Category-level localization

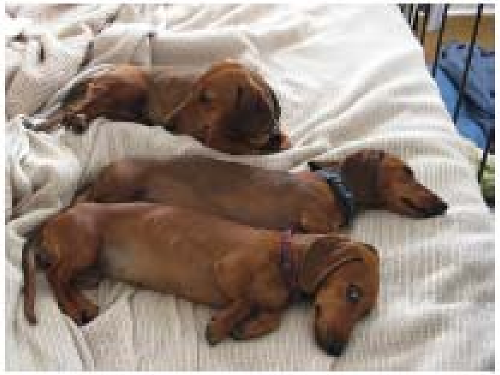
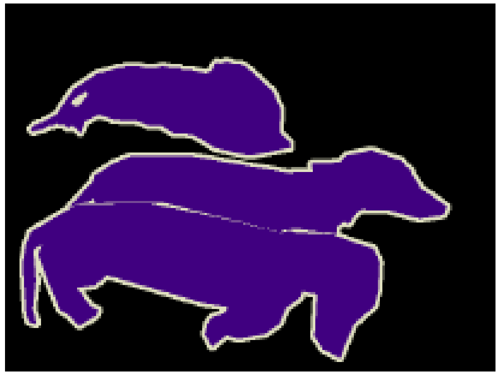
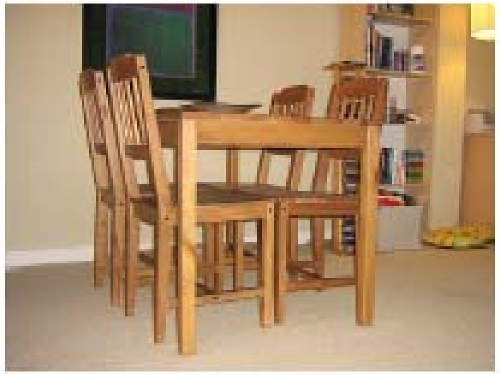
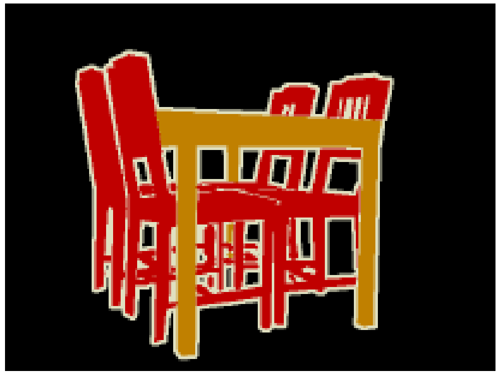
Cordelia Schmid

Recognition

- Classification
  - Object present/absent in an image
  - Often presence of a significant amount of background clutter

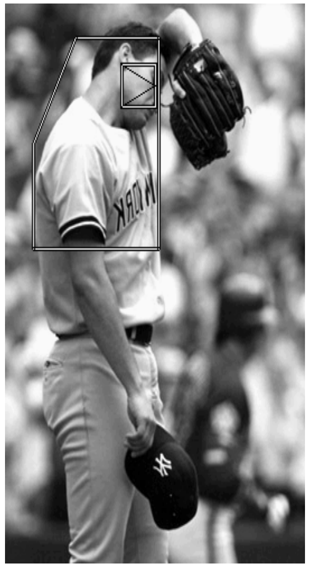
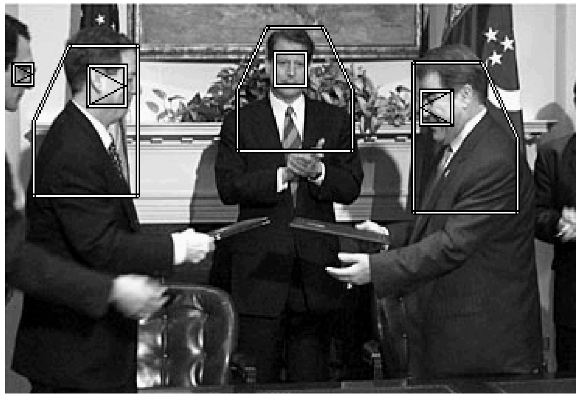
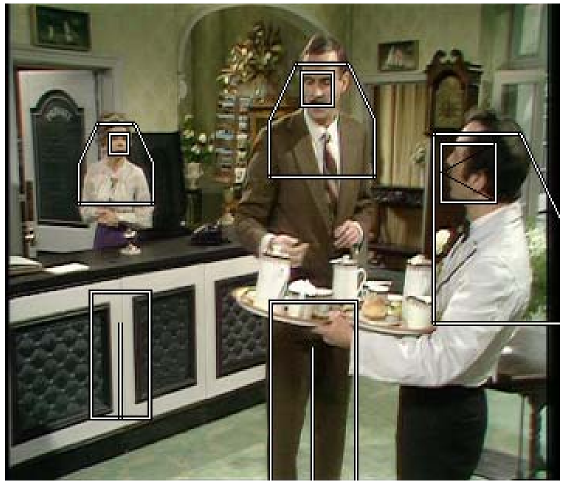
- Localization / Detection
  - Localize object within the frame
  - Bounding box or pixel-level segmentation

Pixel-level object classification

Difficulties

- Intra-class variations

- Scale and viewpoint change
- Multiple aspects of categories

Approaches

- Intra-class variation

  => Modeling of the variations, mainly by learning from a large dataset

- Scale + limited viewpoints changes

  => multi-scale approach

- Multiple aspects of categories

  => separate detectors for each aspect, front/profile face, build an approximate 3D “category” model

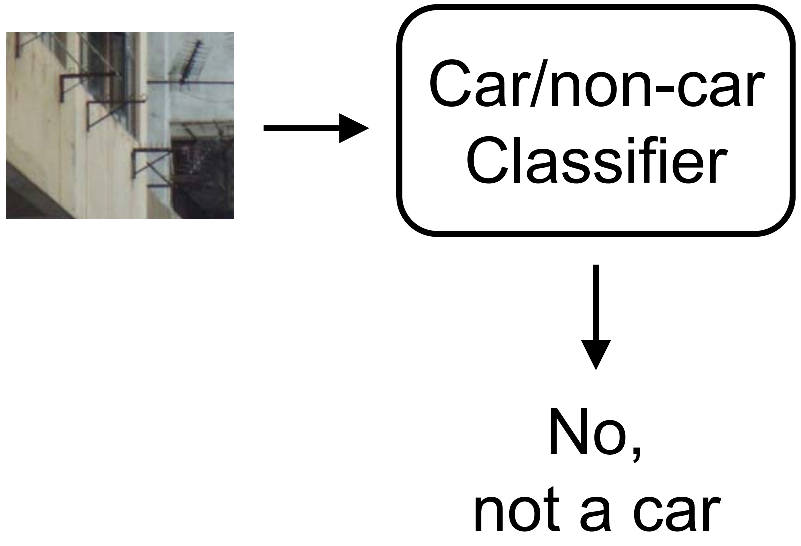
  => high capacity classifiers, i.e. Fisher vector, CNNs

Outline

1. *Sliding window detectors*
2. Features and adding spatial information
3. Histogram of Oriented Gradients (HOG)
4. State of the art algorithms
5. PASCAL VOC and MSR Coco

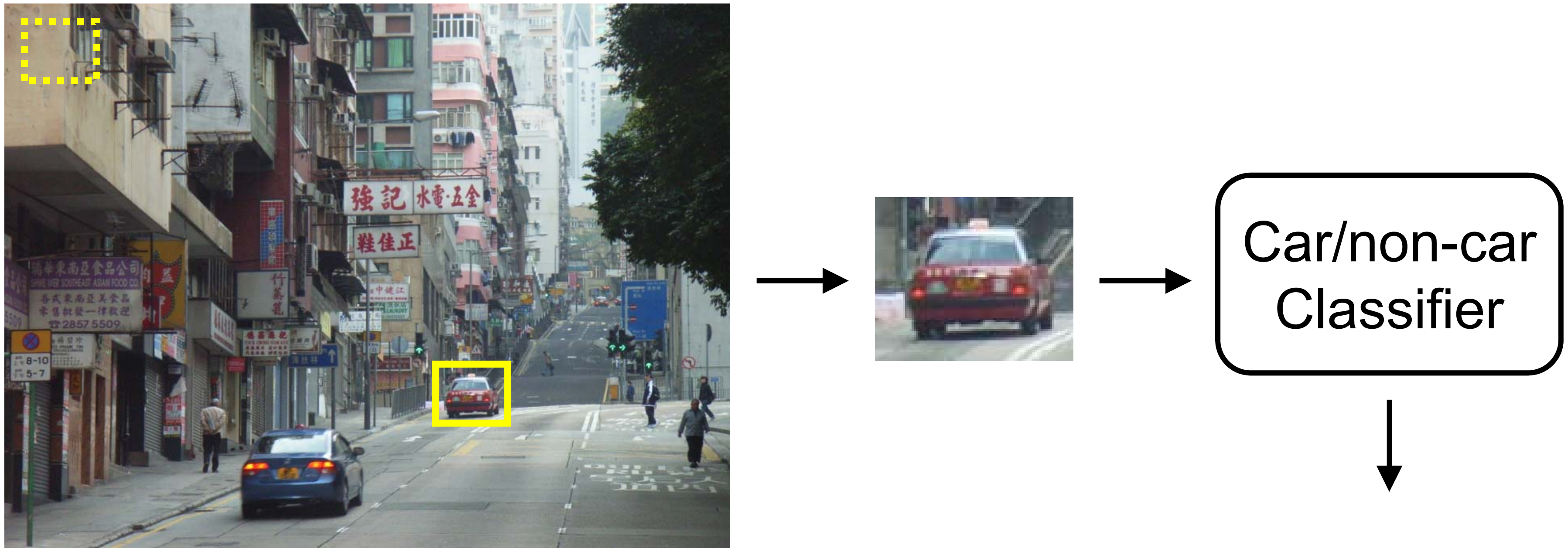
Sliding window detector

- Basic component: binary classifier



Sliding window detector

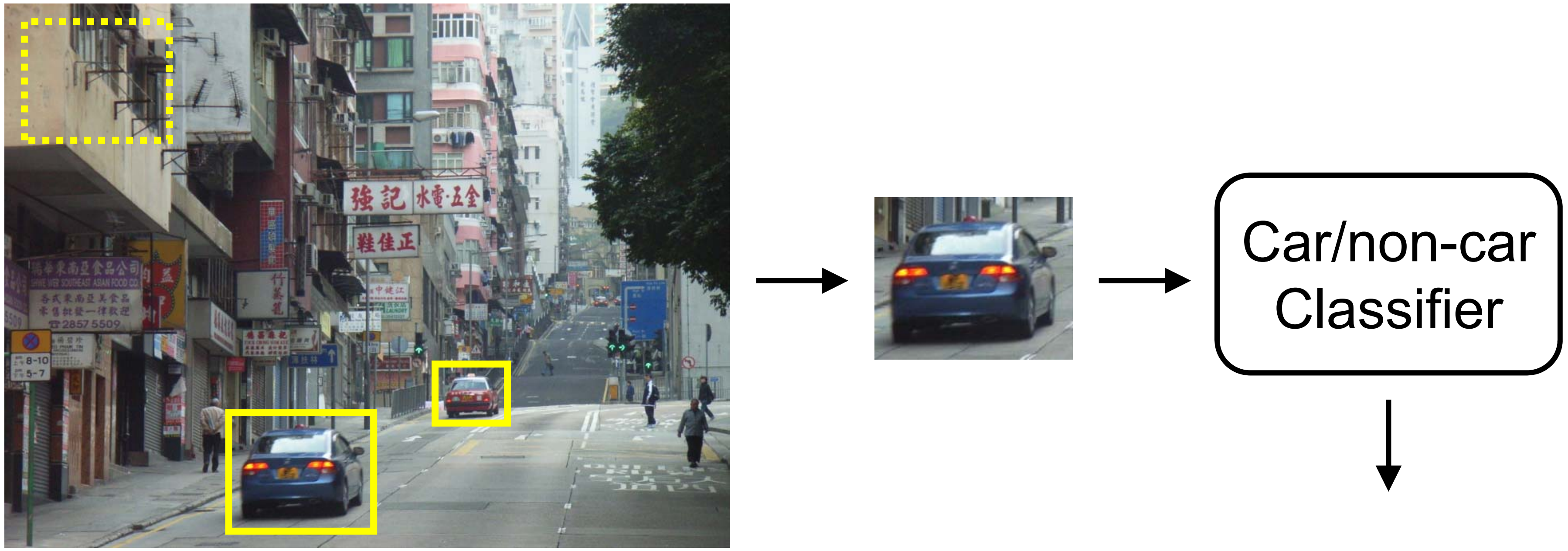
- Detect objects in clutter by **search**



