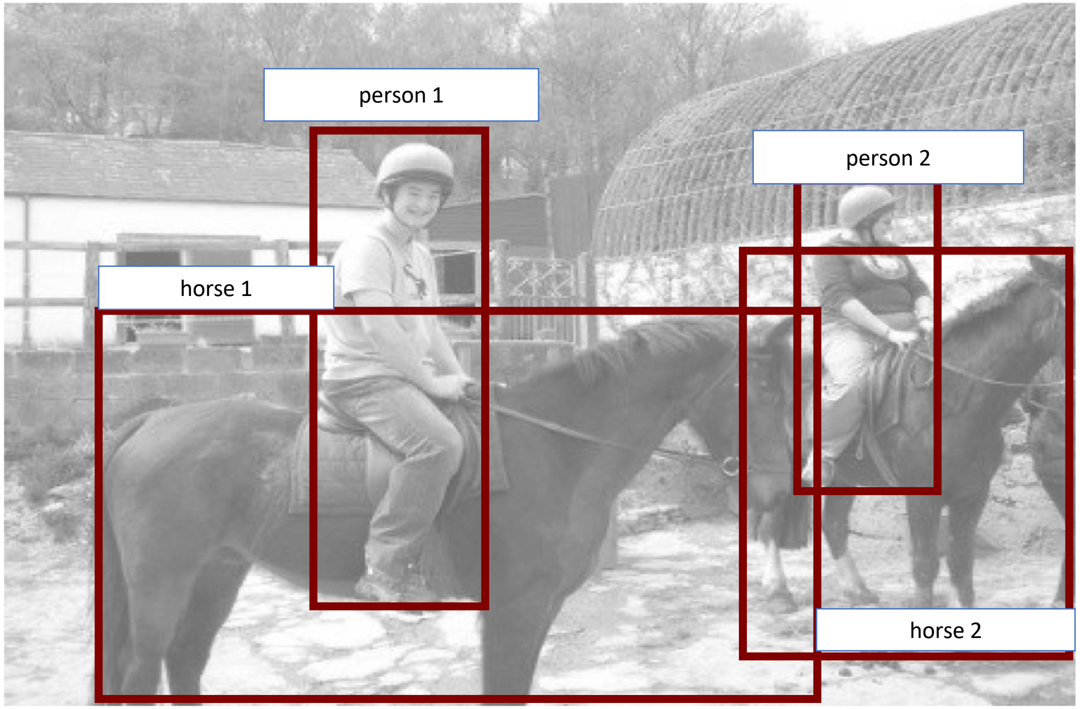
Object detection

The Task



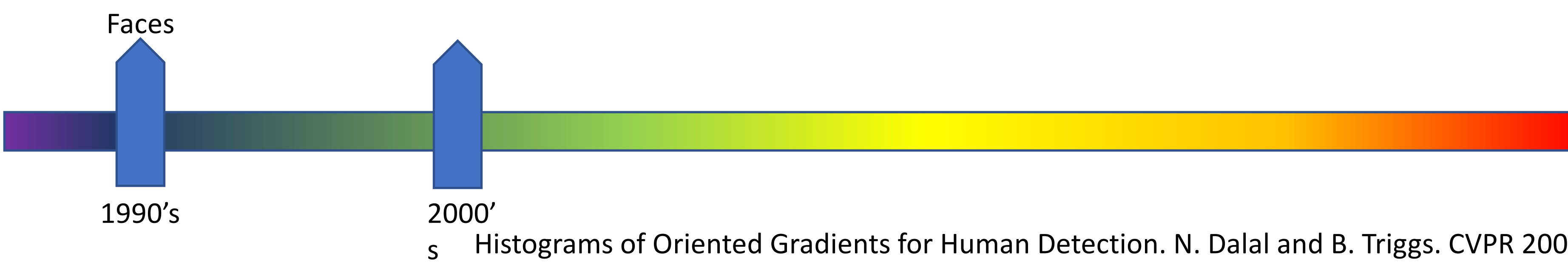
Datasets

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

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

Pedestrians

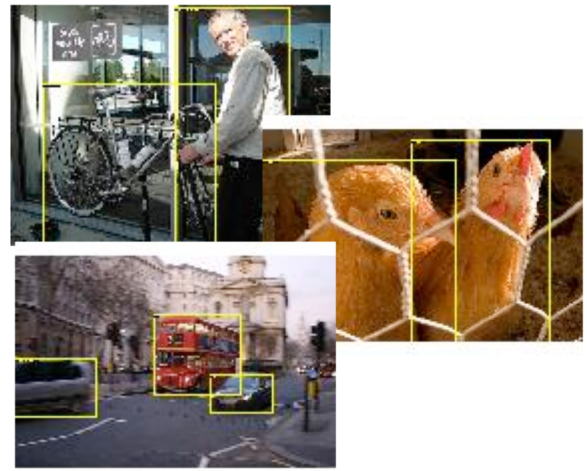
- One category: pedestrians
- Slight pose variations and small distortions
- Partial occlusions

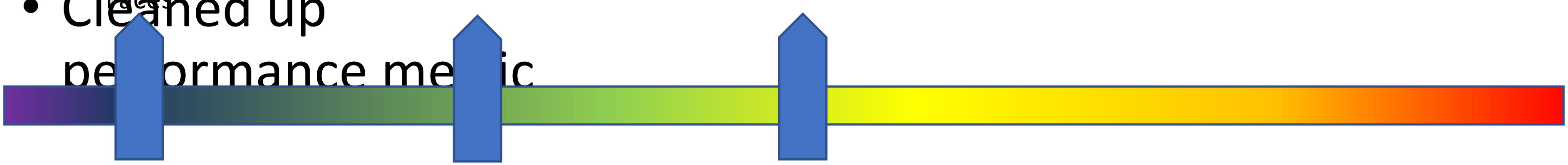


Histograms of Oriented Gradients for Human Detection. N. Dalal and B. Triggs. CVPR 2005

PASCAL VOC

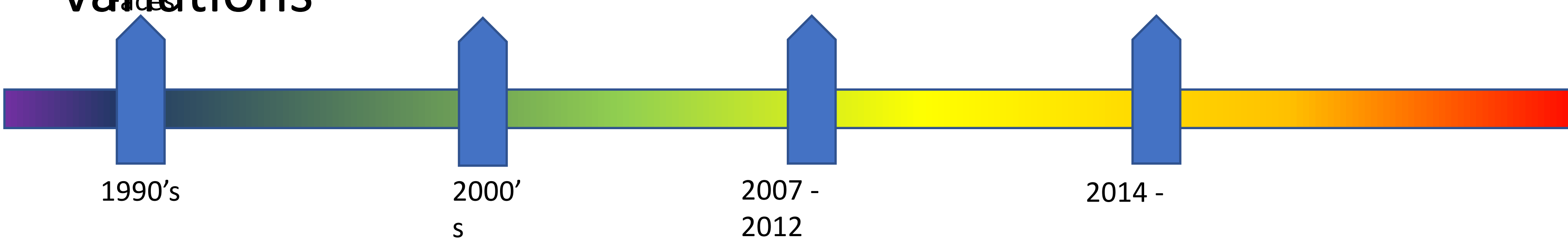
- 20 categories
- 10K images
- Large pose variations, heavy occlusions
- Generic scenes
- Cleaned up performance metric





