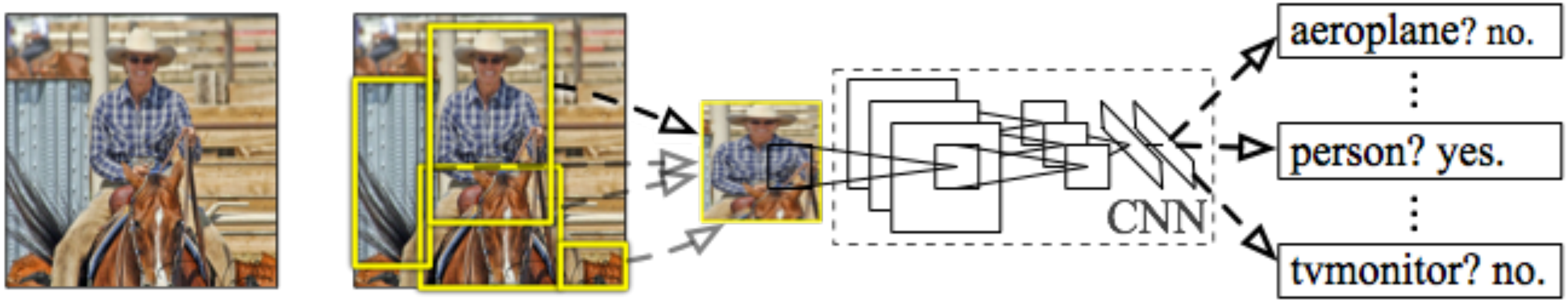
Two classes of object detection approaches

- Object proposal-based
- Also called two-stage detectors
- Canonical examples
  - R-CNN family
- Pros:
  - Smaller number of candidates to classify
  - Less class imbalance
  - “Cascade” approach
- Cons:
  - More complex, slower
  - Can miss due to missed proposals

- Scanning window-based
- Also called single-stage detectors
- Canonical examples
  - SSD family
- Pros
  - Simpler
  - Faster
- Cons
  - Larger number of candidates, more class imbalance
  - Can miss due to mismatched size

ConvNet-based object detection

R-CNN: Regions with CNN features



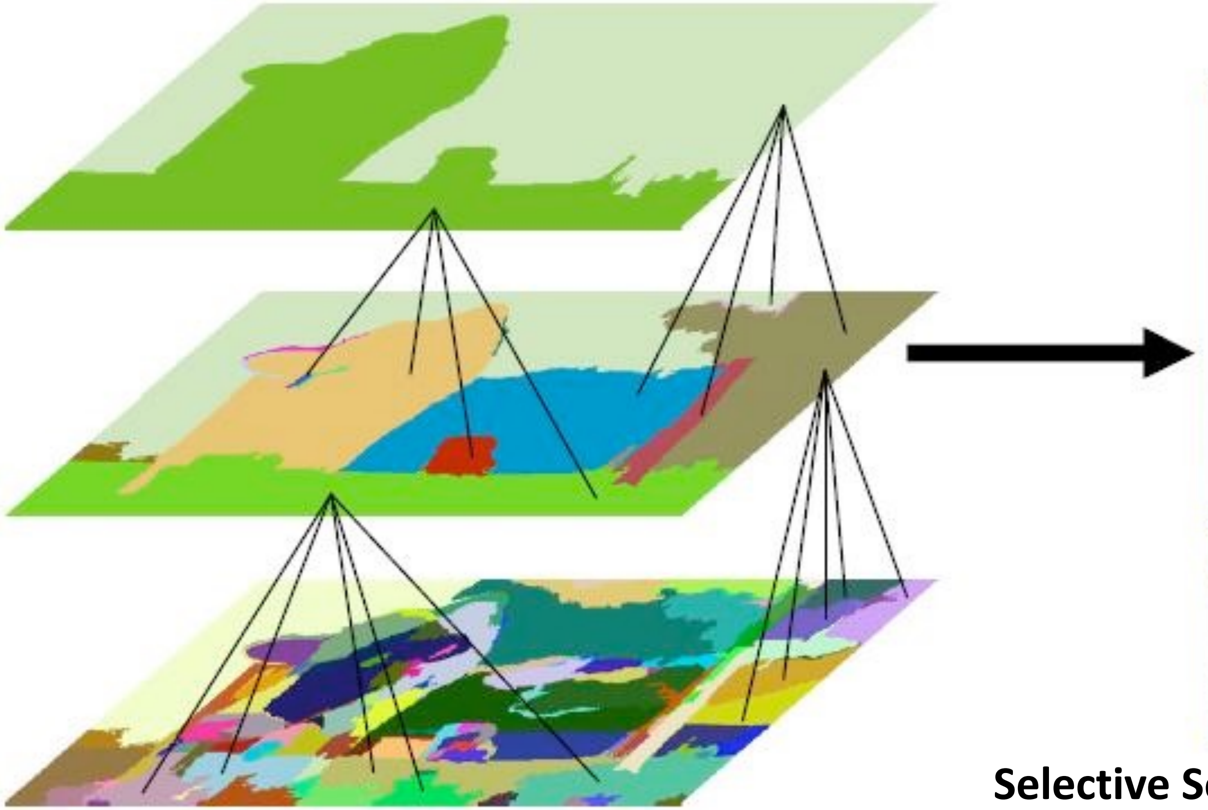
Input image | Extract region proposals (~2k / image) | Compute CNN features | Classify regions (linear SVM)

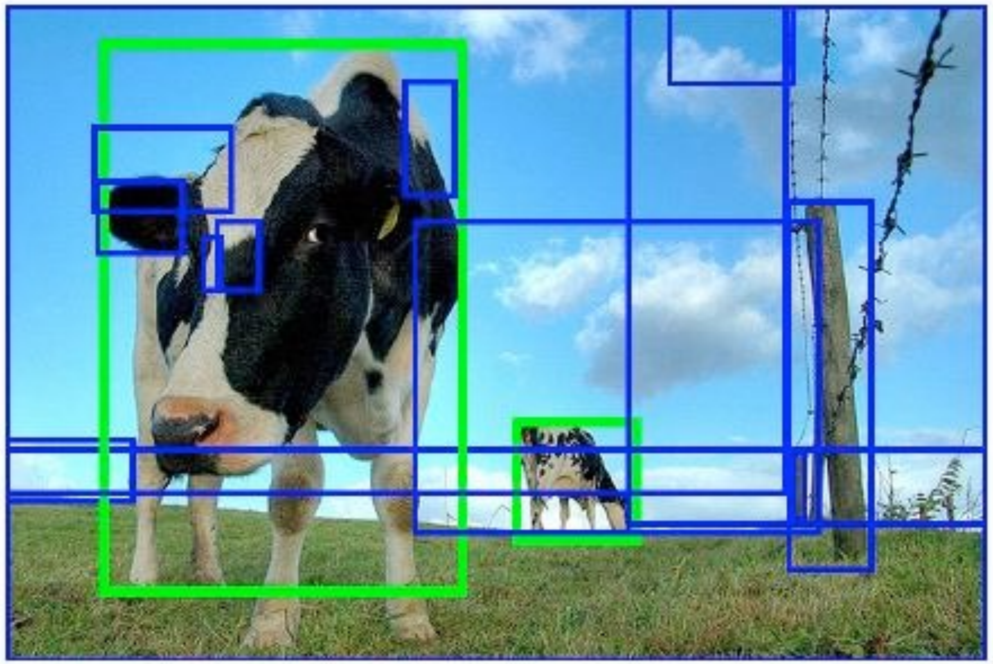
Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation
R. Girshick, J. Donahue, T. Darrell, J. Malik
IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014

Slide credit : Ross Girshick

Step 1: Object proposals

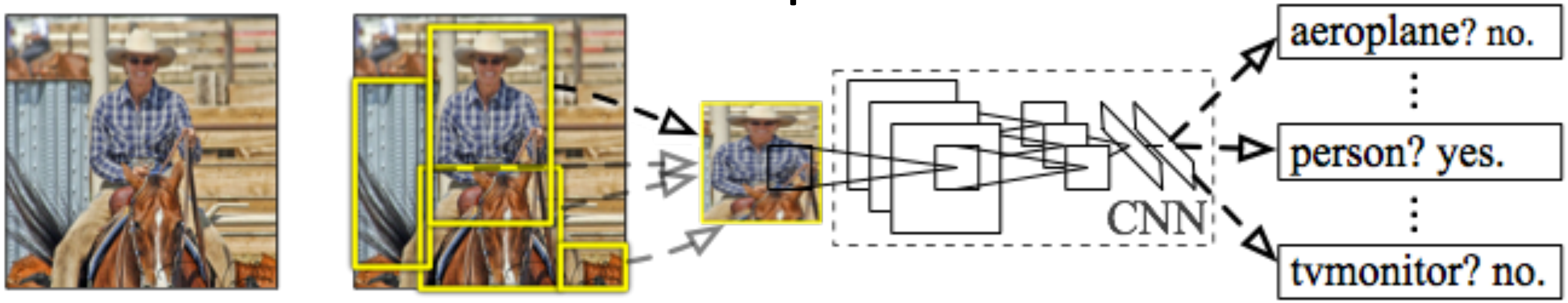
- Use segmentation to produce ~5K candidates





Selective Search for Object Recognition
J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, A. W. M. Smeulders
In International Journal of Computer Vision 2013.