- **Sliding window**: exhaustive search over position and scale

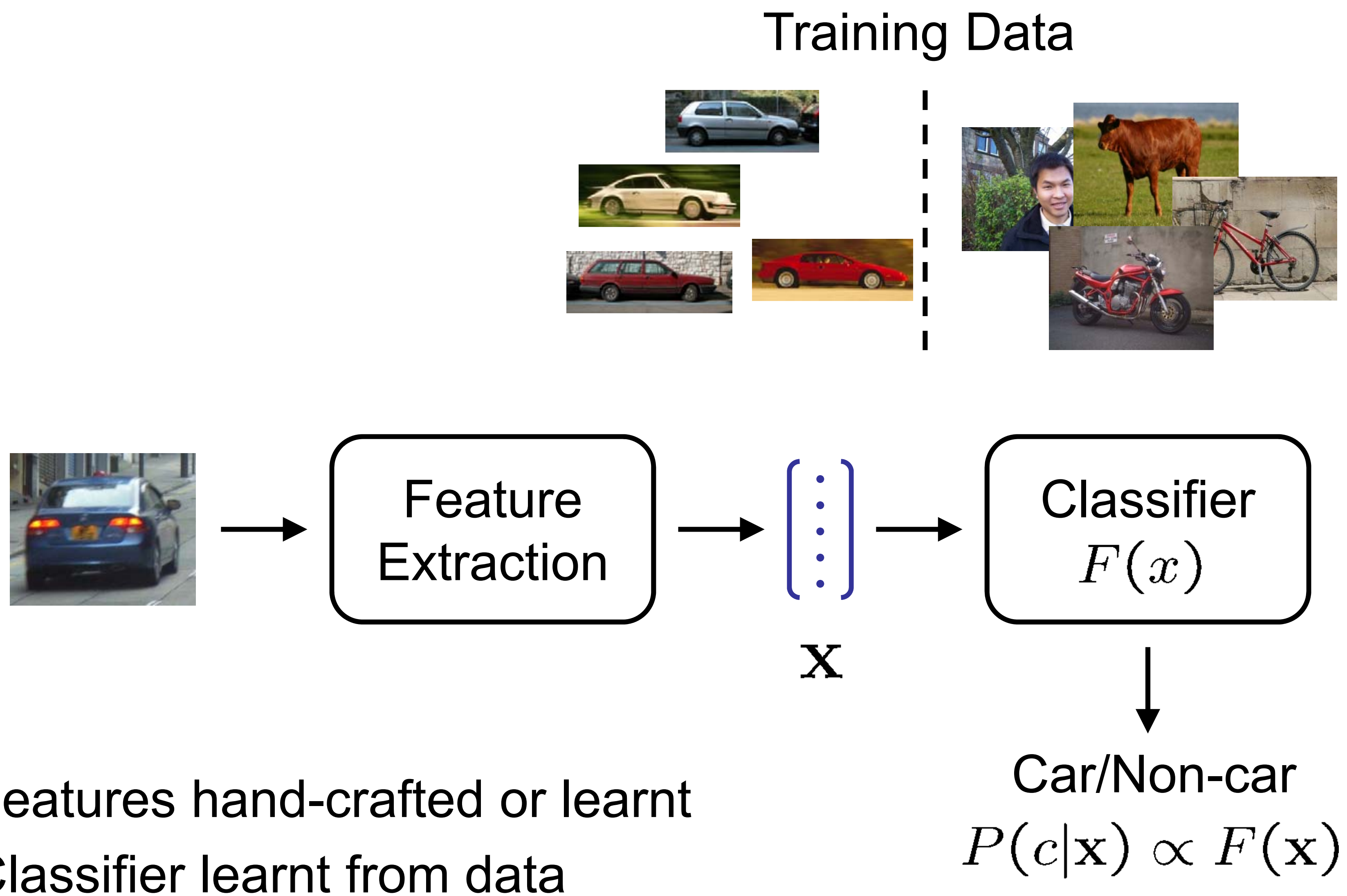
Sliding window detector

- Detect objects in clutter by ***search***



- **Sliding window**: exhaustive search over position and scale

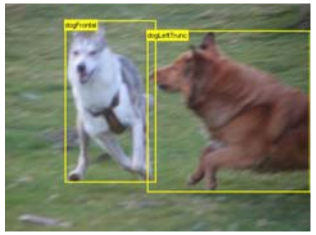
Window (Image) Classification

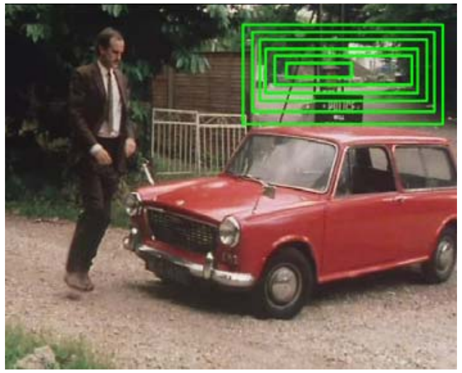


- Features hand-crafted or learnt
- Classifier learnt from data

Problems with sliding windows …

- aspect ratio
- granularity (finite grid)
- partial occlusion
- multiple responses





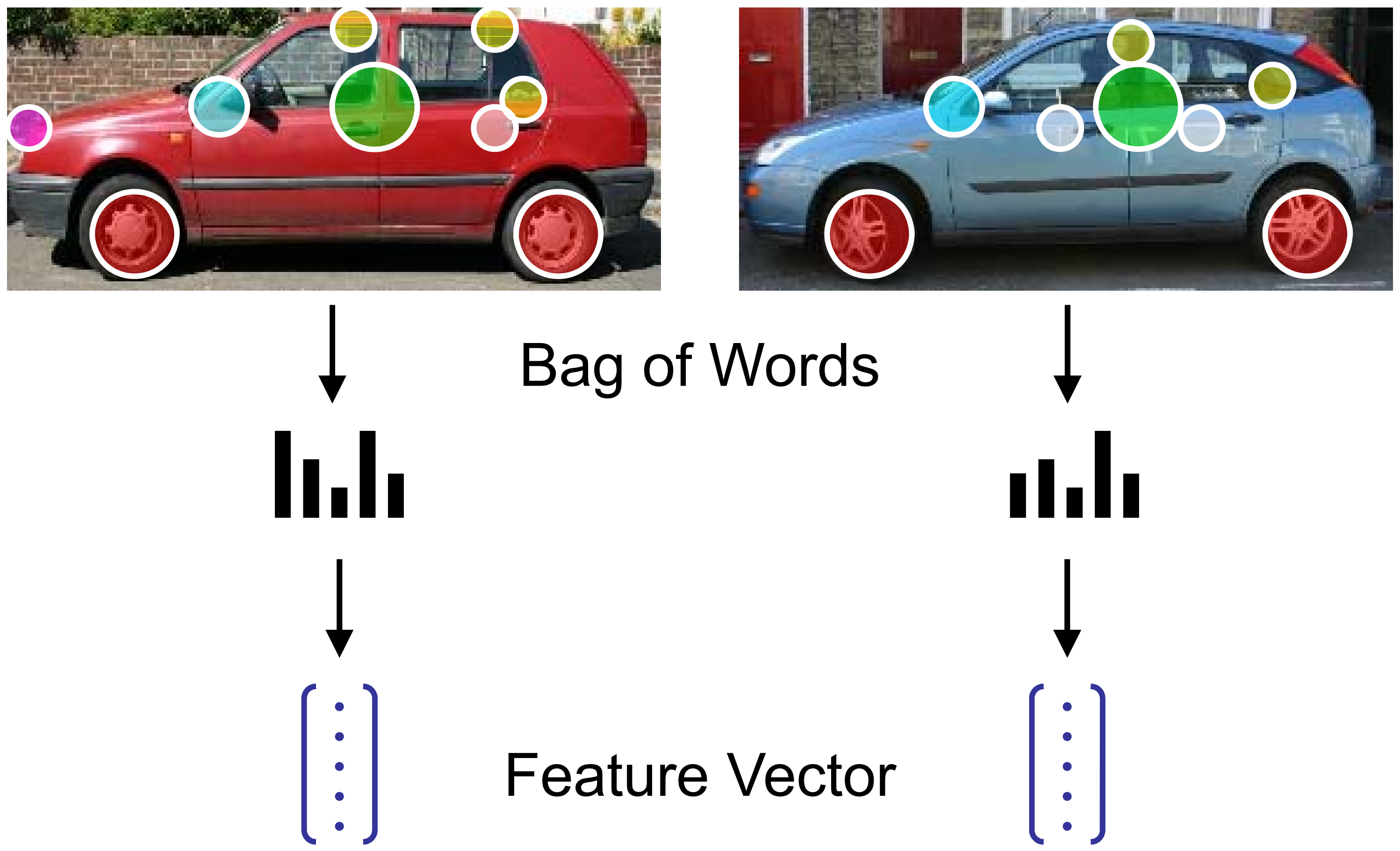
Outline

1. Sliding window detectors
2. *Features and adding spatial information*
3. Histogram of Oriented Gradients (HOG)
4. State of the art algorithms
5. PASCAL VOC and MSR Coco