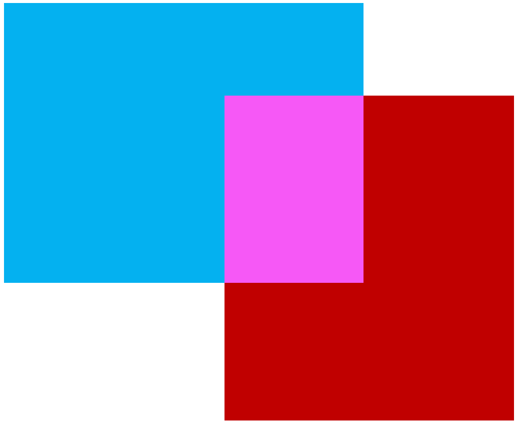
Coco

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



Evaluation metric

Matching detections to ground truth



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

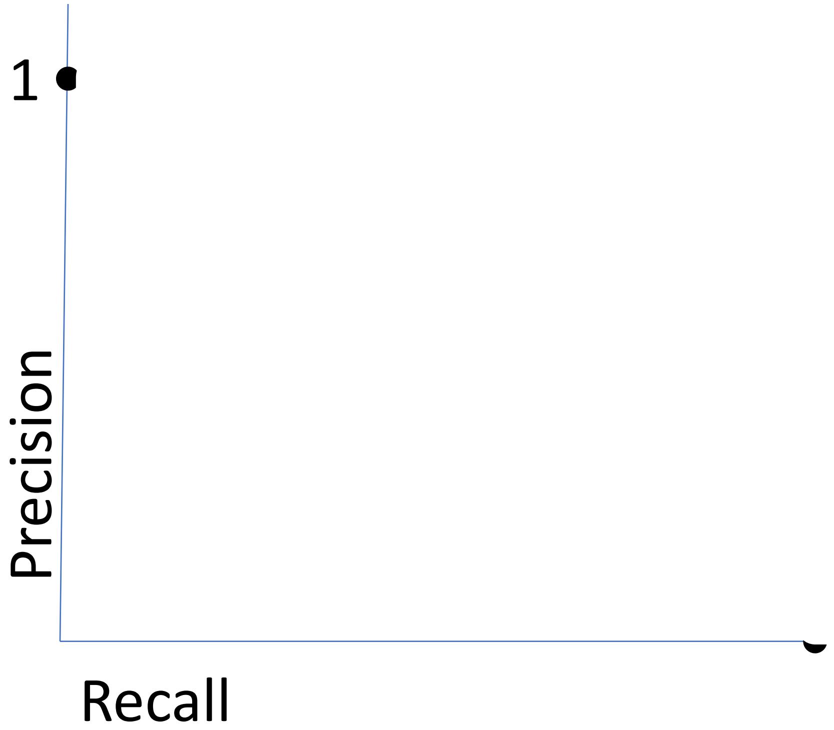
Matching detections to ground truth

- 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

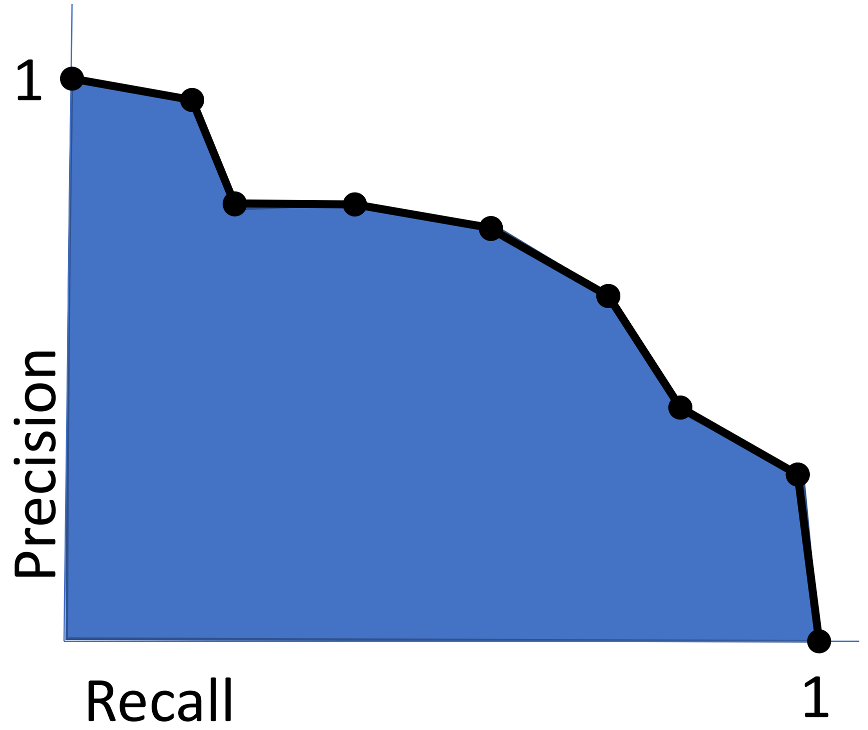
Tradeoff between precision and recall

- 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

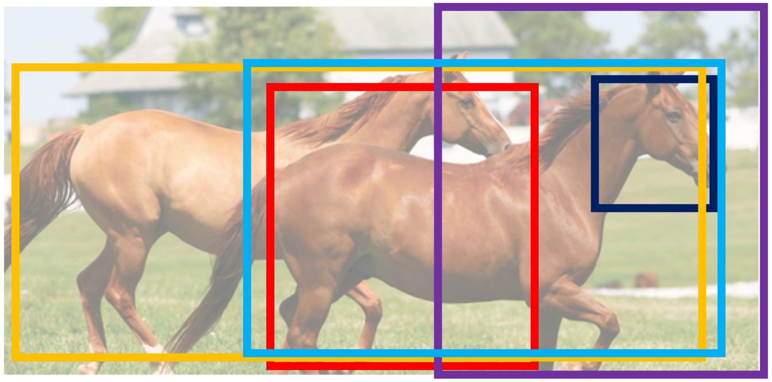
- AP marks detections with overlap > 50% as correct
- But may need better localization
- *Average* AP across multiple overlap thresholds
- Confusingly, still called average precision
- Introduced in COCO

Mean and category-wise AP

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

Why is detection hard(er)?

- Precise localization



Why is detection hard(er)?