R-CNN at test time: Step 2



Input image

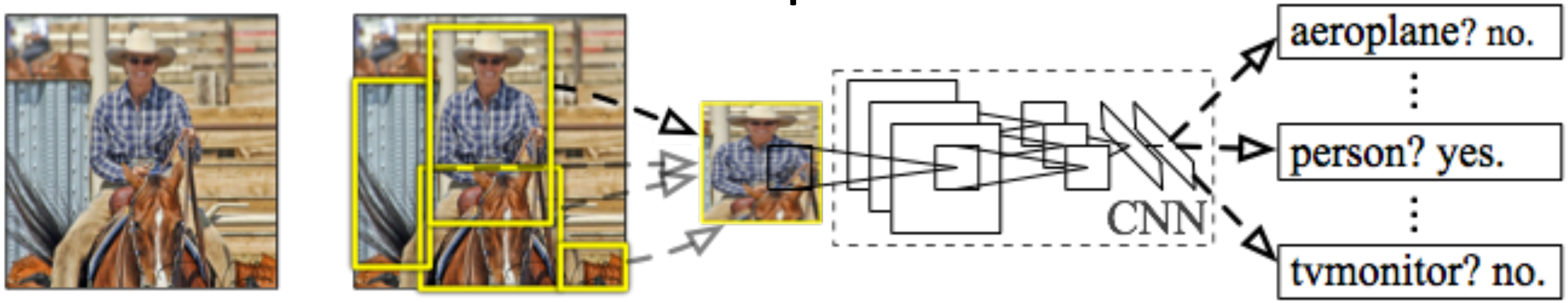
Extract region proposals (~2k / image)

Compute CNN features

a. Crop

Slide credit : Ross Girshick

R-CNN at test time: Step 2



Input image

Extract region proposals (~2k / image)

→ Compute CNN features

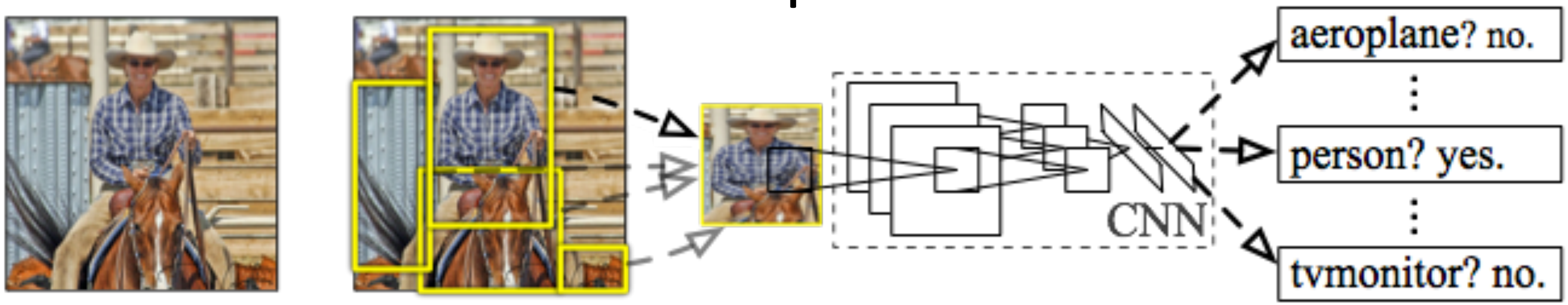
227 x 227

a. Crop

b. Scale (anisotropic)

Slide credit : Ross Girshick

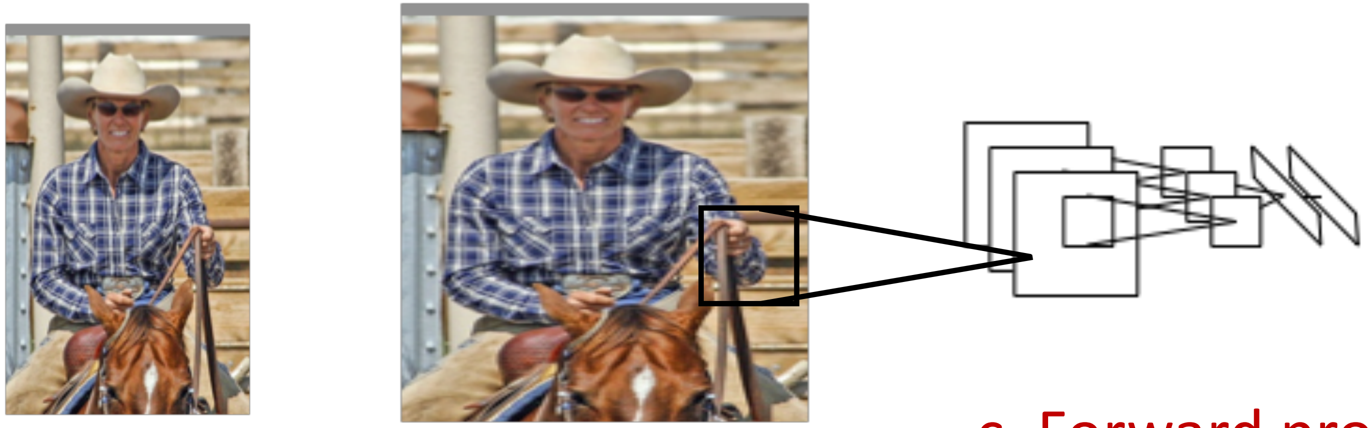
R-CNN at test time: Step 2



Input image

Extract region proposals (~2k / image)

Compute CNN features



1. Crop

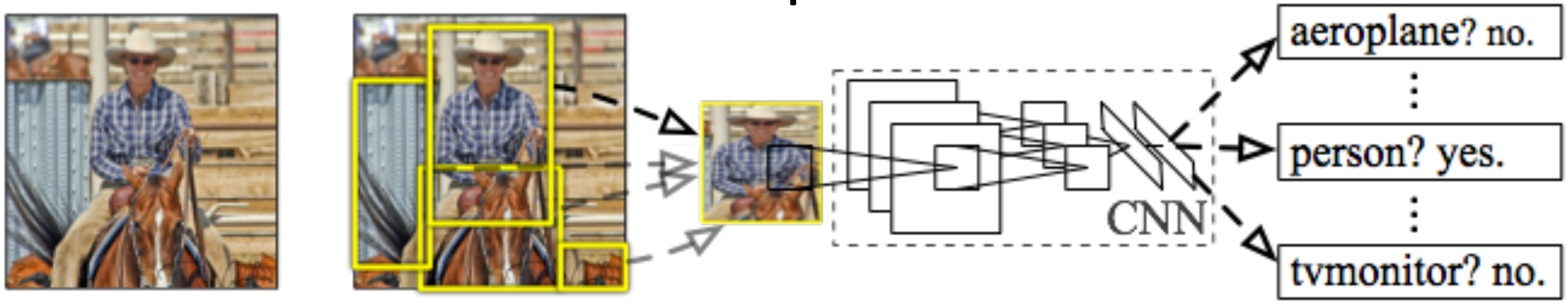
b. Scale (anisotropic)

c. Forward propagate

Output: “fc_7” features

Slide credit : Ross Girshick

R-CNN at test time: Step 3

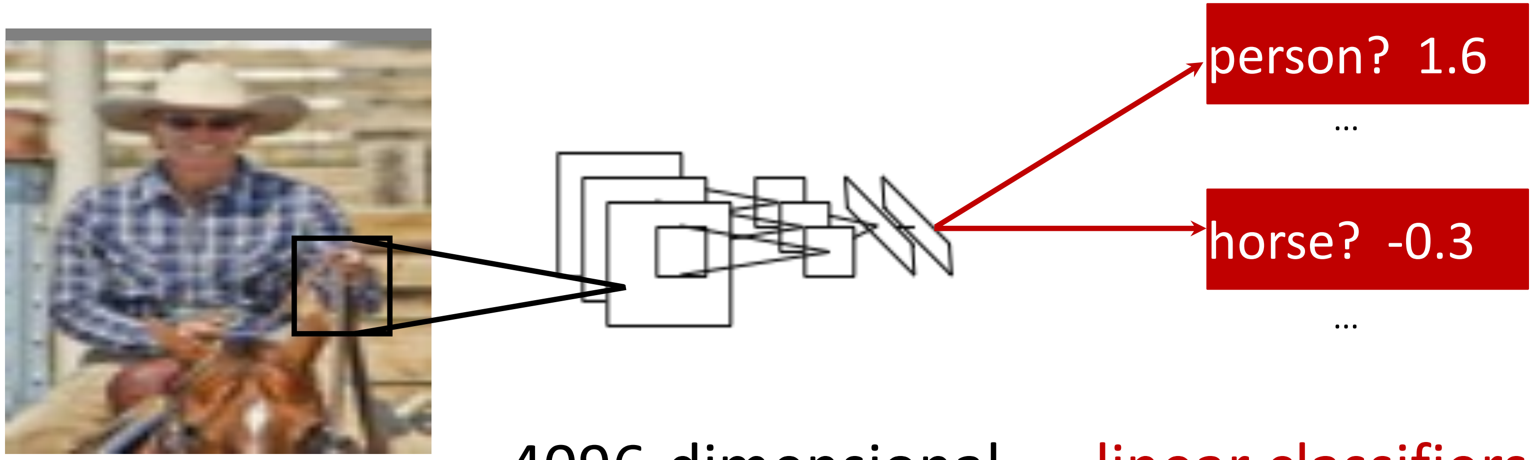


Input image

Extract region proposals (~2k / image)