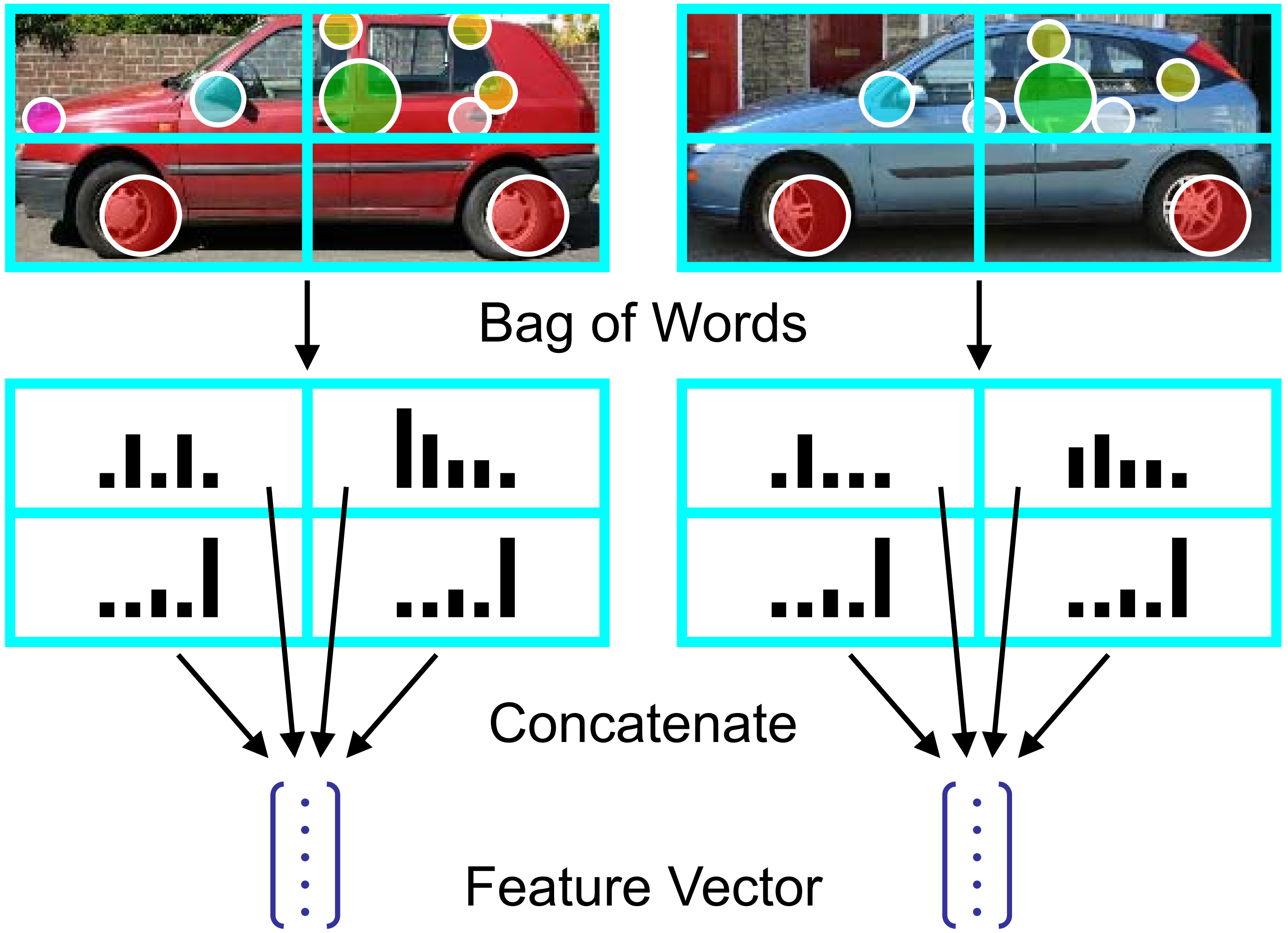
BOW + Spatial pyramids

Start from BoW for region of interest (ROI)

- no spatial information recorded
- sliding window detector

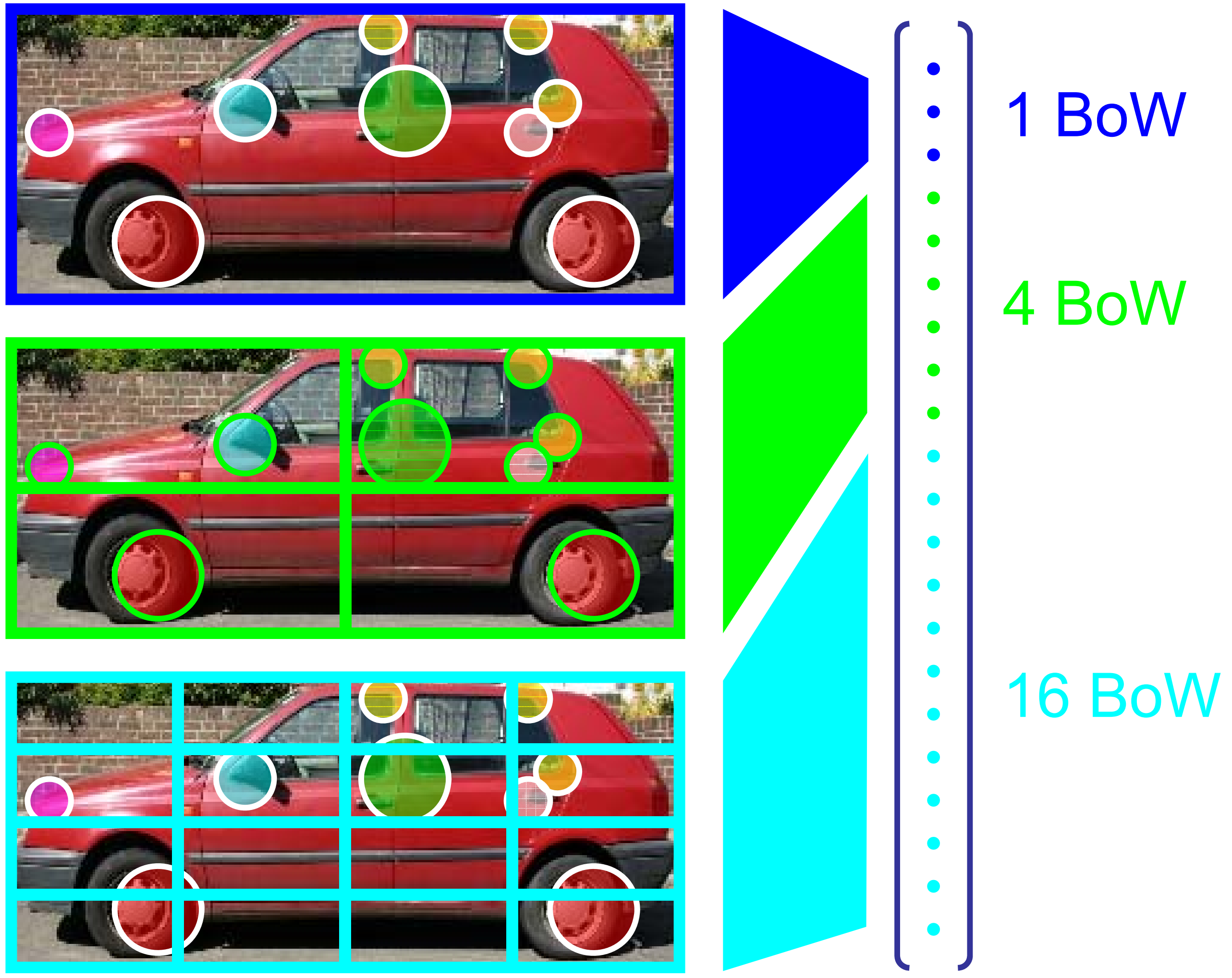


Adding Spatial Information to Bag of Words



Keeps fixed length feature vector for a window

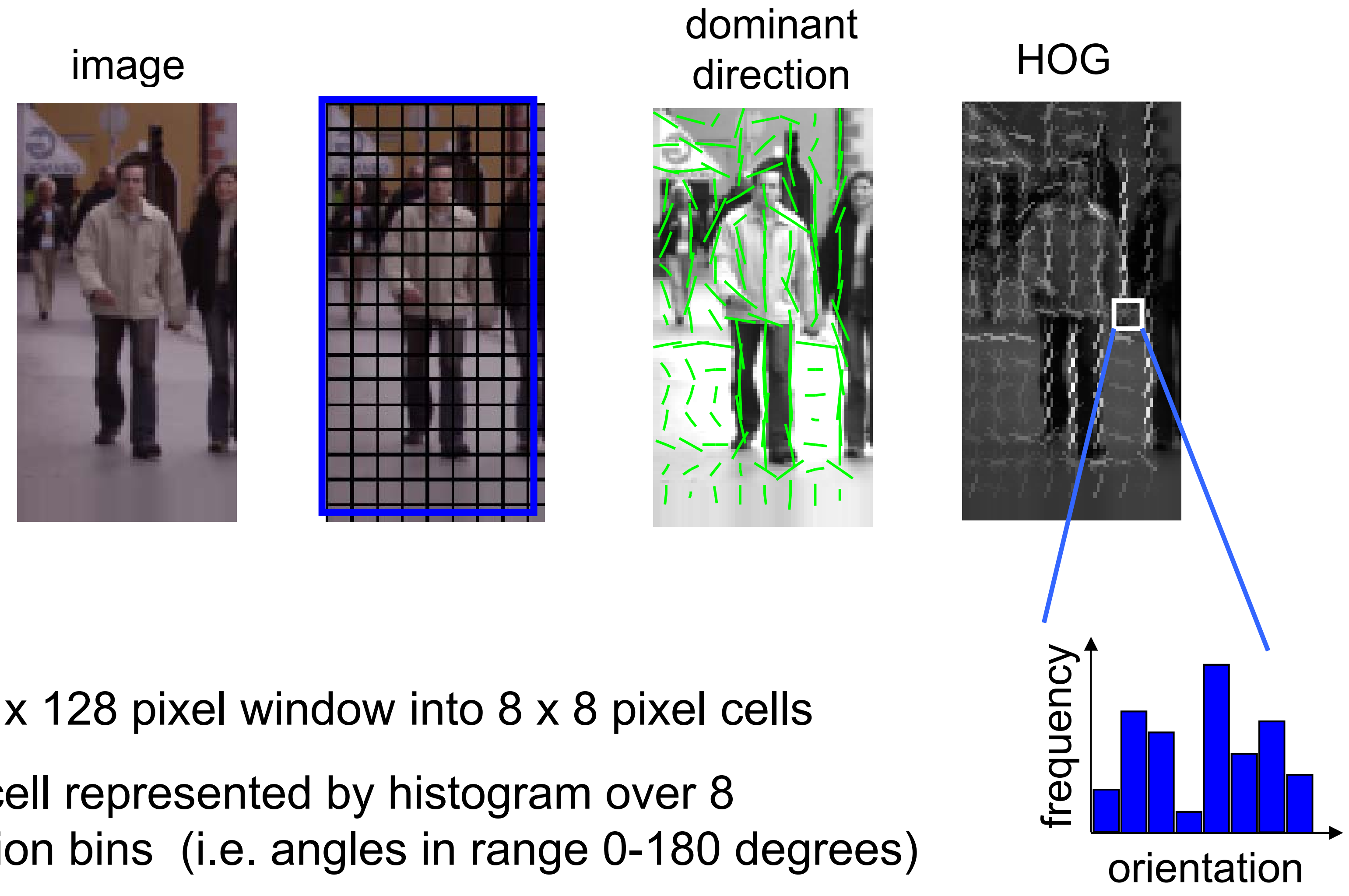
Spatial Pyramid – represent correspondence



Outline

1. Sliding window detectors
2. Features and adding spatial information
3. *Histogram of Oriented Gradients + linear SVM classifier*
4. State of the art algorithms
5. PASCAL VOC and MSR Coco