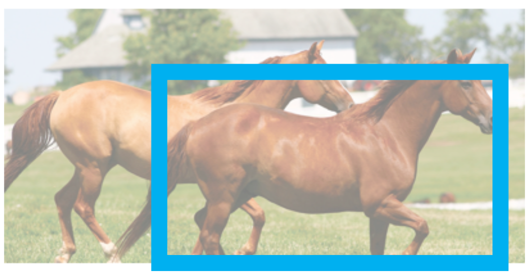
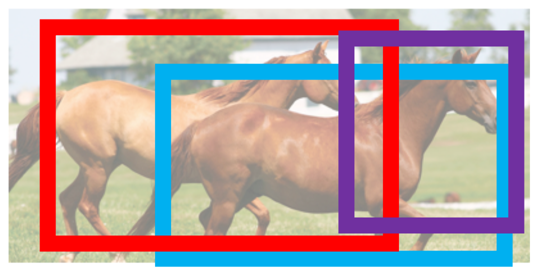
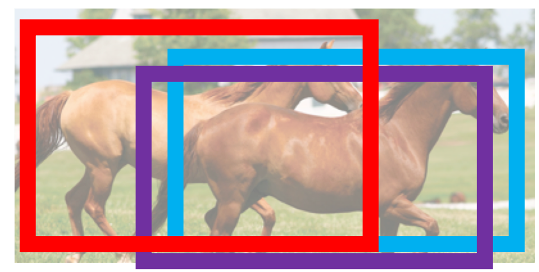
- Much larger impact of pose

Why is detection hard(er)?

- Occlusion makes localization difficult

Why is detection hard(er)?

- Counting

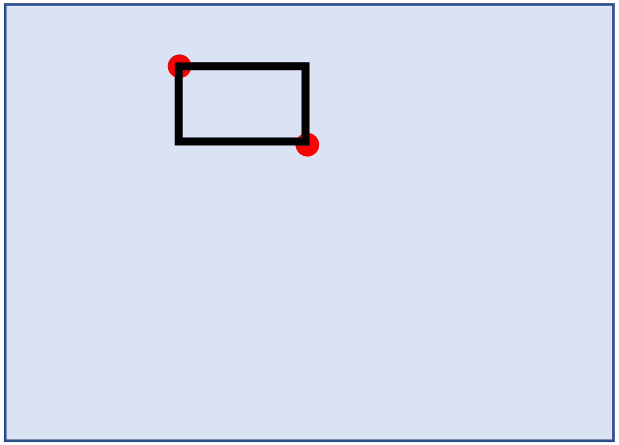
  

Why is detection hard(er)?

- Small objects

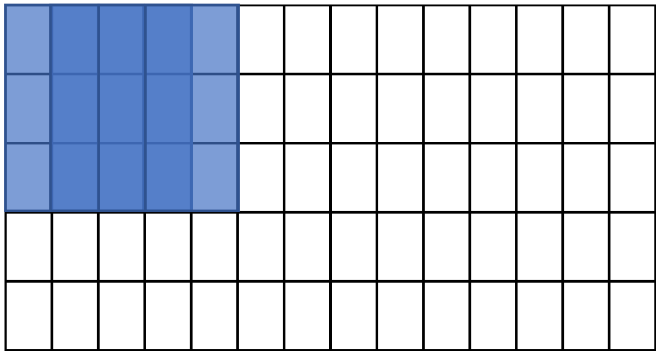
Detection as classification

- Run through every possible box and classify
- 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!



Idea 1: scanning window

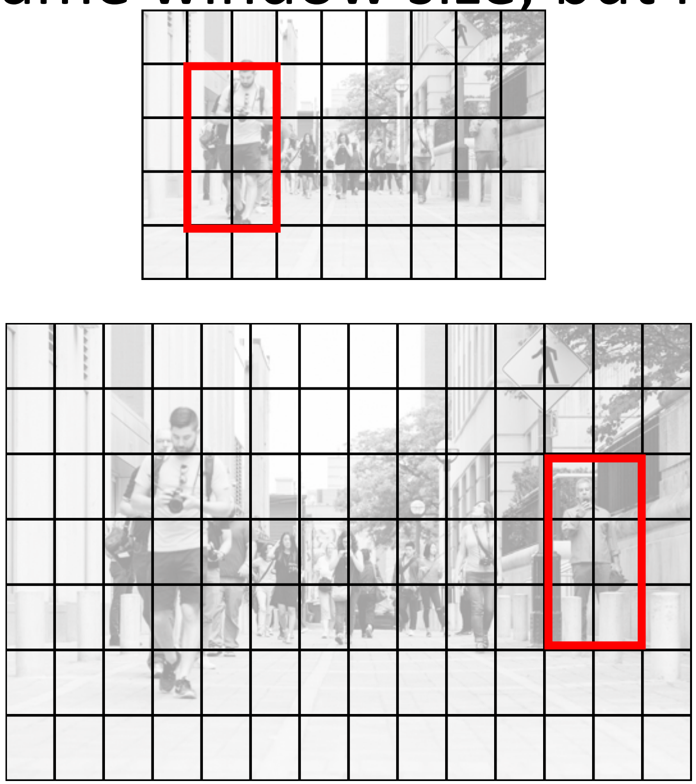
- Fix size
  - Can take a few different sizes
- Fixed stride
- Convolution with a filter
  - Classic: compute HOG features over entire image



Dealing with scale

Dealing with scale

- Use same window size, but run on *image pyramid*



Issues

- Classifies millions of boxes, so must be very fast
- Needs ultra-fine sampling of scales and object sizes, can still miss outlier sizes