Compute CNN features

Classify regions



Warped proposal

4096-dimensional fc7 feature vector

linear classifiers (SVM or softmax)

Slide credit : Ross Girshick

Step 4: Object proposal refinement

Original
proposal

Linear regression

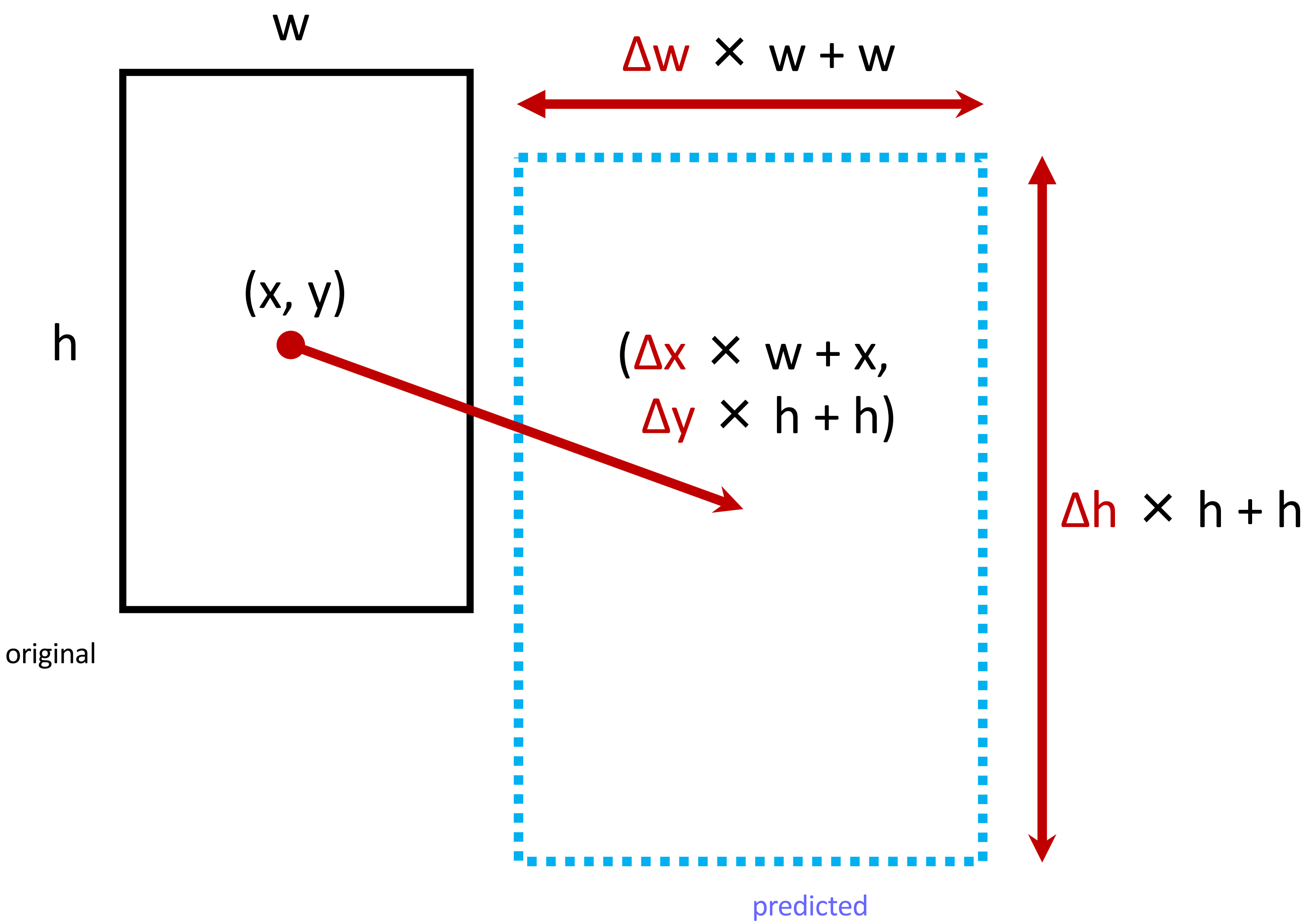
on CNN features

Predicted
object bounding box

Bounding-box regression

Slide credit : Ross Girshick

Bounding-box regression



Slide credit : Ross Girshick

Training R-CNN

- Produce proposals using off-the-shelf approach (not learning-based).
- Pre-process proposals: label proposals with overlap > 0.5 with class label, others as background
- In each iteration, sample 75% negative proposals and 25% positive proposals

Other details - Non-max suppression

Non-max suppression

- Might find the same object with different sized-boxes and different scales
- But must fire exactly once on each object
- Idea: if two detections overlap significantly (>50% IoU), drop lower scoring one

Other details - Non-max suppression

- Go down the list of detections starting from highest scoring
- Eliminate any detection that overlaps highly with a higher scoring detection
- Separate, heuristic step

Training R-CNN

- Train convolutional network on ImageNet classification
- *Finetune* on detection
  - Classification problem!
  - Proposals with IoU > 50% are positives
  - Sample fixed proportion of positives in each batch because of imbalance

R-CNN results on PASCAL

	VOC 2007	VOC 2010
DPM v5 (Girshick et al. 2011)	33.7%	29.6%
UVA sel. search (Uijlings et al. 2013)		35.1%
Regionlets (Wang et al. 2013)	**41.7%**	39.7%
SegDPM (Fidler et al. 2013)		**40.4%**

Reference systems

metric: mean average precision (higher is better)

Slide credit : Ross Girshick

R-CNN results on PASCAL

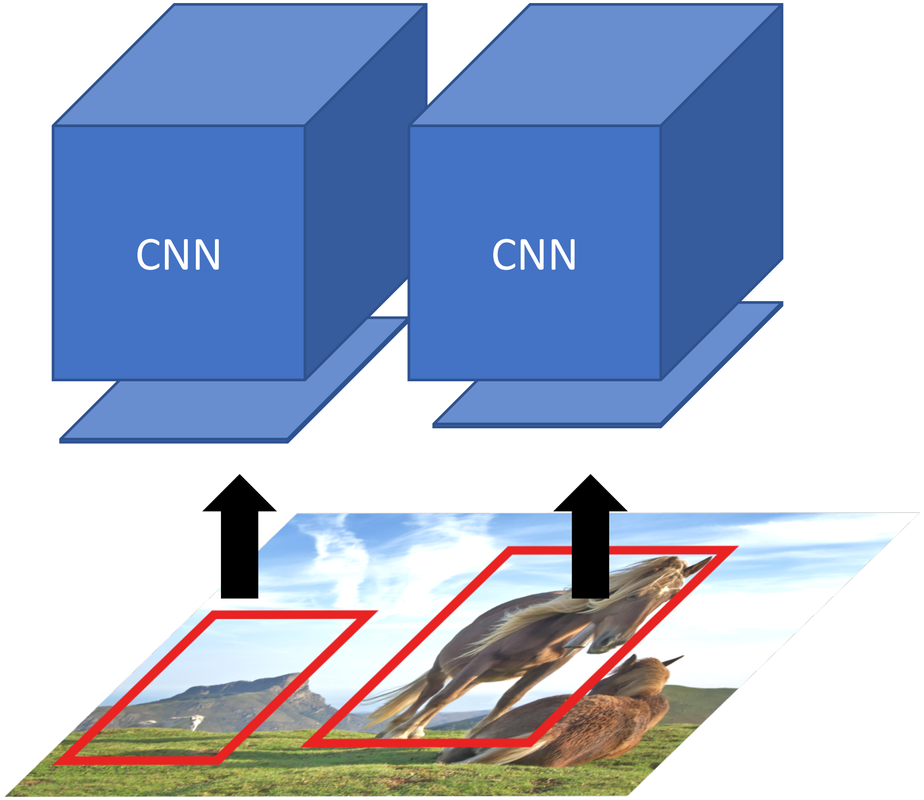
	VOC 2007	VOC 2010
DPM v5 (Girshick et al. 2011)	33.7%	29.6%
UVA sel. search (Uijlings et al. 2013)		35.1%
Regionlets (Wang et al. 2013)	41.7%	39.7%
SegDPM (Fidler et al. 2013)		40.4%
R-CNN	54.2%	50.2%
R-CNN + bbox regression	**58.5%**	**53.7%**

metric: mean average precision (higher is better)