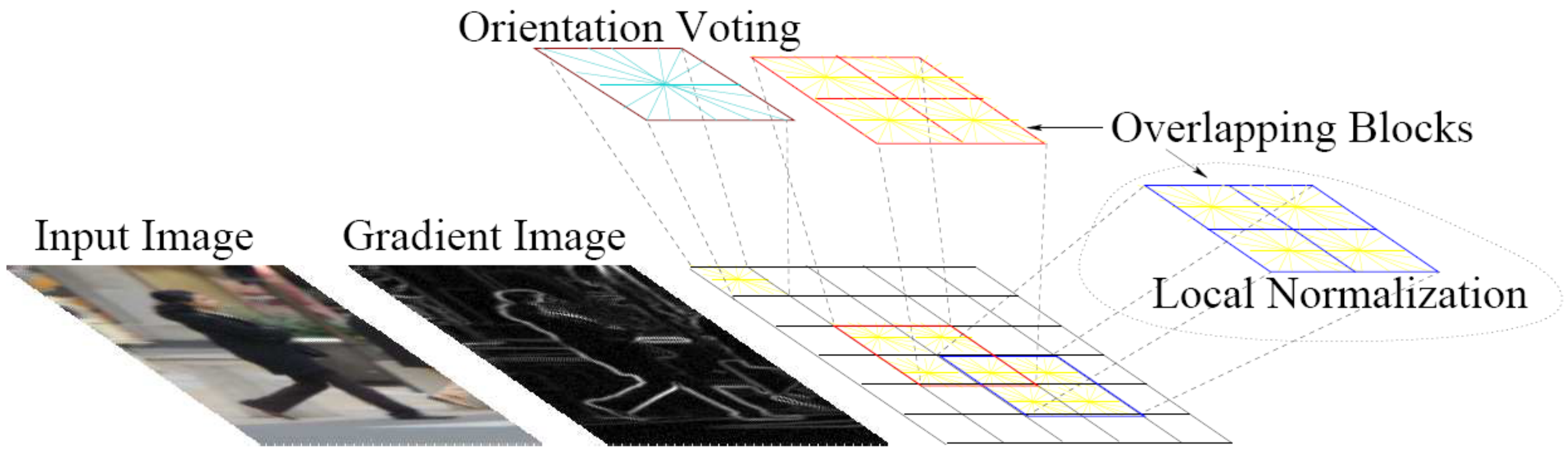
Feature: Histogram of Oriented Gradients (HOG)



• tile 64 x 128 pixel window into 8 x 8 pixel cells

• each cell represented by histogram over 8 orientation bins (i.e. angles in range 0-180 degrees)

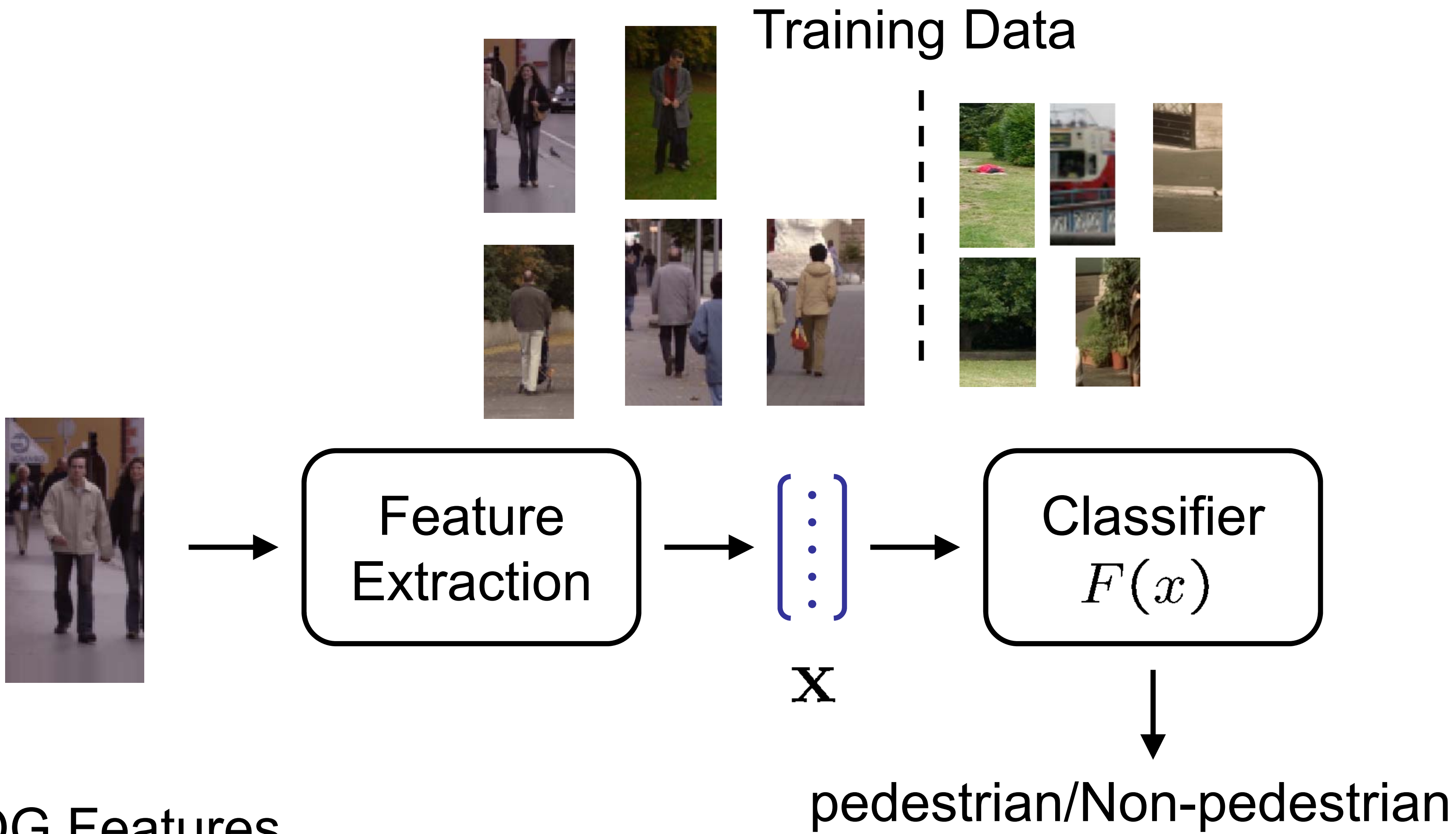
Histogram of Oriented Gradients (HOG) continued



- Adds a second level of overlapping spatial bins re-normalizing orientation histograms over a larger spatial area

- Feature vector dimension (approx) = 16 x 8 (for tiling) x 8 (orientations) x 4 (for blocks) = 4096

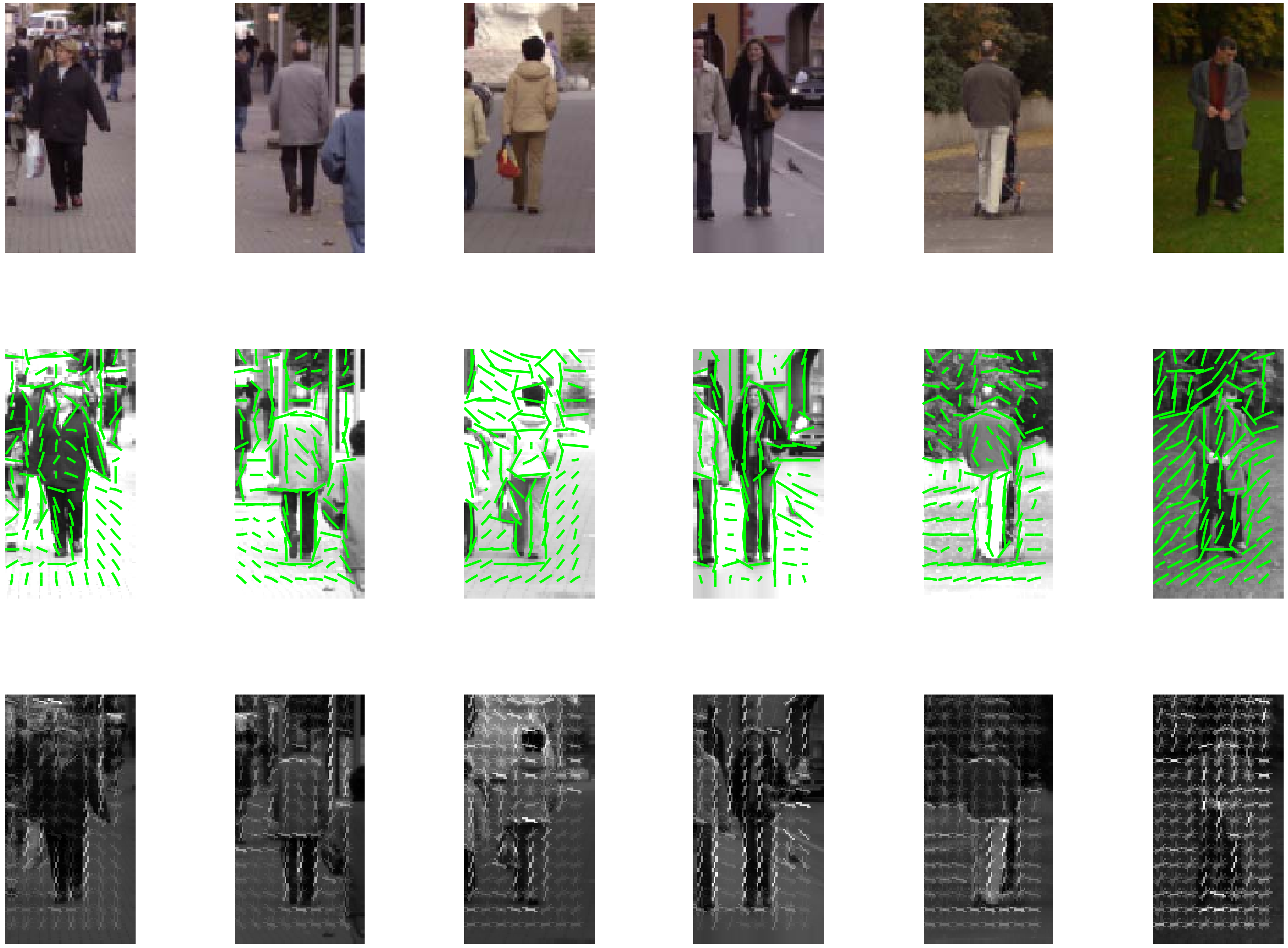
Window (Image) Classification



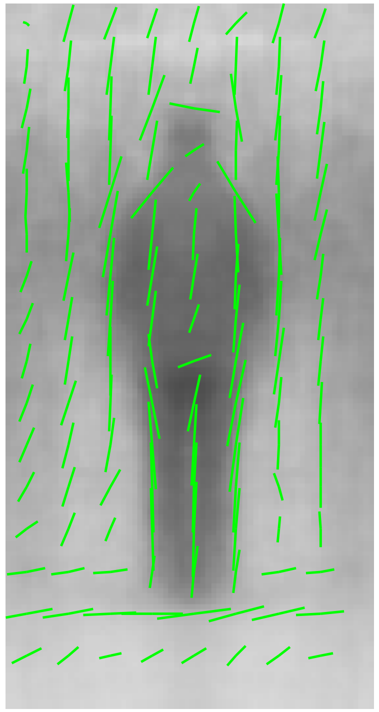
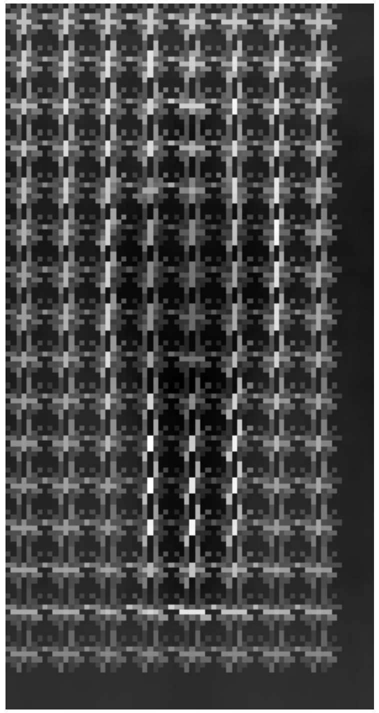
$$P(c|\mathbf{x}) \propto F(\mathbf{x})$$