Scanning window results on PASCAL

	VOC 2007	VOC 2010
DPM v5 (Girshick et al. 2011)	33.7%	29.6%

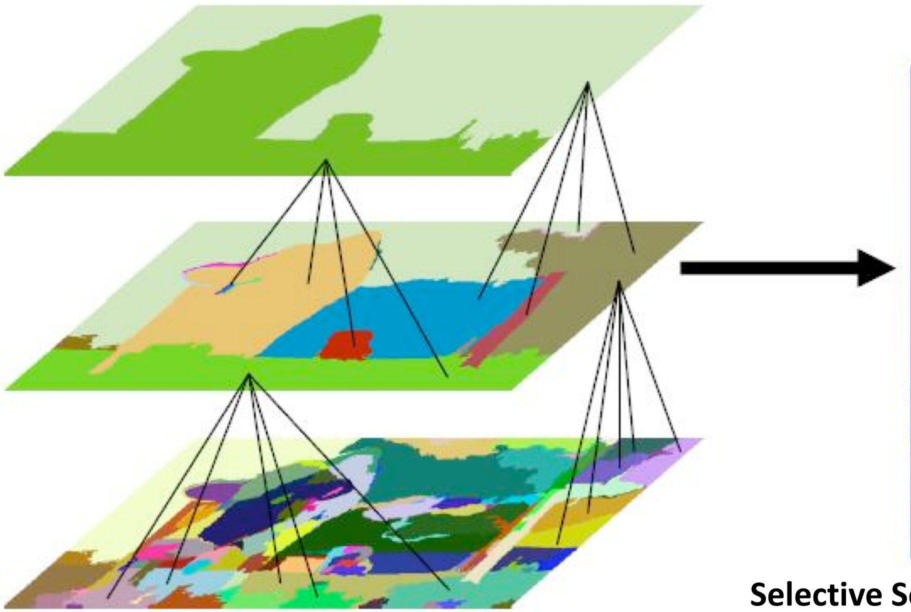
Reference systems

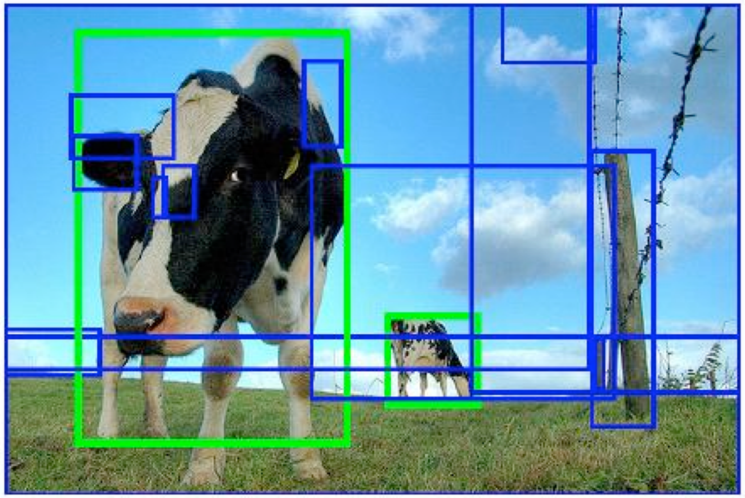
metric: mean average precision (higher is better)

Slide credit : Ross Girshick

Idea 2: Object proposals

- Use segmentation to produce ~5K candidates





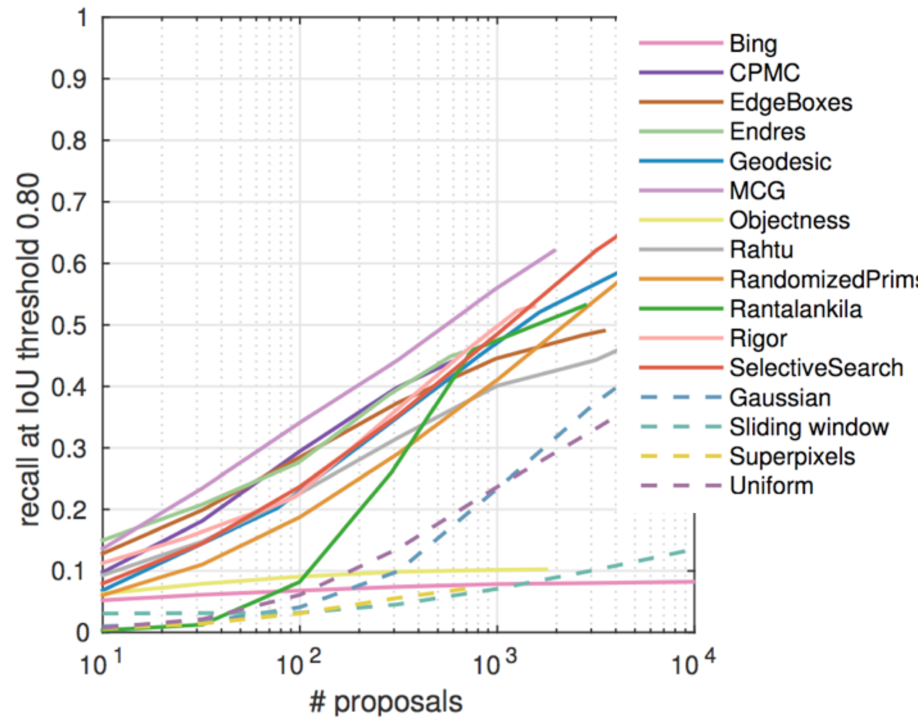
Selective Search for Object Recognition
J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, A. W. M. Smeulders
In International Journal of Computer Vision 2013.

Idea 2: object proposals

- Many different segmentation algorithms (k-means on color, k-means on color+position, N-cuts….)
- Many hyperparameters (number of clusters, weights on edges)
- Try everything!
    - Every cluster is a candidate object
    - Thousands of segmentations -> thousands of candidate objects

Idea 2: Object proposals

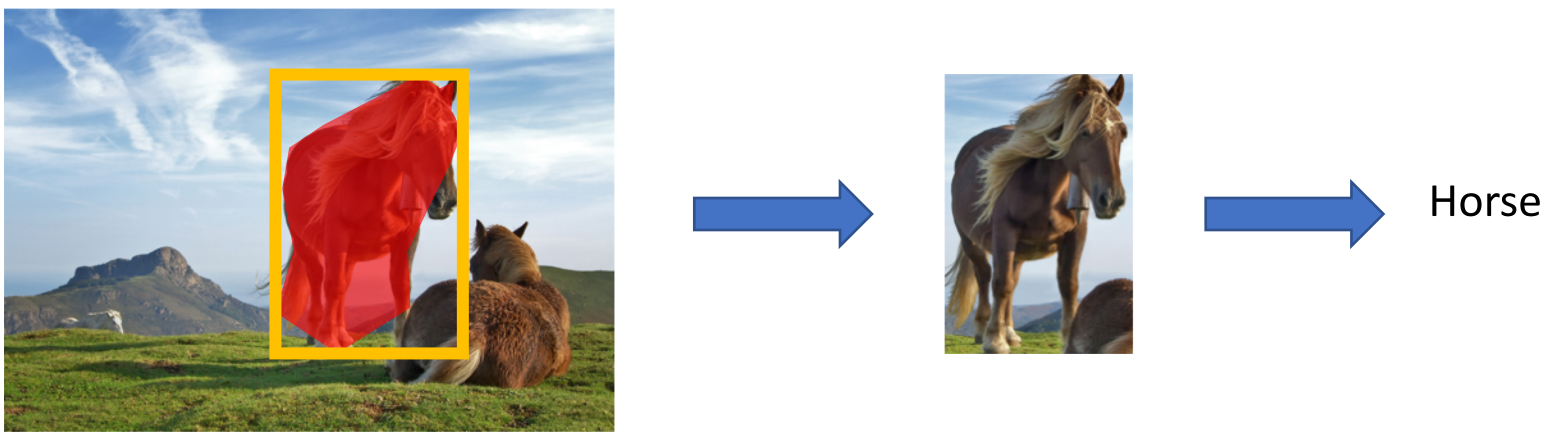
- Tens of ways of generating candidates ("proposals")
- What fraction of ground truth objects have proposals near them?



What makes for effective detection proposals? J. Hosang, R. Benenson, P. Dollar, B. Schiele. In TPAMI

What do we do with proposals?