Slide credit : Ross Girshick

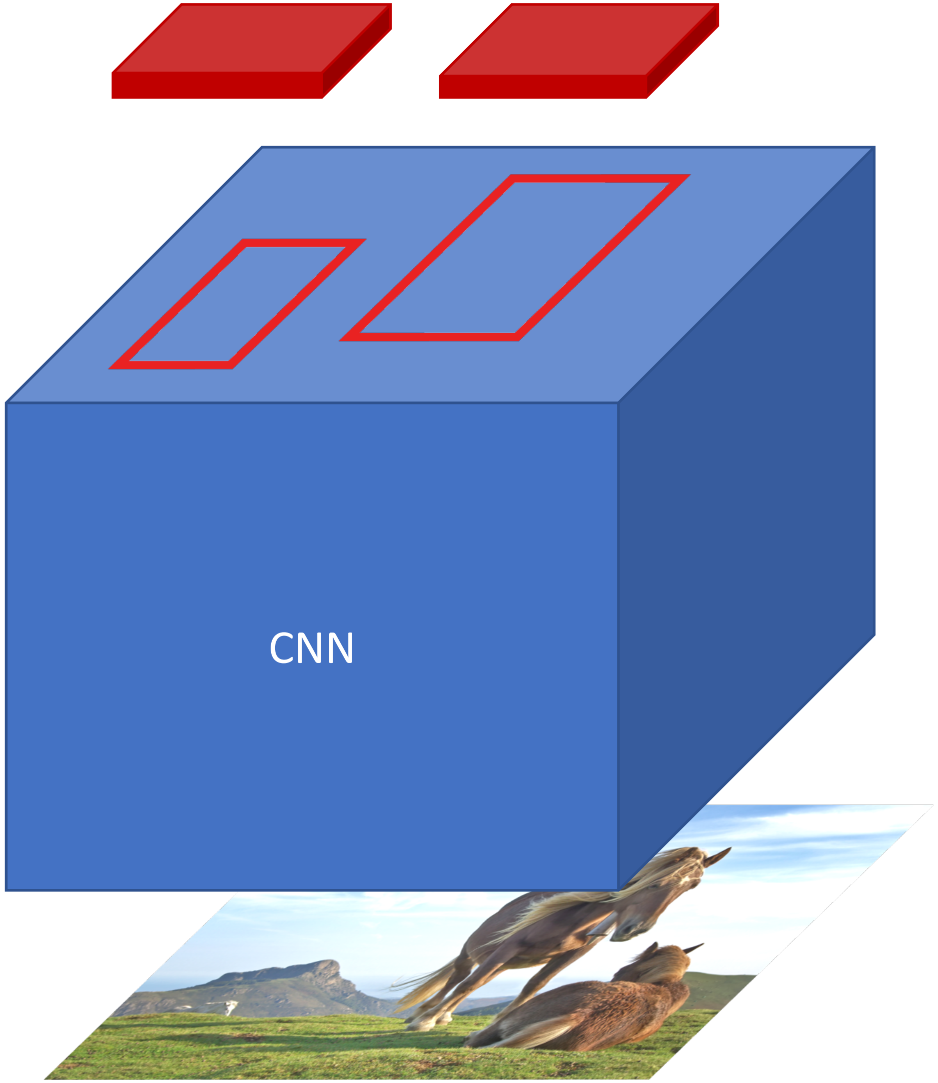
Speeding up R-CNN

- Each box requires a ConvNet run
- 2k boxes → 2000 times slower than classification!
- Can we share feature computation between the boxes?



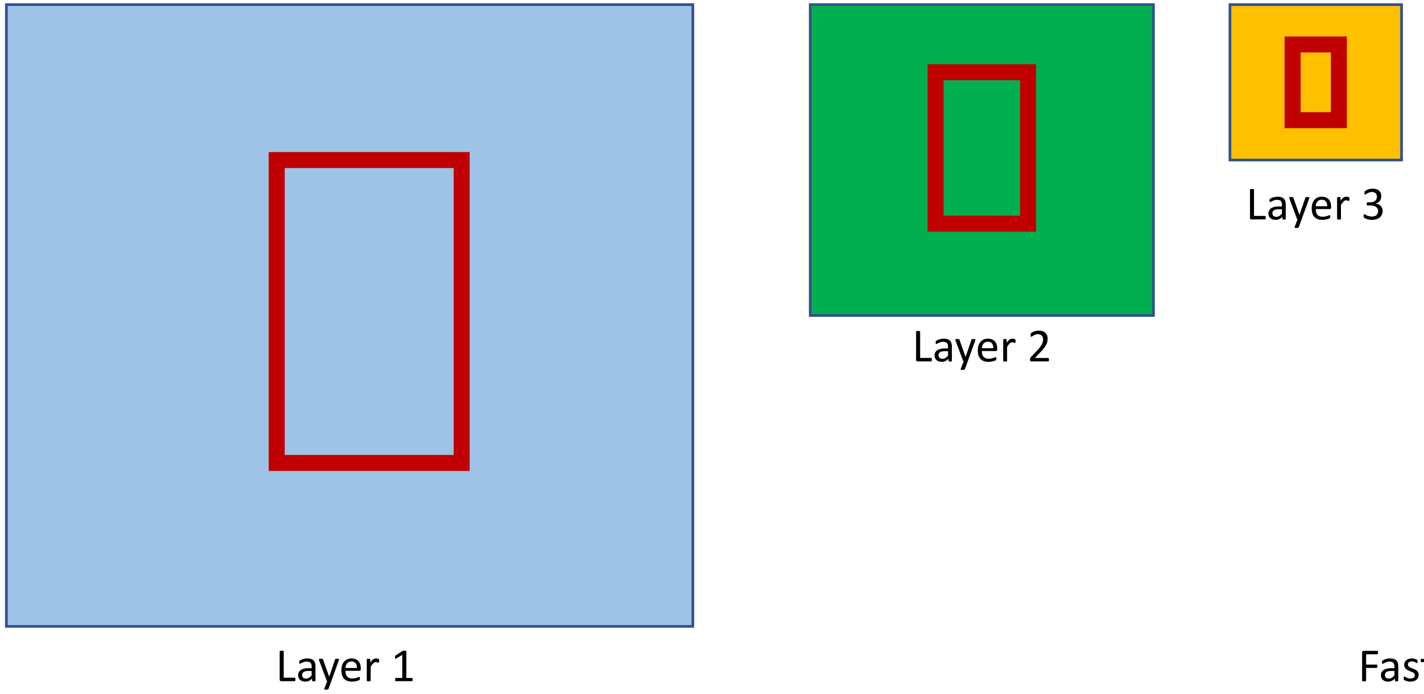
Speeding up R-CNN

- Each box requires a ConvNet run
- 2k boxes → 2000 times slower than classification!
- Can we share feature computation between the boxes?



ROI Pooling

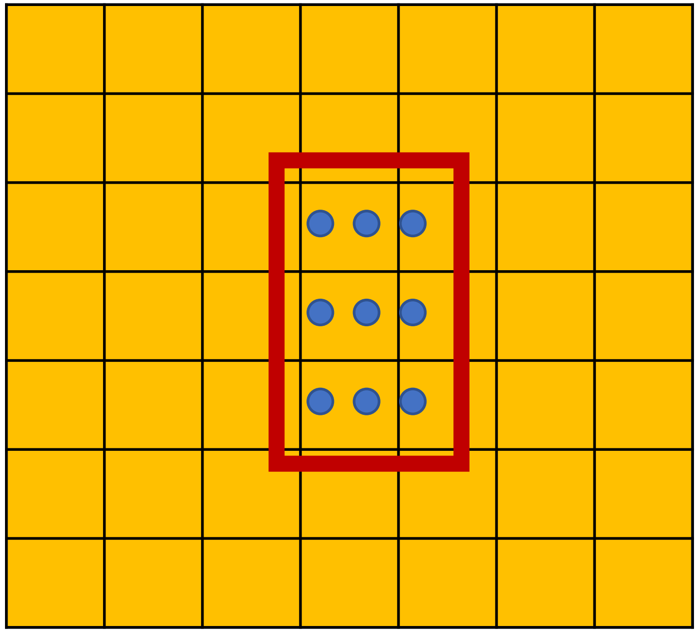
- How do we crop from a feature map?
- Step 1: Resize boxes to account for subsampling