- HOG Features
- Linear SVM classifier

HOG features

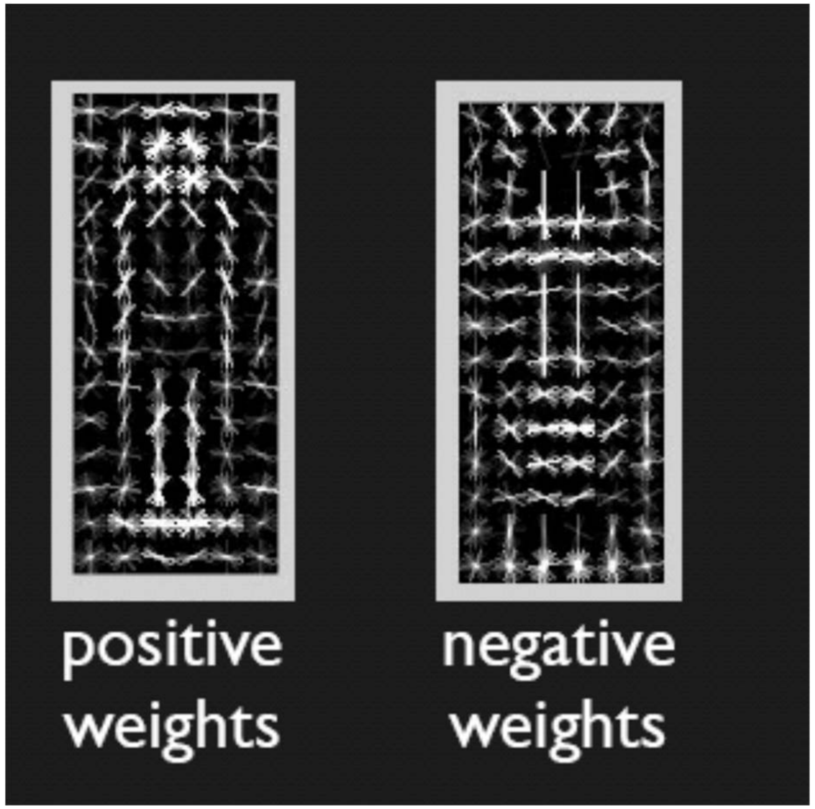


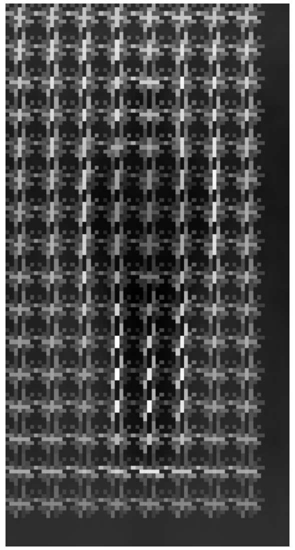
Averaged examples

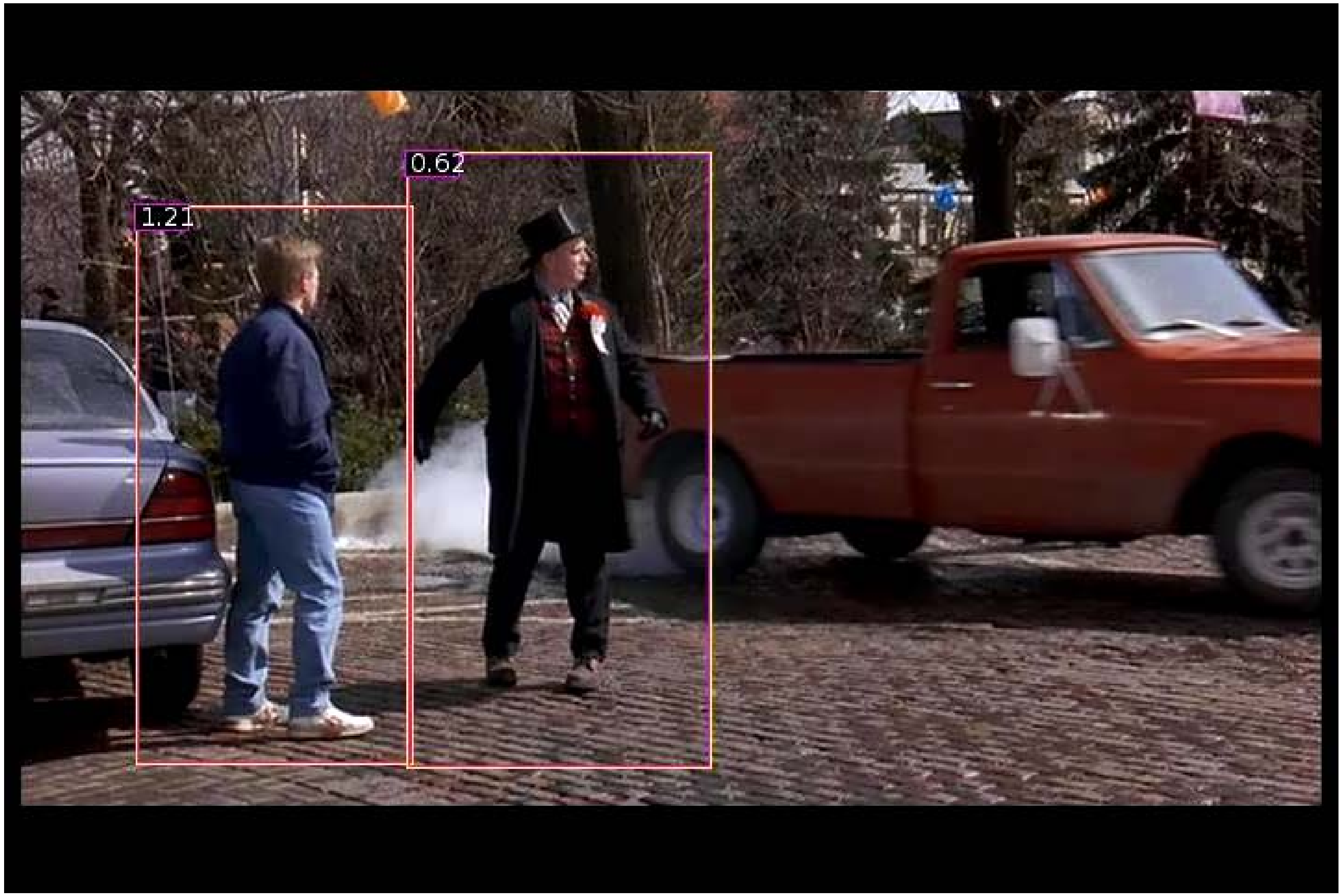



Learned model

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b$$




average over positive training data

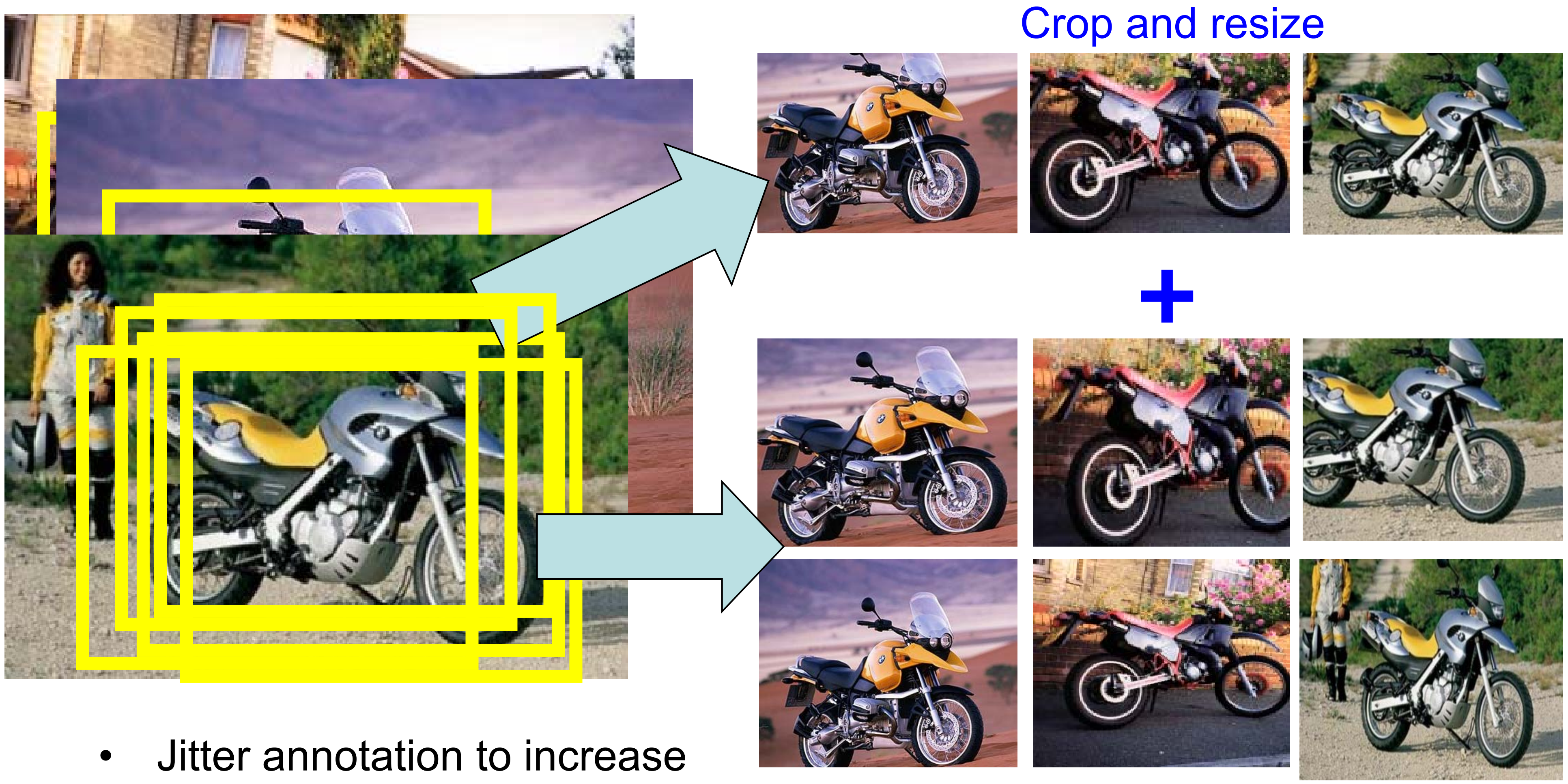


Dalal and Triggs, CVPR 2005

Training a sliding window detector

- Unlike training an image classifier, there are a (virtually) infinite number of possible negative windows
- Training (learning) generally proceeds in three distinct stages:
  1. Bootstrapping: learn an initial window classifier from positives and random negatives, jittering of positives
  2. Hard negatives: use the initial window classifier for detection on the training images (inference) and identify false positives with a high score
  3. Retraining: use the hard negatives as additional training data