- Each proposal is a group of pixels
- Take tight fitting box and *classify it*
- *Can leverage any image classification approach*



Proposal methods results

	VOC 2007	VOC 2010
DPM v5 (Girshick et al. 2011)	33.7%	29.6%
UVA sel. search (Uijlings et al. 2013)		35.1%

Reference systems

metric: mean average precision (higher is better)

Slide credit : Ross Girshick

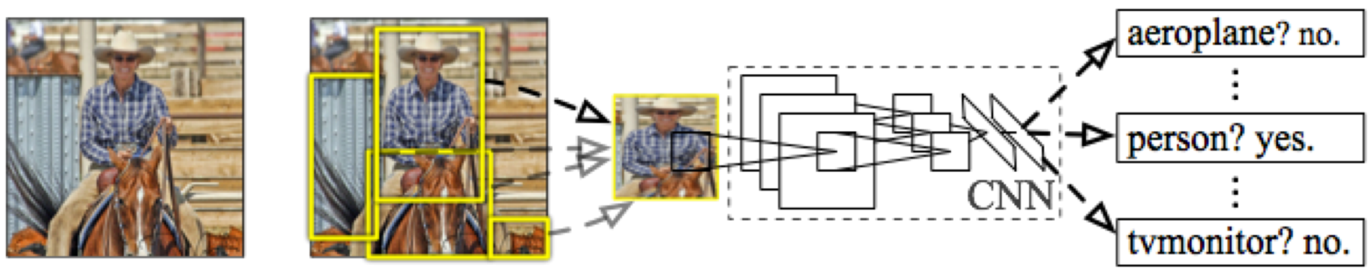
Proposal methods results

	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

Slide credit : Ross Girshick

R-CNN: Regions with CNN features



Input image

Extract region proposals (~2k / image)

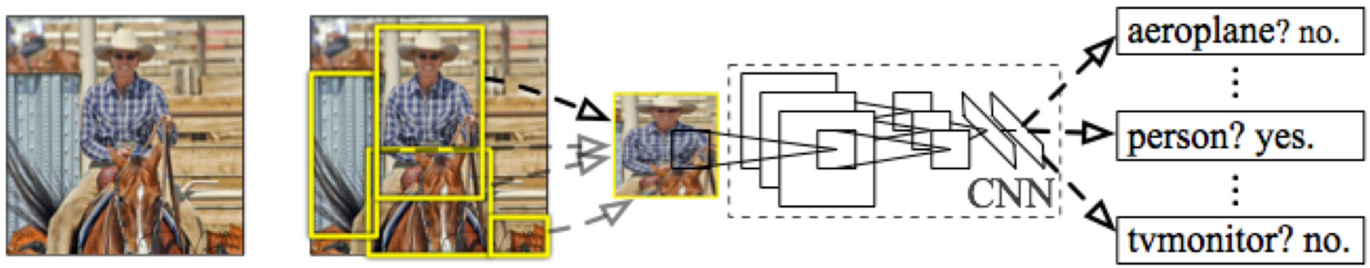
Compute CNN features

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

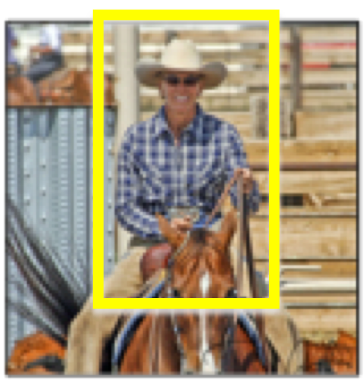
R-CNN at test time: Step 2



Input image

Extract region proposals (~2k / image)

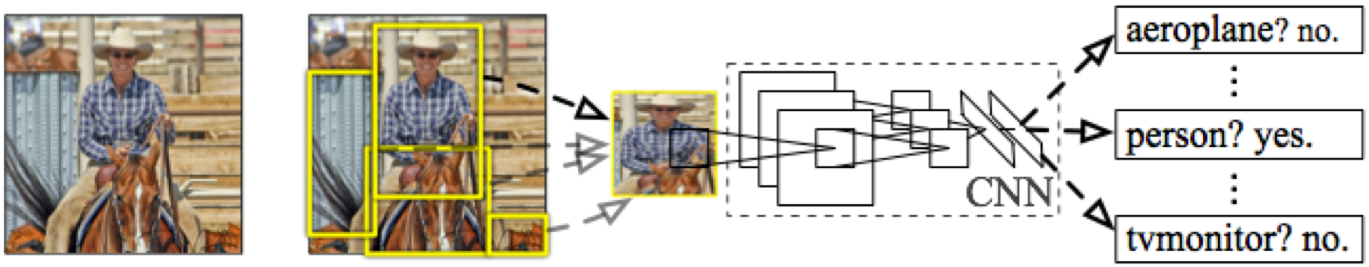
Compute CNN features



a. Crop

Slide credit : Ross Girshick

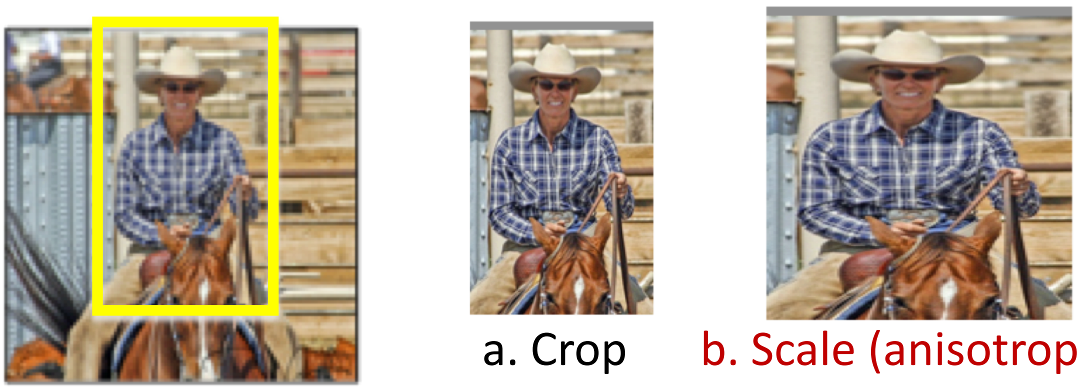
R-CNN at test time: Step 2



Input image

Extract region proposals (~2k / image)

Compute CNN features



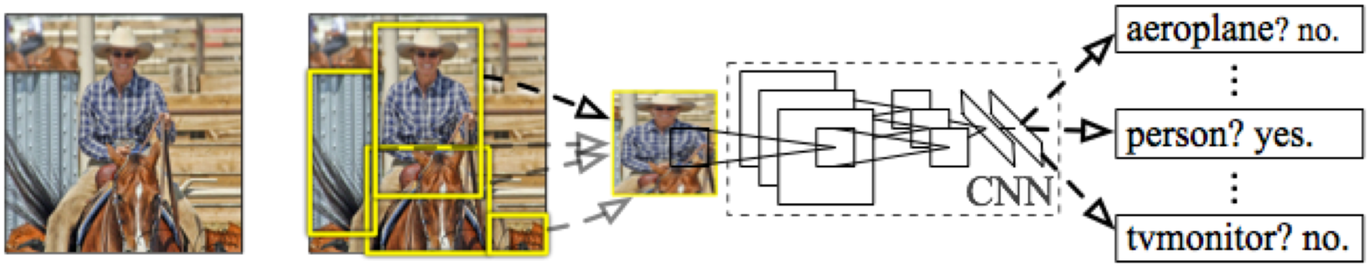
227 x 227

a. Crop

b. Scale (anisotropic)