Fast R-CNN. Ross Girshick. In ICCV 2015

Other details - ROI Align

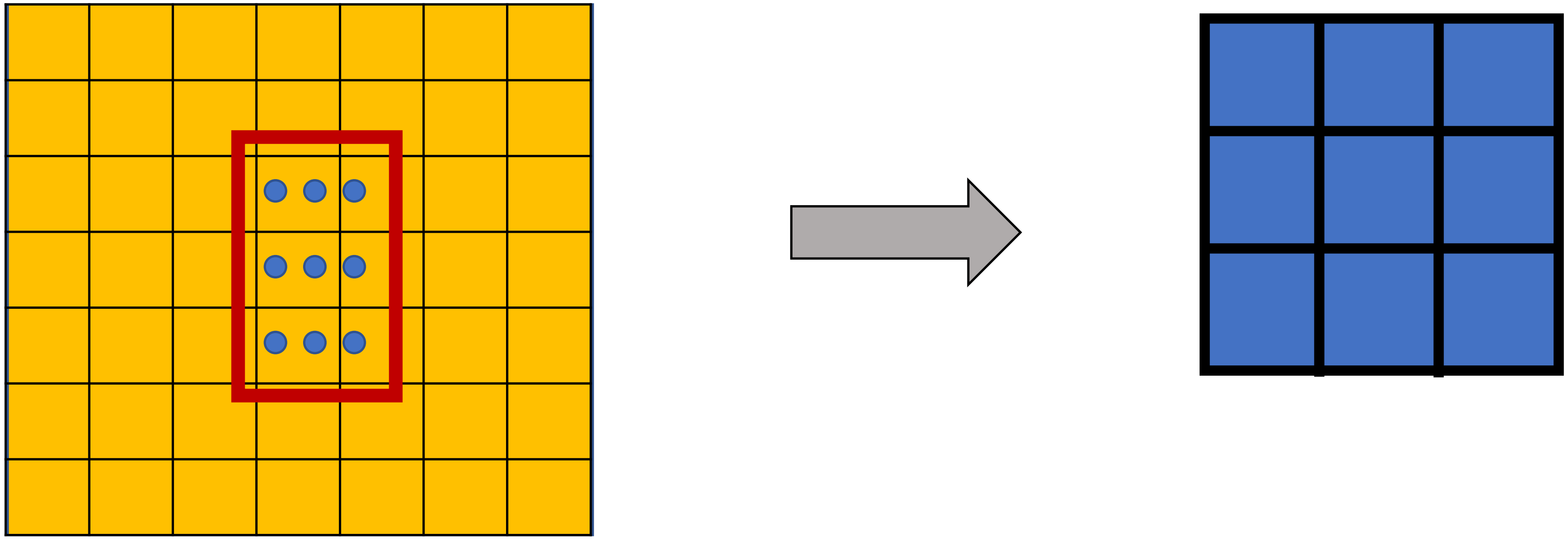
- Snapping box to grid introduces quantization artifacts
- Instead, use bilinear interpolation



Mask R-CNN. K. He, G. Gkioxari, P. Dollar, R. Girshick. In *ICCV* 2017.

Other details - ROI Align

- Snapping box to grid introduces quantization artifacts
- Instead, use bilinear interpolation



Mask R-CNN. K. He, G. Gkioxari, P. Dollar, R. Girshick. In *ICCV* 2017.

Fast R-CNN

	Fast R-CNN	R-CNN
Train time (h)	9.5	84
Speedup	8.8x	1x
Test time / image	0.32s	47.0s
Speedup	146x	1x
mean AP	66.9	66.0

Fast R-CNN

- Bottleneck remaining (not included in time):
  - Object proposal generation

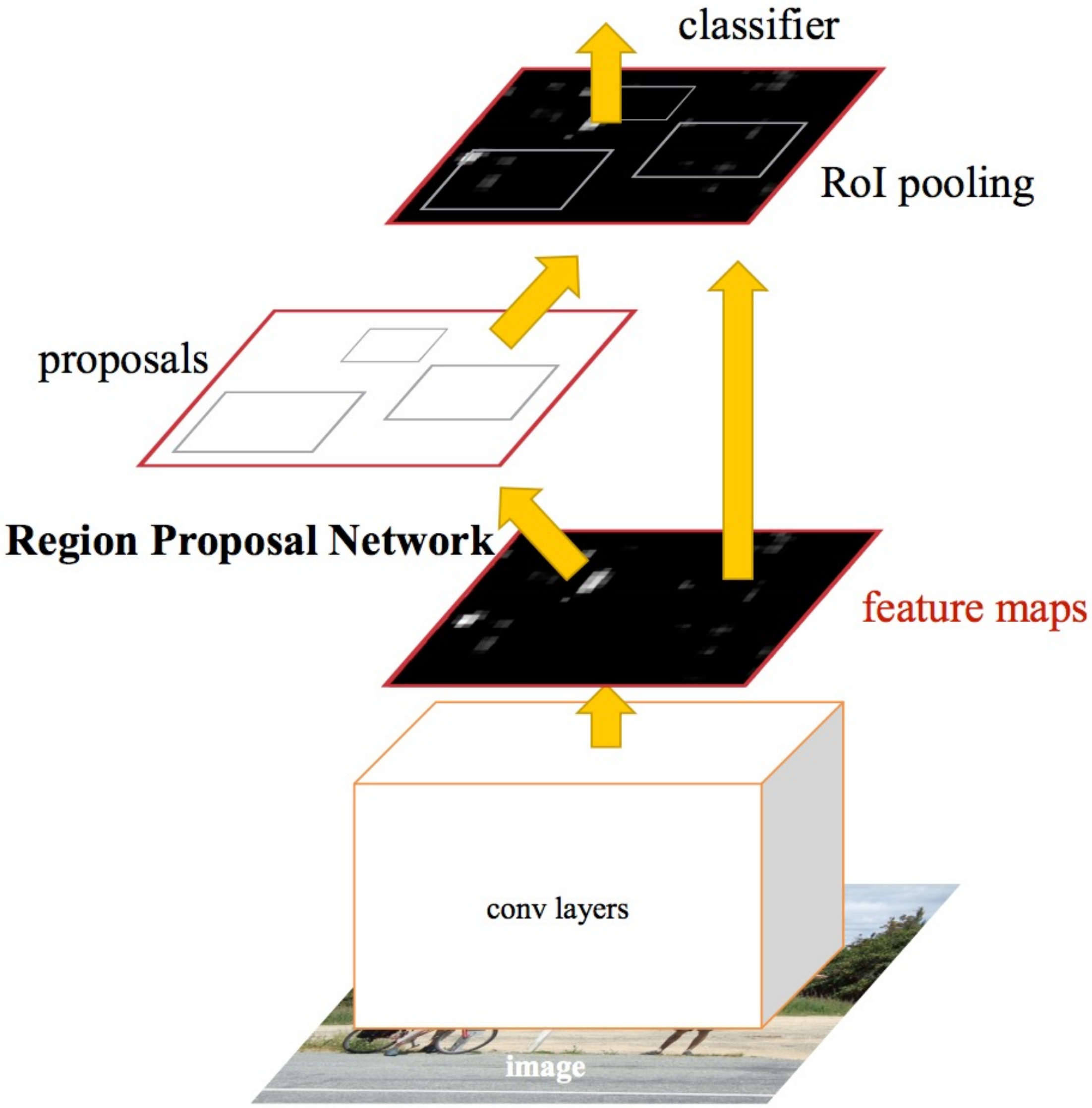
- Slow
  - Requires segmentation
  - O(1s) per image

Fast*er* R-CNN

- Can we produce *object proposals* from convolutional networks?
- A change in intuition
  - Instead of using grouping
  - Recognize likely objects?
- For every possible box, score if it is likely to correspond to an object
- *Cascade*

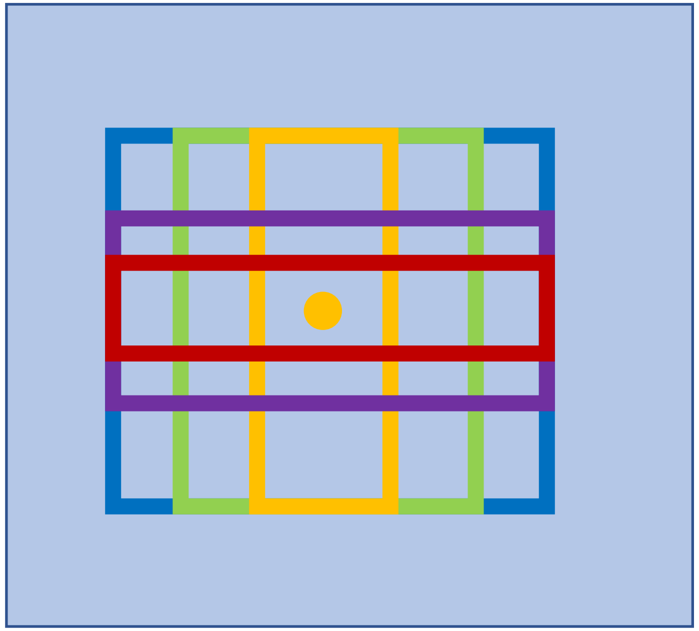
Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. S. Ren, K. He, R. Girshick, J. Sun. In *NIPS* 2015.

Faster R-CNN


classifier
RoI pooling
proposals
Region Proposal Network
feature maps
conv layers
image

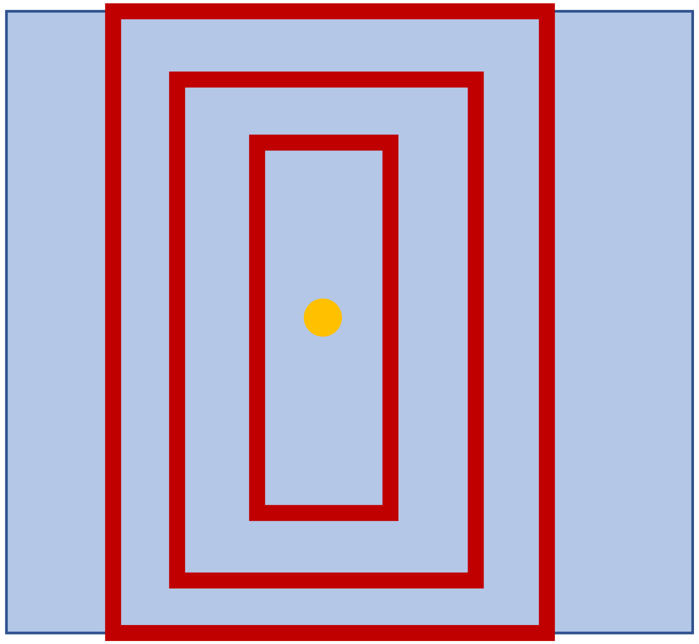
Faster R-CNN

- At each location, consider boxes of many different sizes and aspect ratios
- If k such sizes, use simple convolutional layer to output k ”objectness scores”