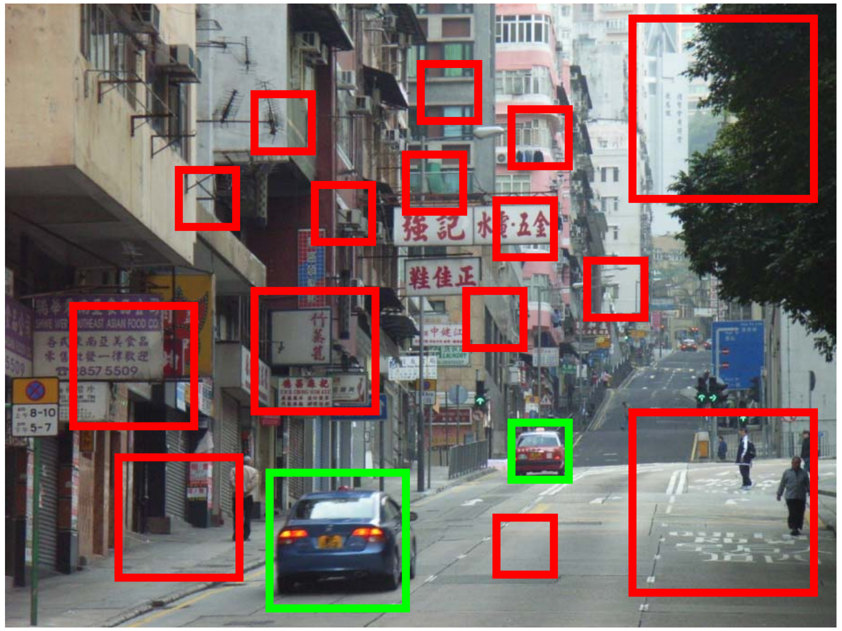
Training: “Jittering” of positive samples



- Jitter annotation to increase the set of positive trainingsamples

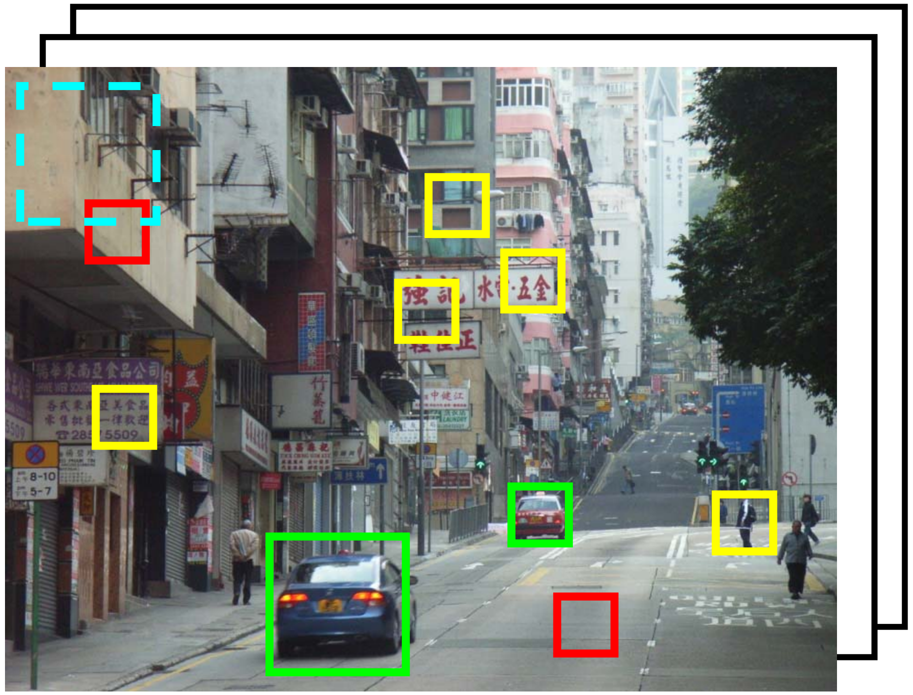
Hard negative mining – why?

- Object **detection** is inherently asymmetric: much more "non-object" than "object" data



- Classifier needs to have very low false positive rate
- Non-object category is very complex – need lots of data

Hard negative mining + retraining



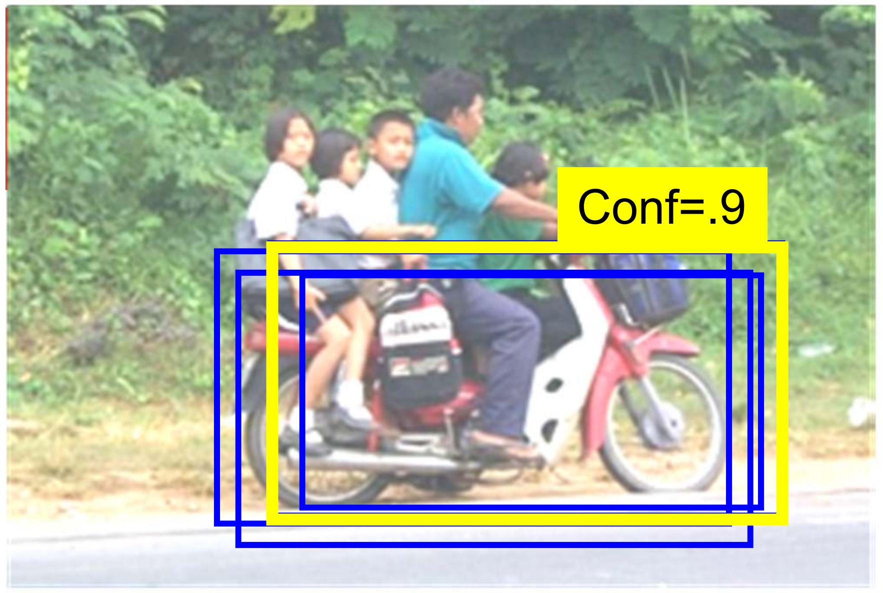
1. Pick negative training set at random
2. Train classifier
3. Run on training data
4. Add false positives to training set
5. Repeat from 2

- Collect a finite but diverse set of non-object windows
- Force classifier to concentrate on **hard negative** examples

- For some classifiers can ensure equivalence to training on entire data set

Test: Non-maximum suppression (NMS)

- Scanning-window detectors typically result in multiple responses for the same object



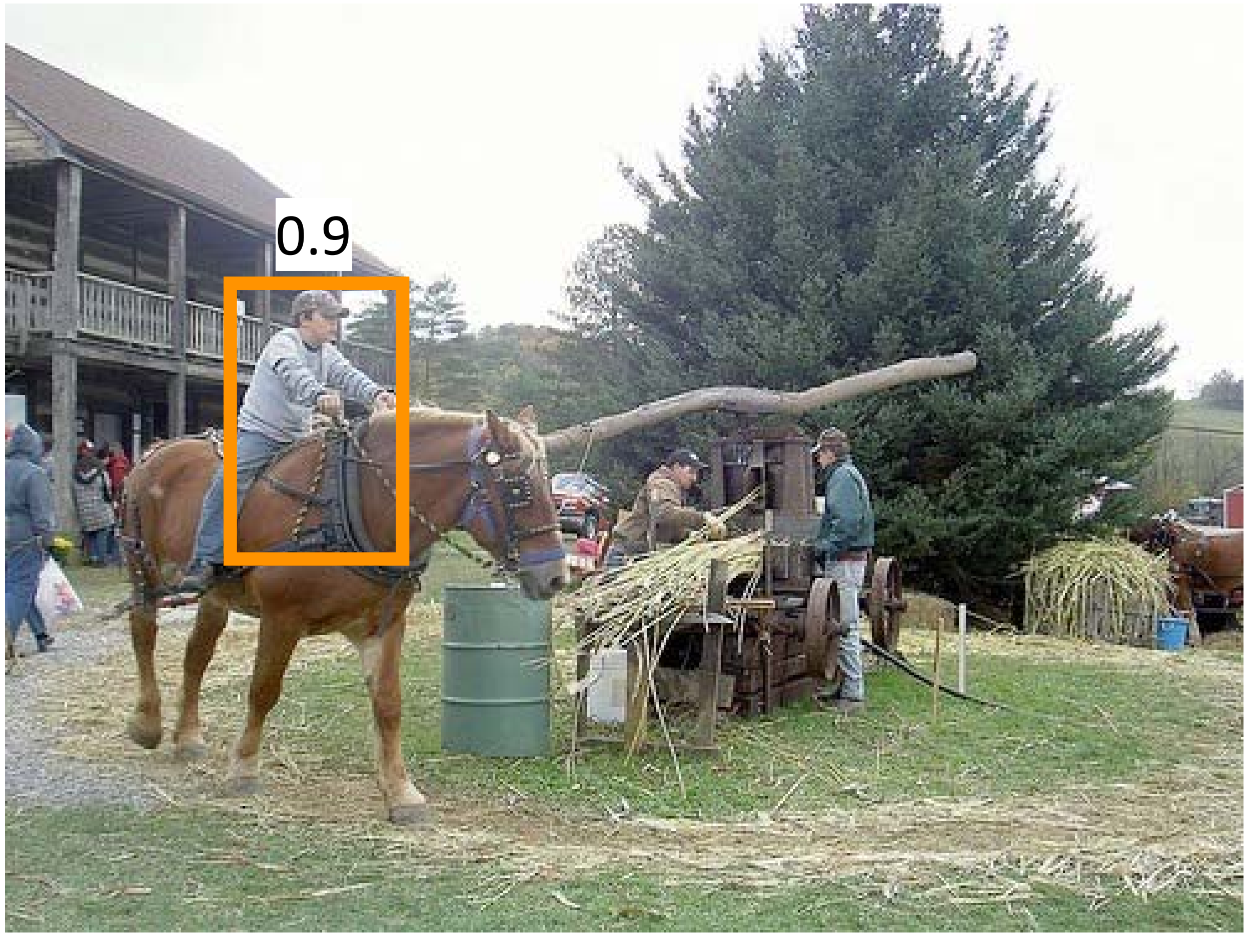
- To remove multiple responses, a simple greedy procedure called “Non-maximum suppression” is applied:

NMS:
1. Sort all detections by detector confidence
2. Choose most confident detection d_i; remove all d_j s.t. $overlap(d_i,d_j)>T$
3. Repeat Step 2. until convergence