Slide credit : Ross Girshick

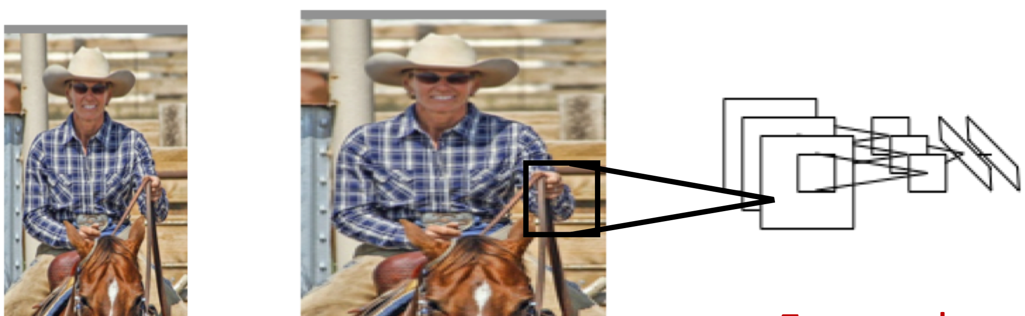
R-CNN at test time: Step 2



Input image

Extract region proposals (~2k / image)

Compute CNN features



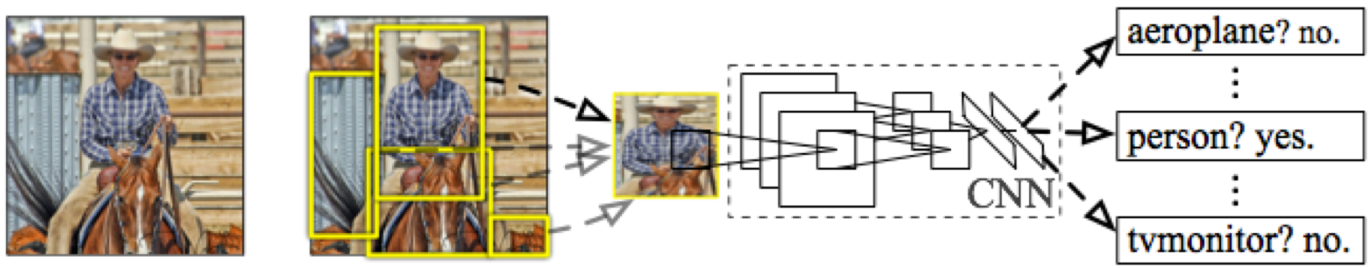
1. Crop

b. Scale (anisotropic)

c. Forward propagate
Output: “fc_7” features

Slide credit : Ross Girshick

R-CNN at test time: Step 3

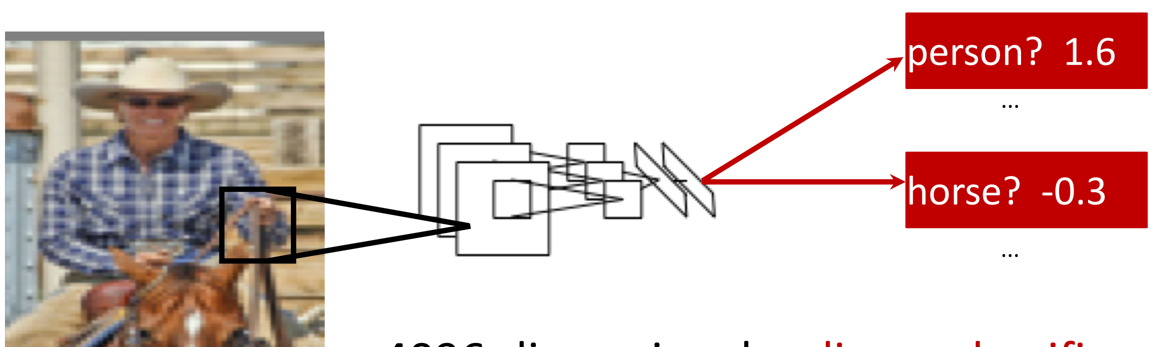


Input image

Extract region proposals (~2k / image)

Compute CNN features

Classify regions



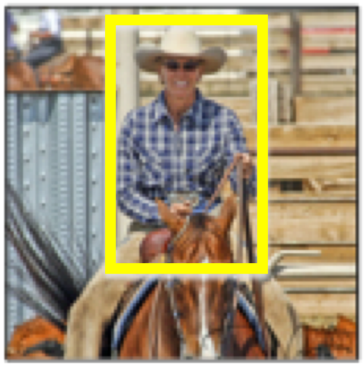
Warped proposal

4096-dimensional fc_7 feature vector

linear classifiers (SVM or softmax)

Slide credit : Ross Girshick

Step 4: Object proposal refinement



Linear regression

on CNN features

Original
proposal

Predicted
object bounding box

Bounding-box regression

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%

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%

Slide credit : Ross Girshick

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

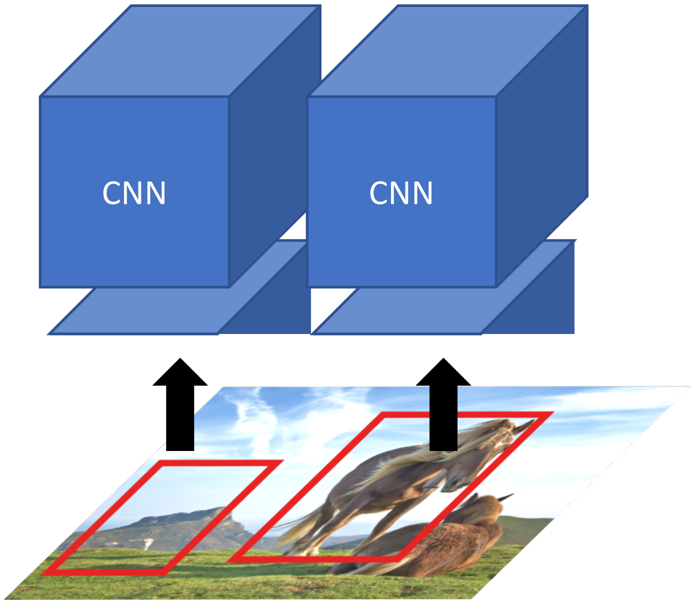
Other details - Non-max suppression

How do we deal with multiple detections on the same object?