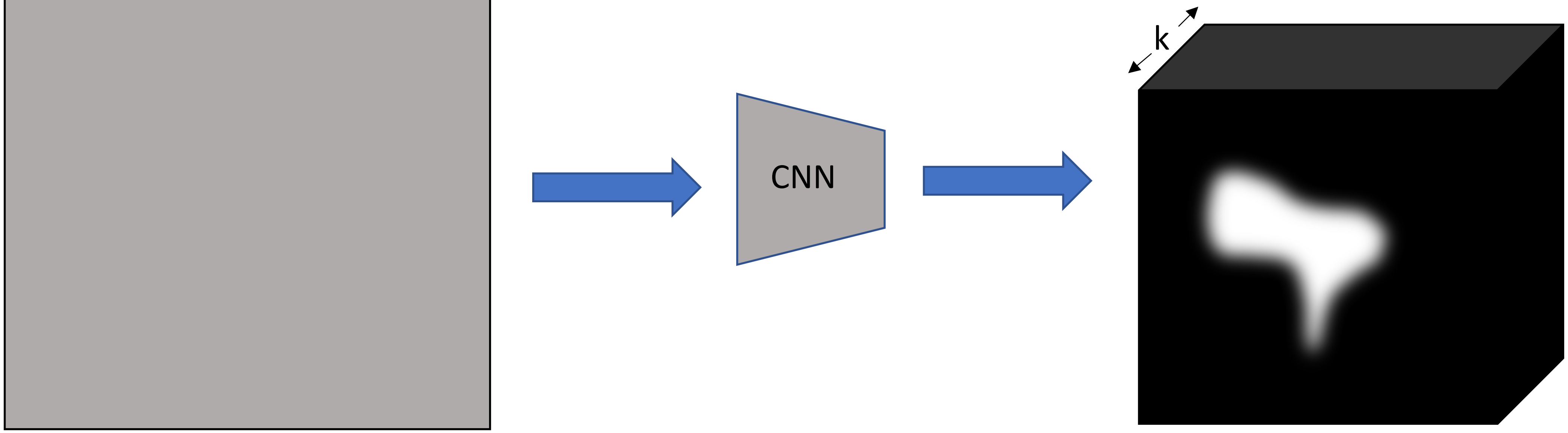
Faster R-CNN

- At each location, consider boxes of many different sizes and aspect ratios
- If k such sizes, use simple convolutional layer to output k ”objectness scores”



Faster R-CNN

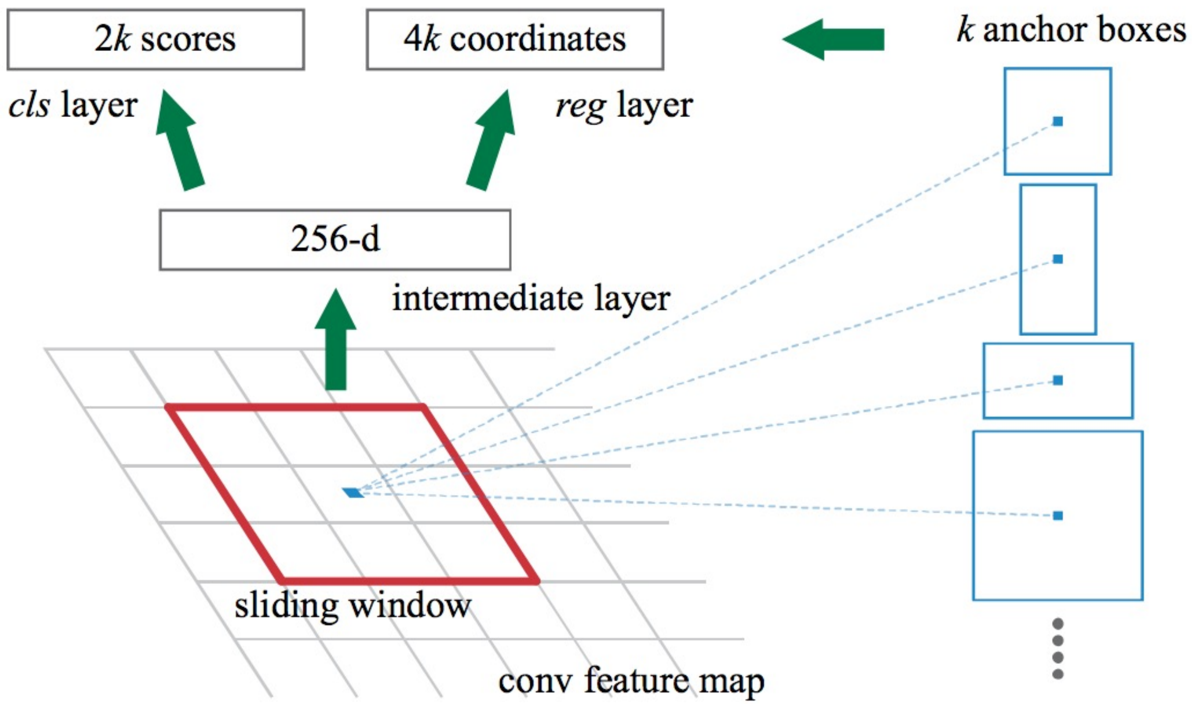
- At each location, consider boxes of many different sizes and aspect ratios
- If k such sizes, use simple convolutional layer to output k ”objectness scores”



Faster R-CNN

- At each location, consider boxes of many different sizes and aspect ratios

- Produce scores for each box using a convolution
- Also produce regressed coordinates using another convolution



Faster R-CNN

- s scales $\times a$ aspect ratios = sa anchor boxes
- Use convolutional layer on top of filter map to produce sa scores
- Another convolution to produce $4sa$ bounding box offsets
- Pick top few boxes as proposals

Faster R-CNN

Method	mean AP (PASCAL VOC)
Fast R-CNN	65.7
Faster R-CNN	67.0

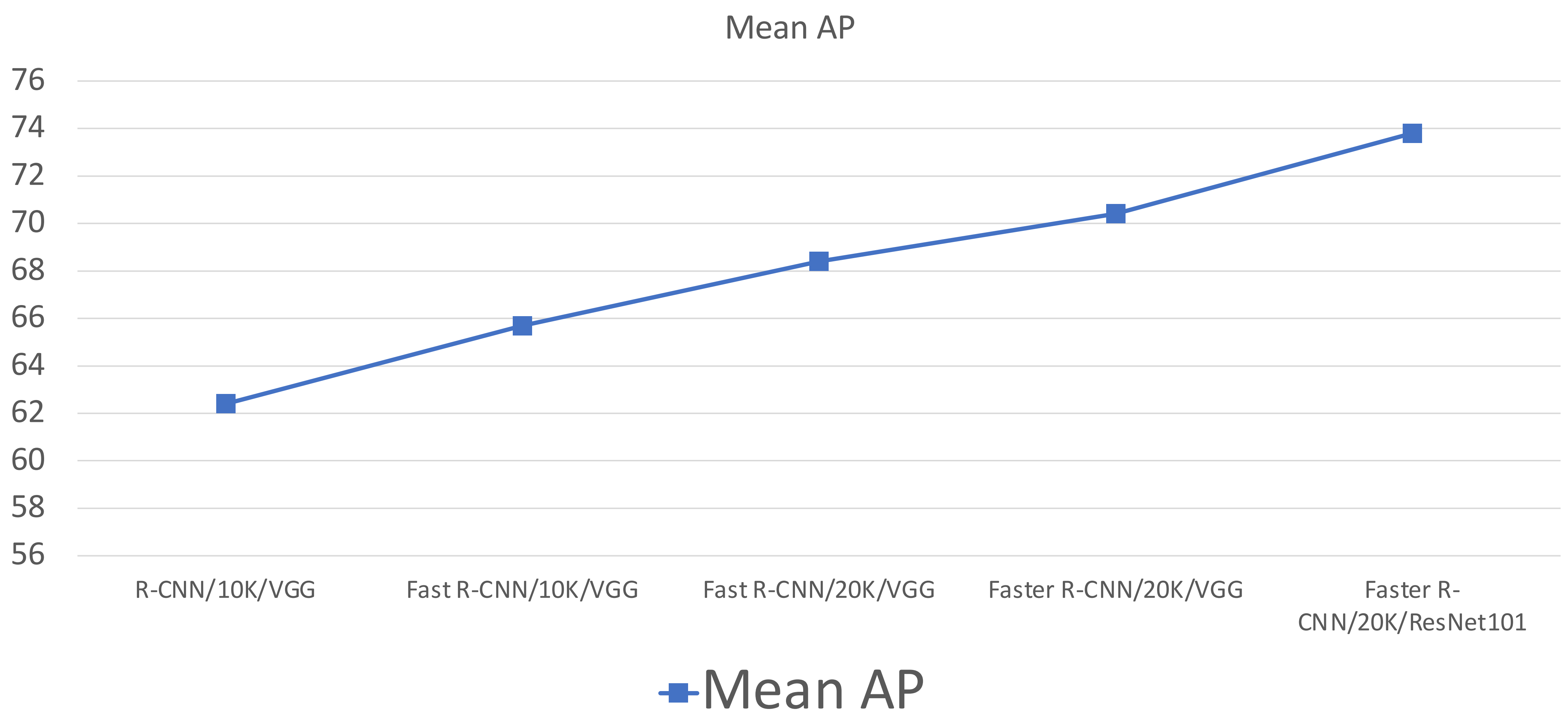
Impact of Feature Extractors

ConvNet	mean AP (PASCAL VOC)
VGG	70.4
ResNet 101	73.8

Impact of Additional Data

Method	Training data	mean AP (PASCAL VOC 2012 Test)
Fast R-CNN	VOC 12 Train (10K)	65.7
Fast R-CNN	VOC07 Trainval + VOC 12 Train	68.4
Faster R-CNN	VOC 12 Train (10K)	67.0
Faster R-CNN	VOC07 Trainval + VOC 12 Train	70.4