Evaluating a detector

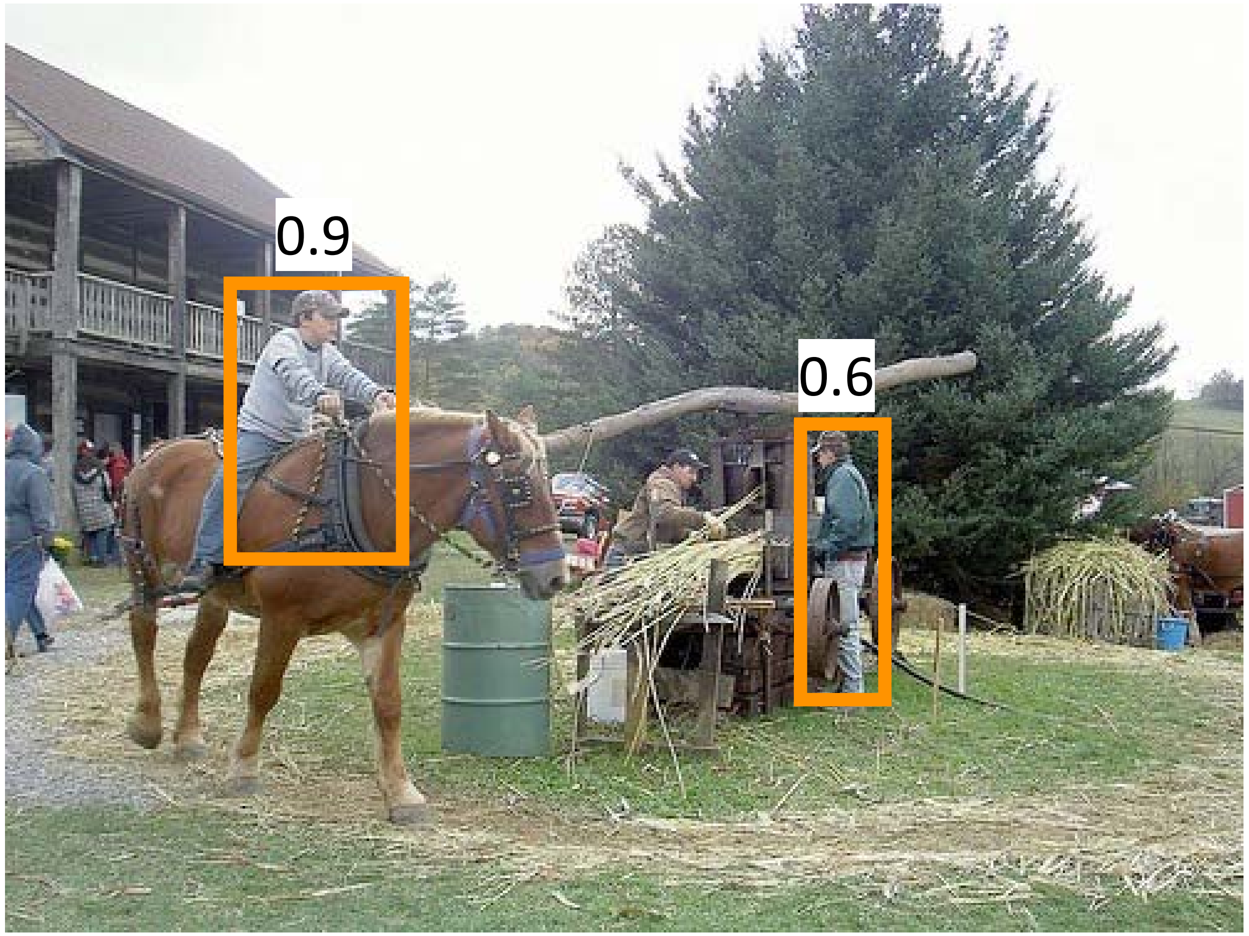
Test image (previously unseen)

First detection ...



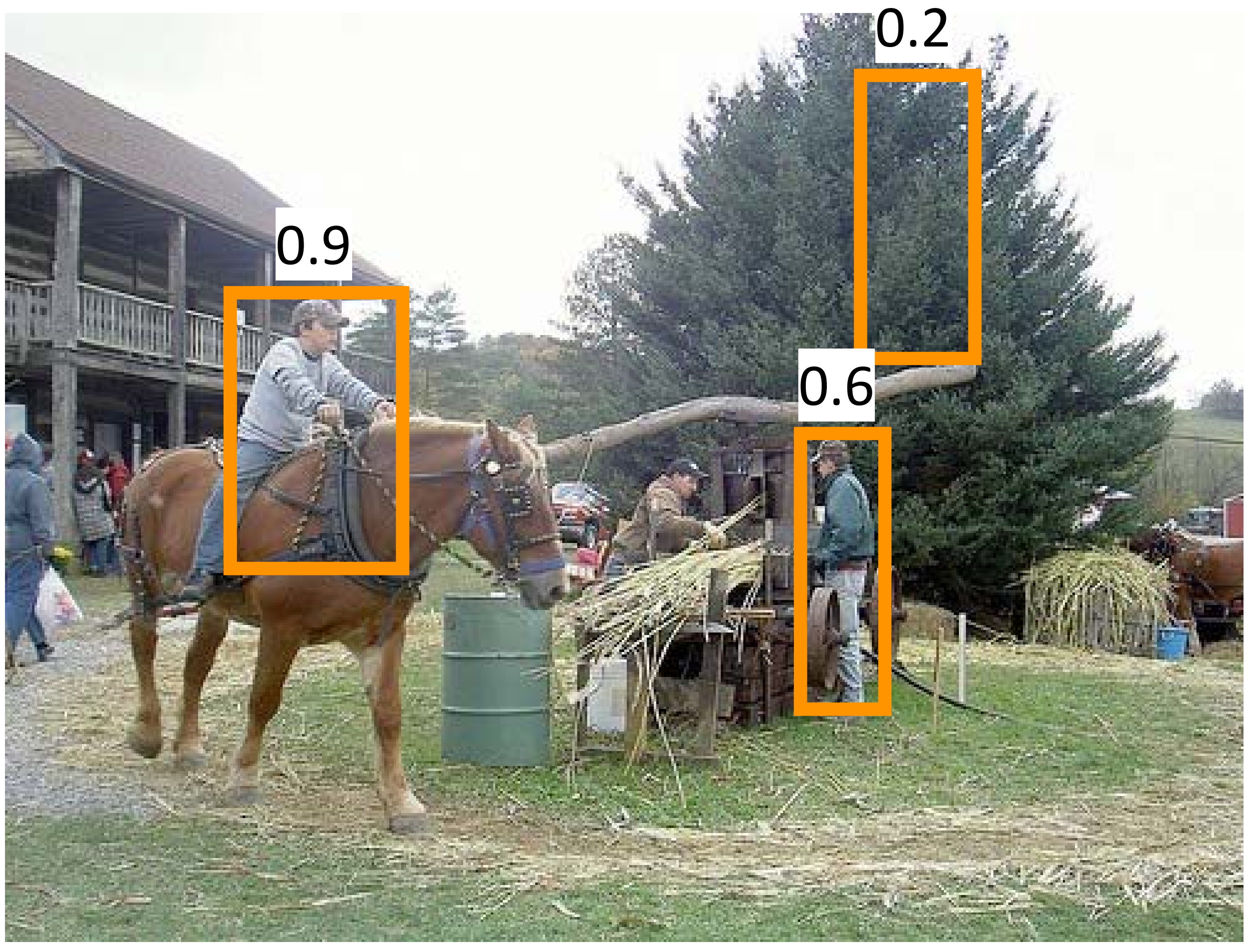
□ ‘person’ detector predictions

Second detection ...



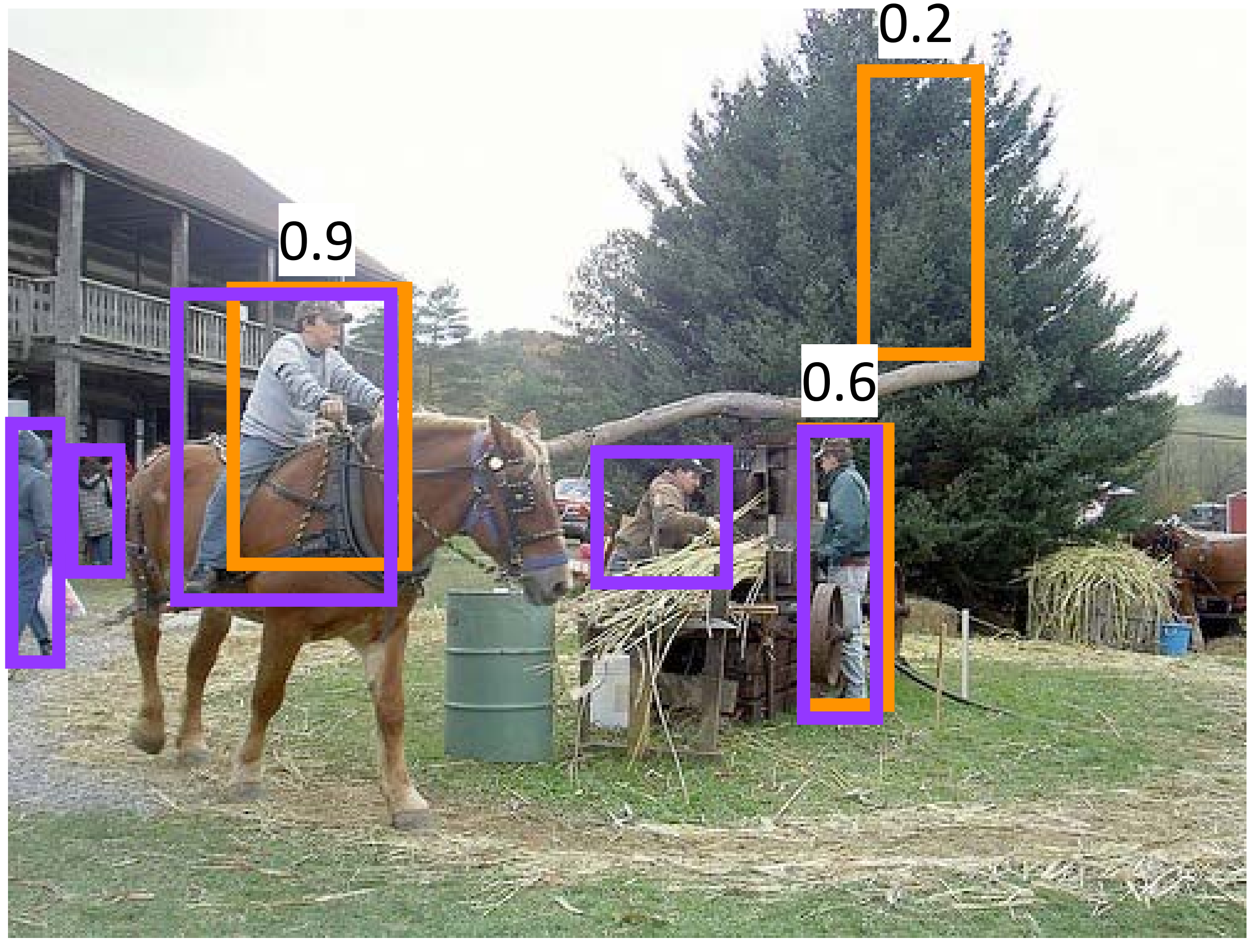
□ 'person' detector predictions

Third detection ...