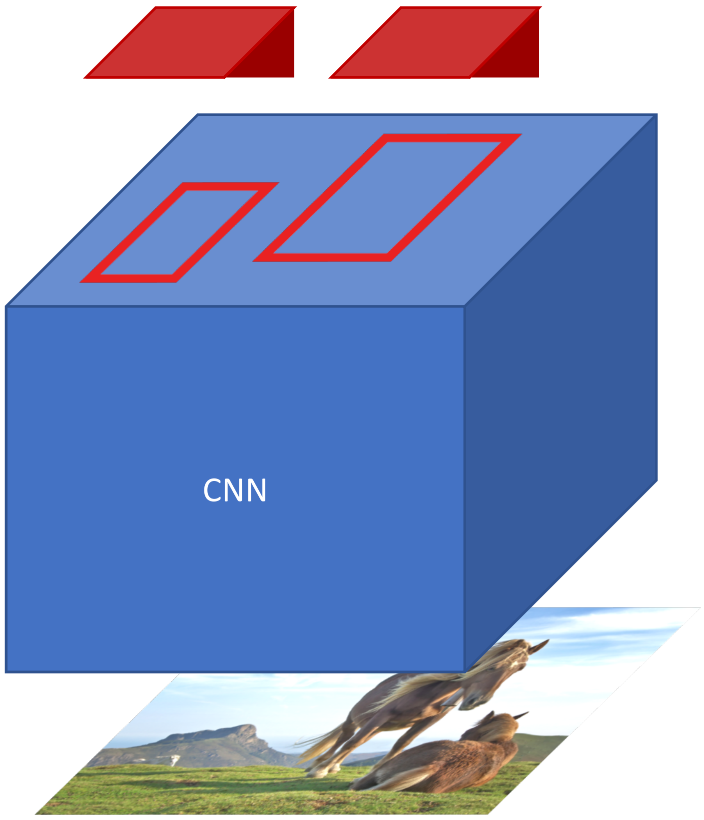
Other details - Non-max suppression

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

Speeding up R-CNN

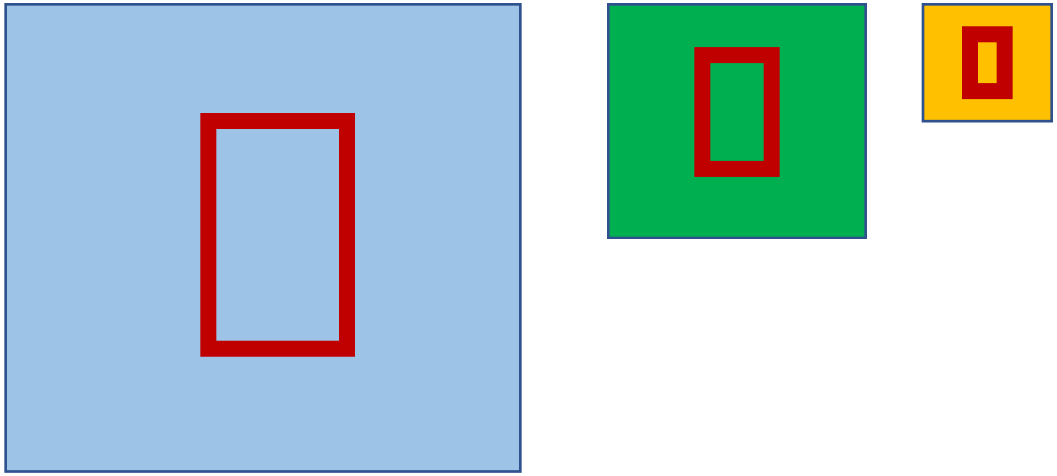


Speeding up R-CNN



ROI Pooling

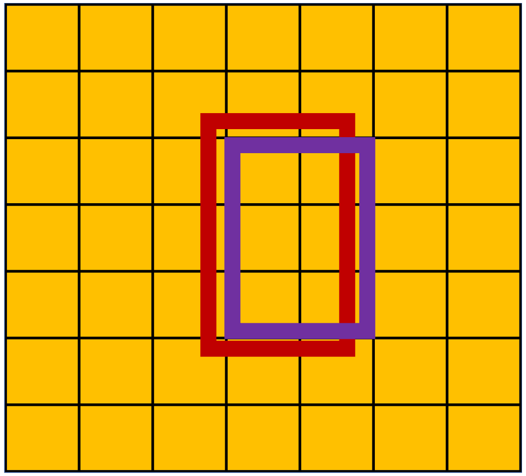
- How do we crop from a feature map?
- Step 1: Resize boxes to account for subsampling