The R-CNN family of detectors



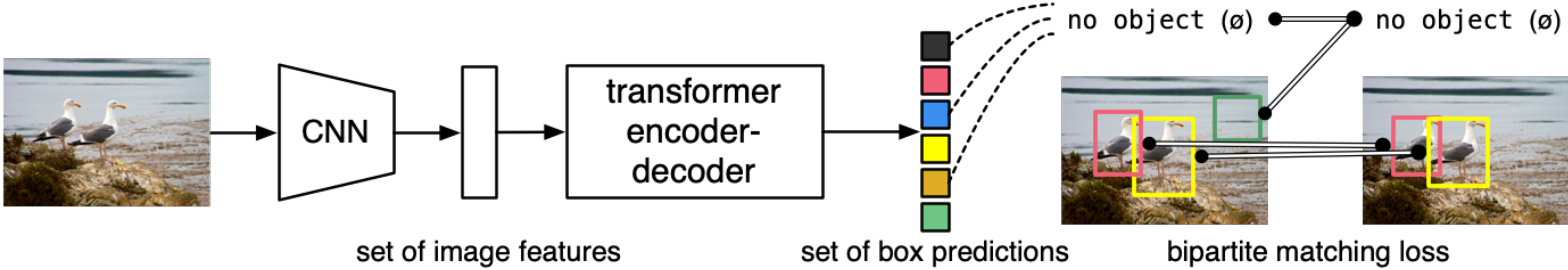
SSD (Single Shot Detector)

- Why go through separate proposals?
- Directly produce class-specific scores at each location for every scale and aspect ratio
  - s scales * a aspects * c classes = sac scores per location

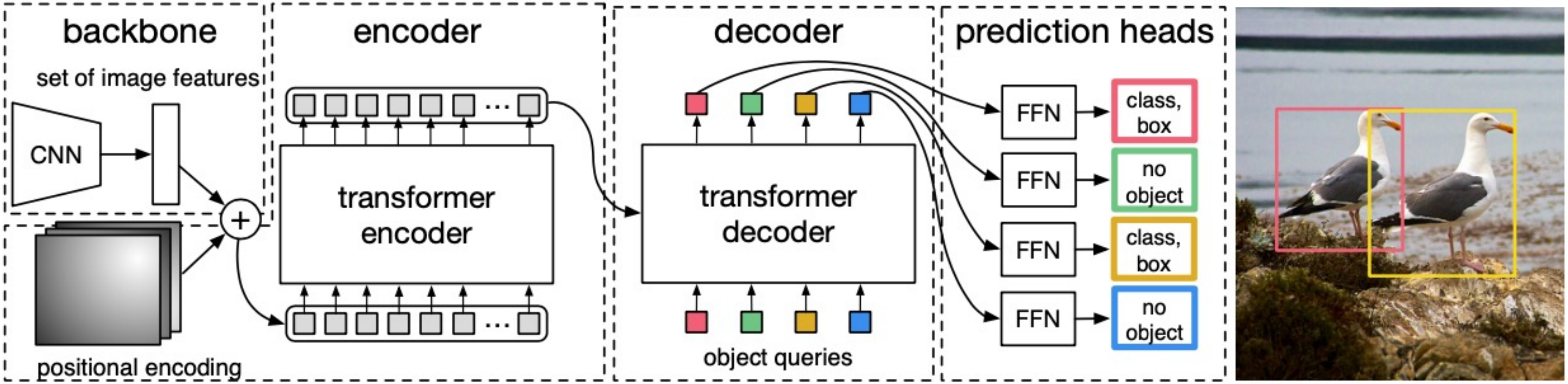
SSD: Single Shot MultiBox Detector. Wei Liu , Dragomir Anguelov , Dumitru Erhan , Christian Szegedy , Scott Reed , Cheng-Yang Fu , Alexander C. Berg. In ECCV, 2016

Transformer-based detectors - DETR

- Transformers process sets
- Detection produces sets



DETR

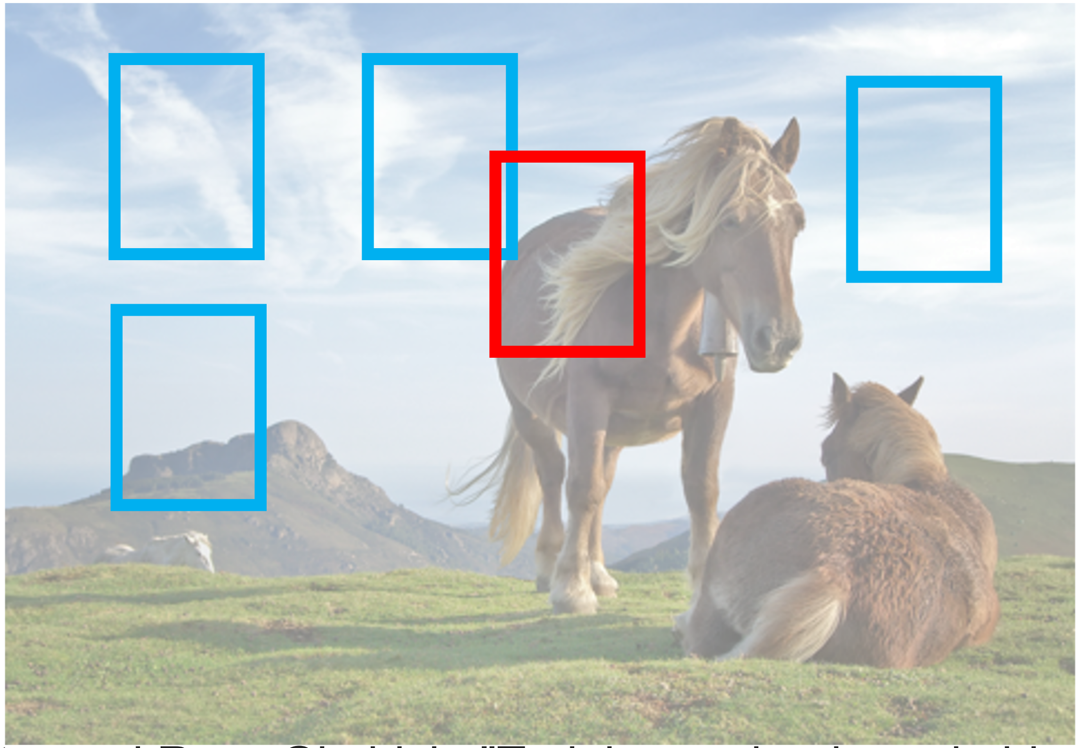

backbone
set of image features
CNN
positional encoding
encoder
transformer encoder
decoder
transformer decoder
object queries
prediction heads
FFN
FFN
FFN
FFN
class, box
no object
class, box
no object

Dealing with class imbalance

- Single stage detectors have extreme class imbalance. How to deal with this?
- Hard negative mining
- Focal loss

Hard negative mining

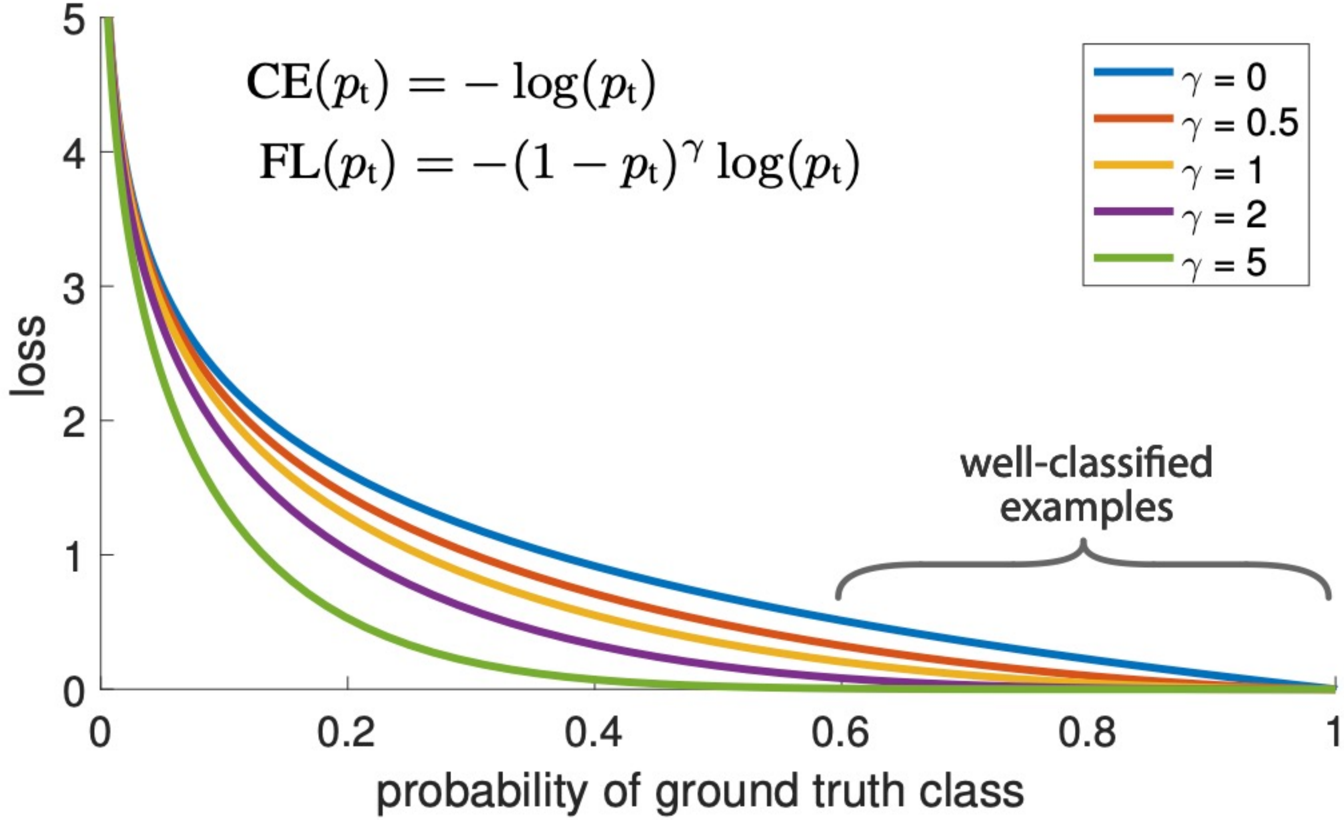
- Key issue: training swamped by easy negative examples with low loss
- Idea: only optimize on "hard" examples: negative examples with high score



Shrivastava, Abhinav, Abhinav Gupta, and Ross Girshick. "Training region-based object detectors with online hard example mining." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Focal loss

- Idea: weigh low loss examples even less

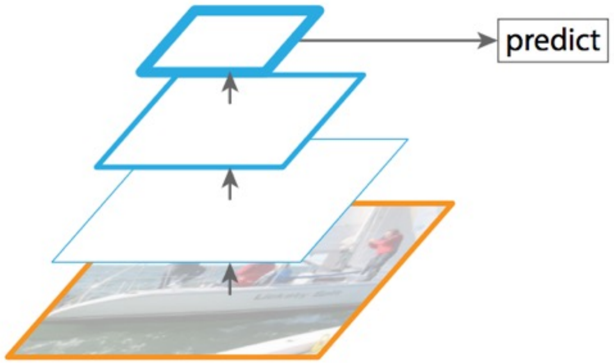


Lin, T. Y., Goyal, P., Girshick, R., He, K., & Dollár, P. (2017). Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 2980-2988).

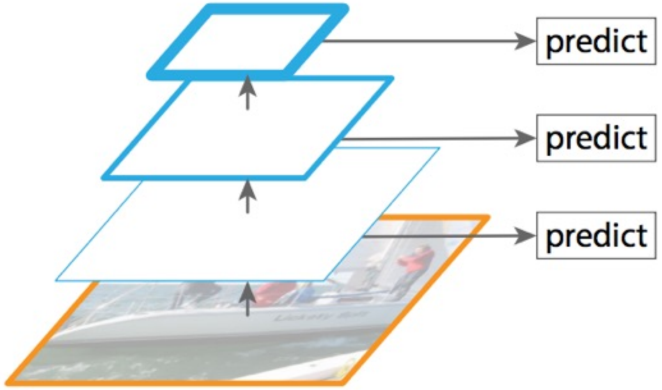
Detecting small objects

- Small objects get low resolution features

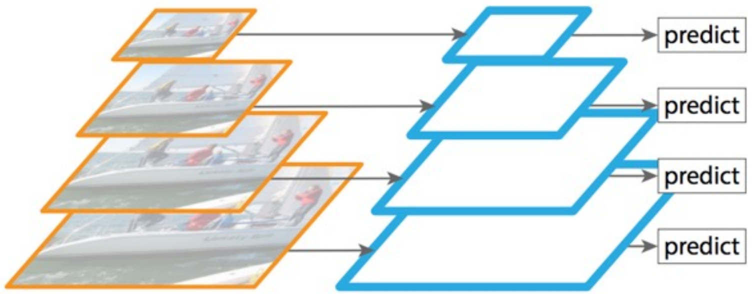
Feature pyramid networks



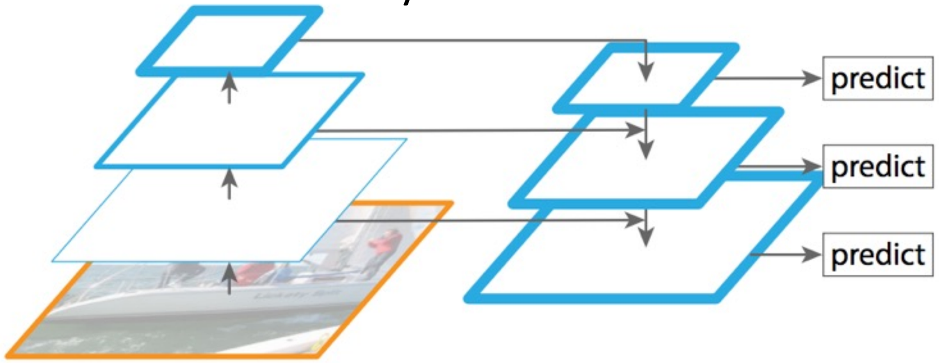
Standard detection



Detection using multiple layers



Detection on image pyramid



Detection using feature pyramid layers

Lin, Tsung-Yi, et al. "Feature Pyramid Networks for Object Detection." *CVPR*. Vol. 1. No. 2. 2017.

Feature pyramid networks

Faster R-CNN	proposals	feature	head	lateral?	top-down?	AP@0.5	AP	AP_s	AP_m	AP_l
(*) baseline from He *et al.* [16]†	RPN, C_4	C_4	conv5			47.3	26.3	-	-	-
(a) baseline on conv4	RPN, C_4	C_4	conv5			53.1	31.6	13.2	35.6	**47.1**
(b) baseline on conv5	RPN, C_5	C_5	*2fc*			51.7	28.0	9.6	31.9	43.1
(c) **FPN**	RPN, $\{P_k\}$	$\{P_k\}$	*2fc*	✓	✓	**56.9**	**33.9**	**17.8**	**37.7**	45.8

Lin, Tsung-Yi, et al. "Feature Pyramid Networks for Object Detection." *CVPR*. Vol. 1. No. 2. 2017.

Other kinds of object detectors

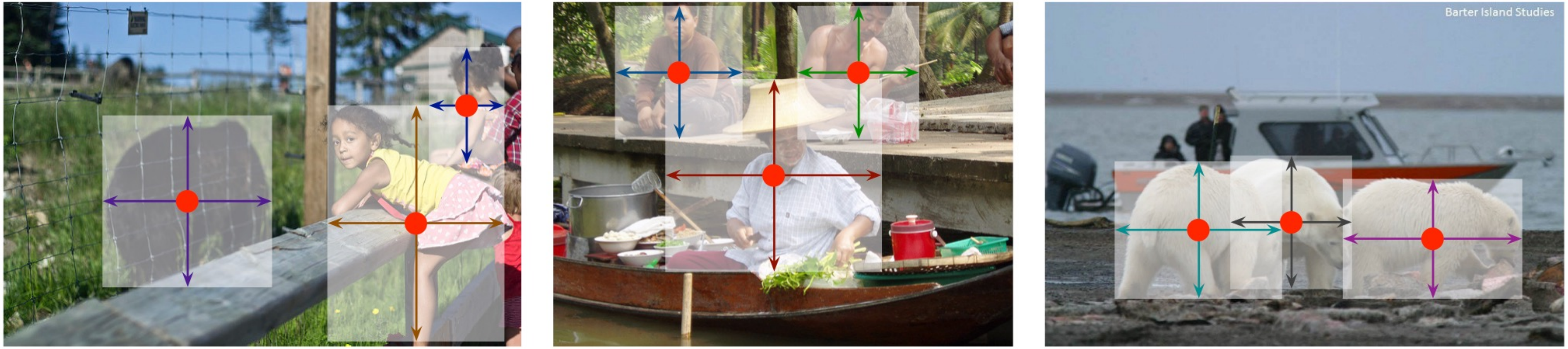


Figure 2: We model an object as the center point of its bounding box. The bounding box size and other object properties are inferred from the keypoint feature at the center. Best viewed in color.

Zhou, Xingyi, Dequan Wang, and Philipp Krähenbühl. "Objects as points." *arXiv preprint arXiv:1904.07850* (2019).