Fast R-CNN. Ross Girshick. In ICCV 2015

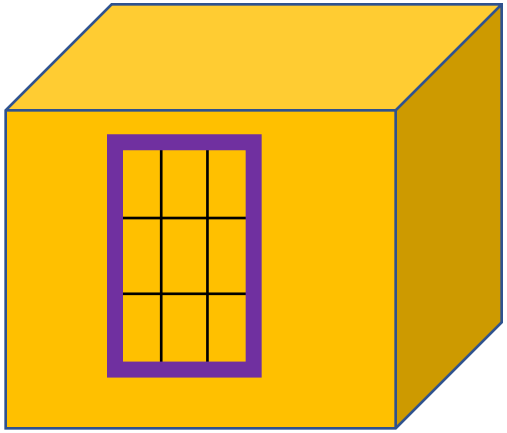
ROI Pooling

- How do we crop from a feature map?
- Step 2: Snap to feature map grid



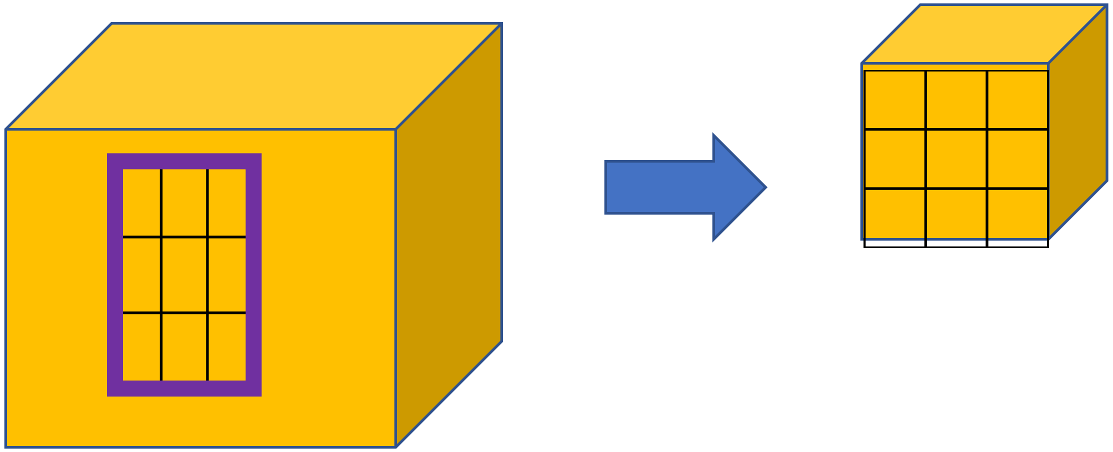
ROI Pooling

- How do we crop from a feature map?
- Step 3: Place a grid of fixed size



ROI Pooling

- How do we crop from a feature map?
- Step 4: Take max in each cell



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

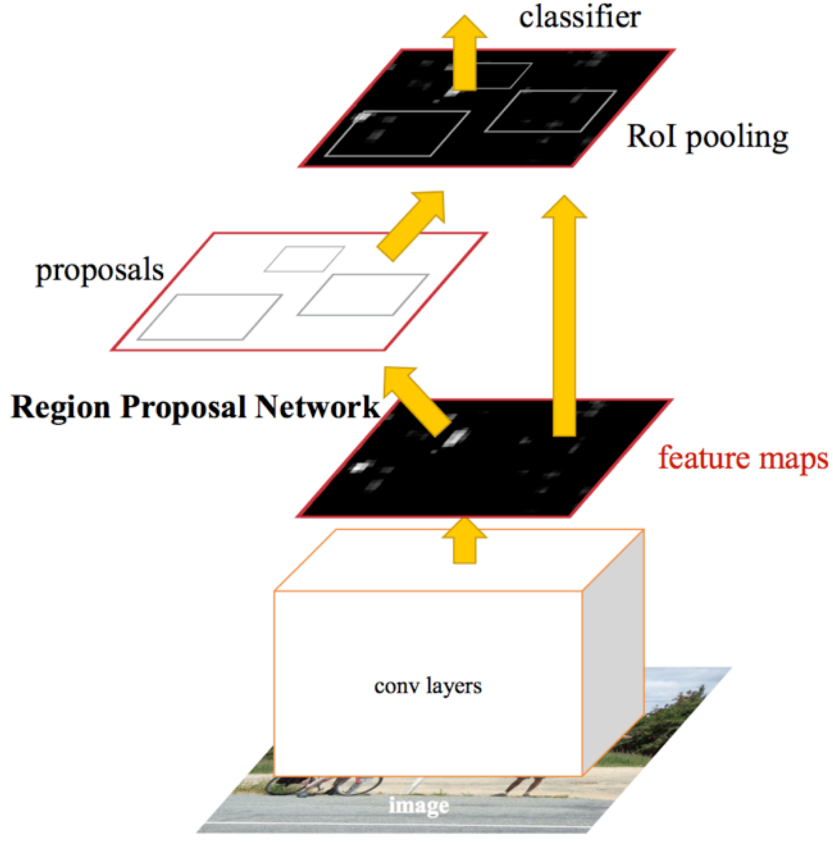
- Slow
  - Requires segmentation
  - O(1s) per image

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

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

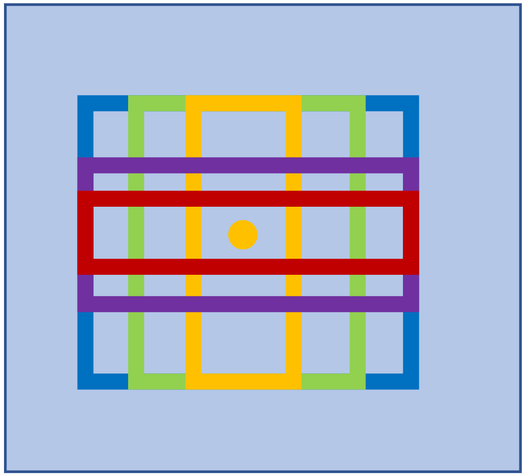
Faster R-CNN



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

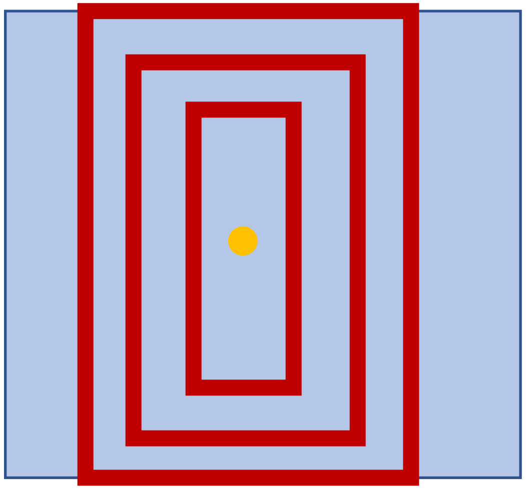
Faster R-CNN

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



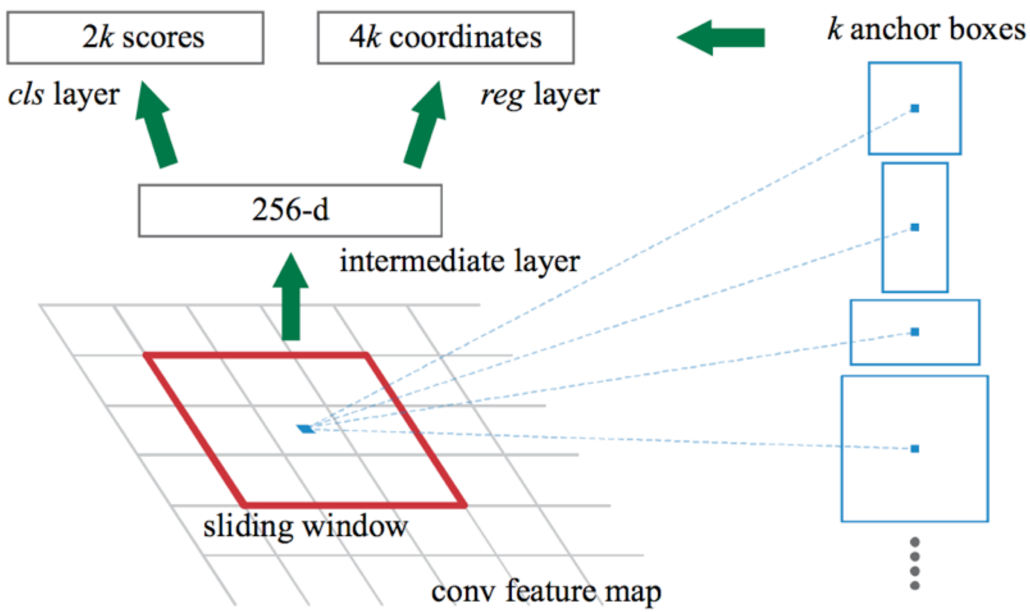
Faster R-CNN

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



Faster R-CNN

- At each location, consider boxes of many different sizes a



Faster R-CNN

- s scales * a aspect ratios = sa anchor boxes
- Use convolutional layer on top of filter map to produce sa scores
- 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