□ ‘person’ detector predictions

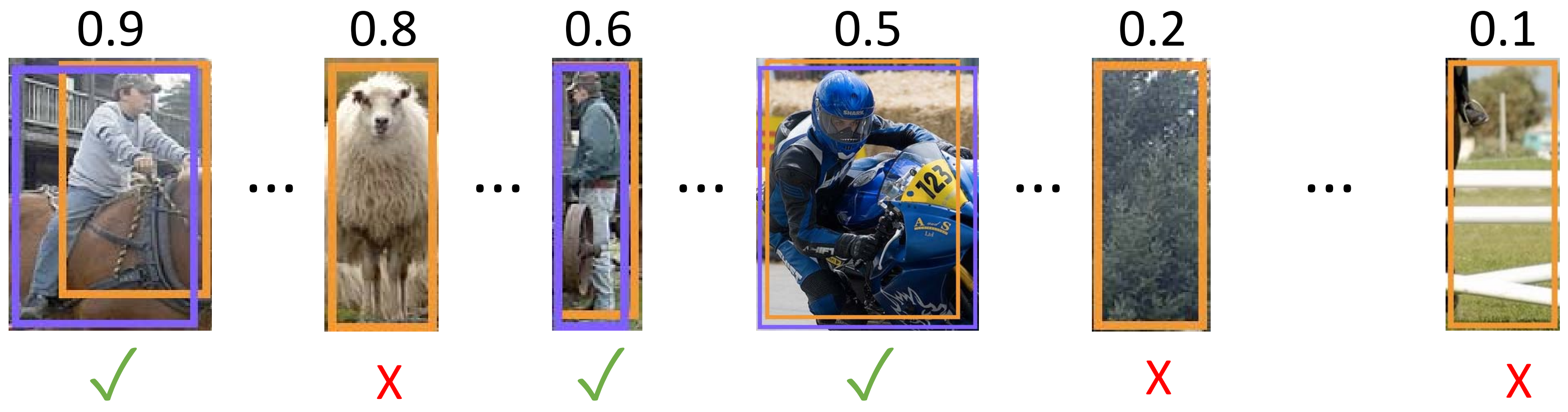
Compare to ground truth



🟧 ‘person’ detector predictions

🟪 ground truth ‘person’ boxes

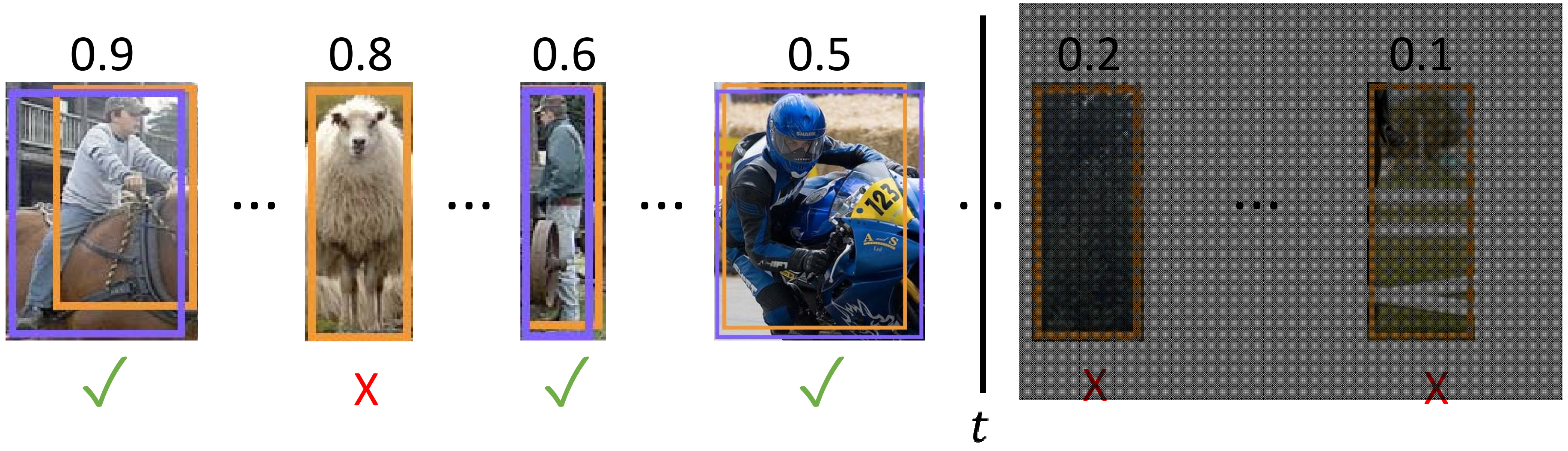
Sort by confidence



true positive
(high overlap)

false positive
(no overlap,
low overlap, or
duplicate)

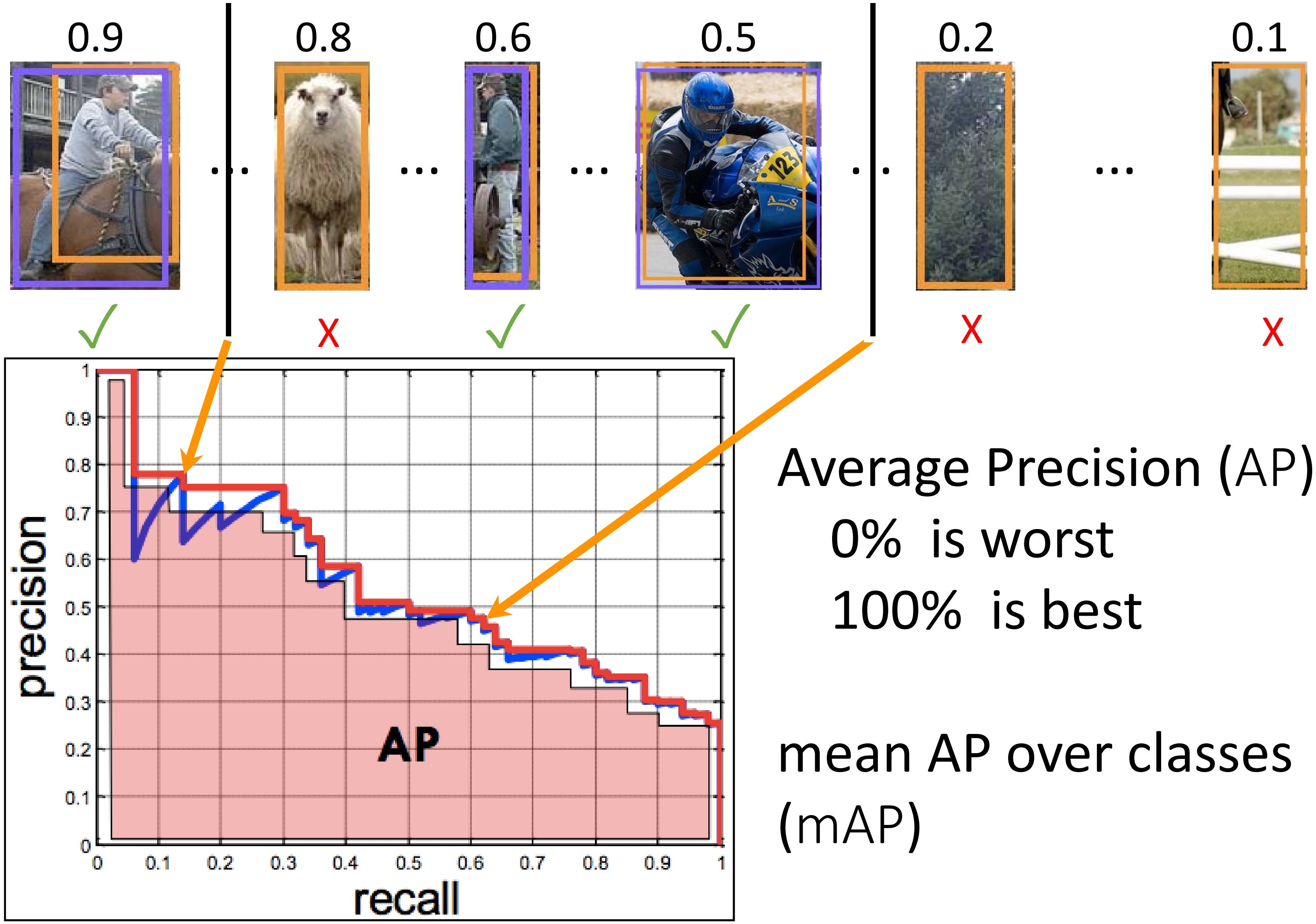
Evaluation metric



$$precision@t = \frac{\#true\ positives@t}{\#true\ positives@t + \#false\ positives@t} \qquad \frac{✓}{✓ + X}$$

$$recall@t = \frac{\#true\ positives@t}{\#ground\ truth\ objects}$$

Evaluation metric



Average Precision (AP)

0% is worst

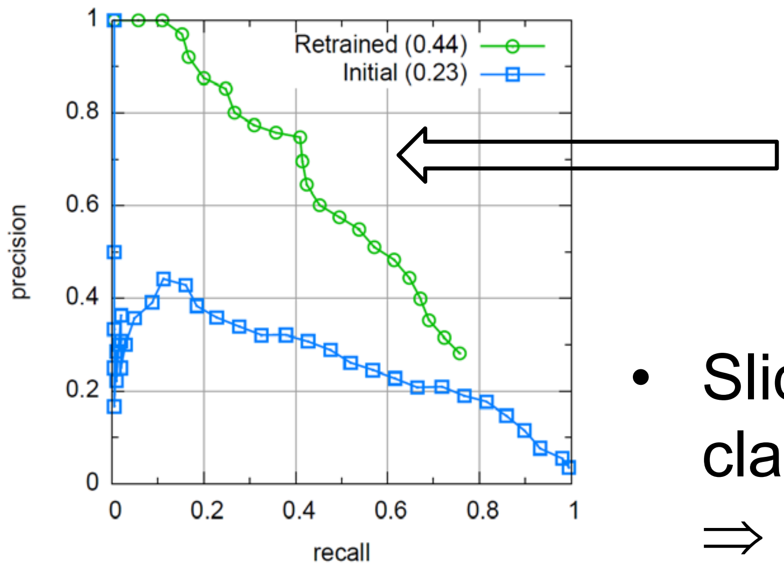
100% is best

mean AP over classes (mAP)

Outline

1. Sliding window detectors
2. Features and adding spatial information
3. HOG + linear SVM classifier
4. *State of the art algorithms*
5. PASCAL VOC and MSR Coco

HOG + SVM Object detector



Far from perfect.
What can be improved?

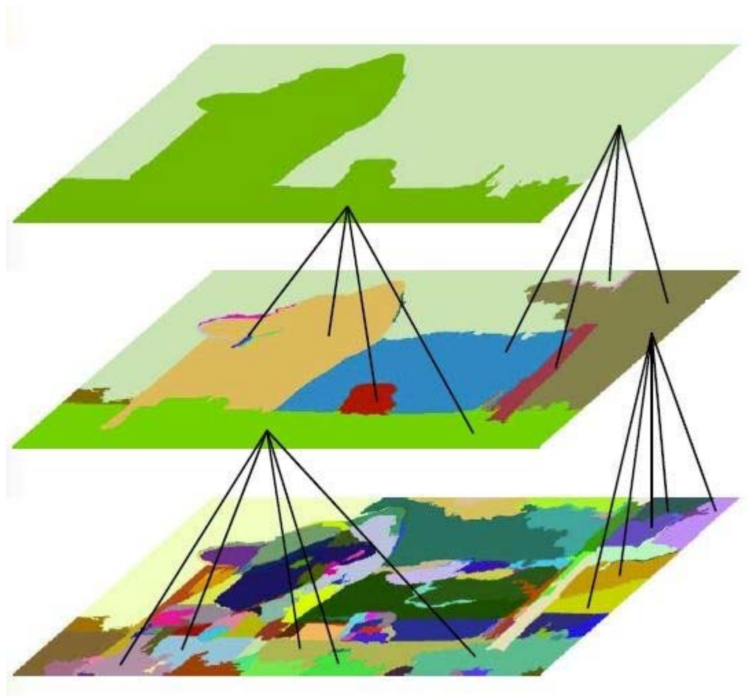
- Sliding-window detectors need to classify 100K samples per image
  $\Rightarrow$ **speed matters**
- HOG + linear SVM is fast but too simple

Approach:

1. Reduce the search space 100K → ~1K windows
   $\Rightarrow$ **Region proposals**
2. Use more complex features and classifiers
   $\Rightarrow$ **CNN**

Region proposals: Selective Search

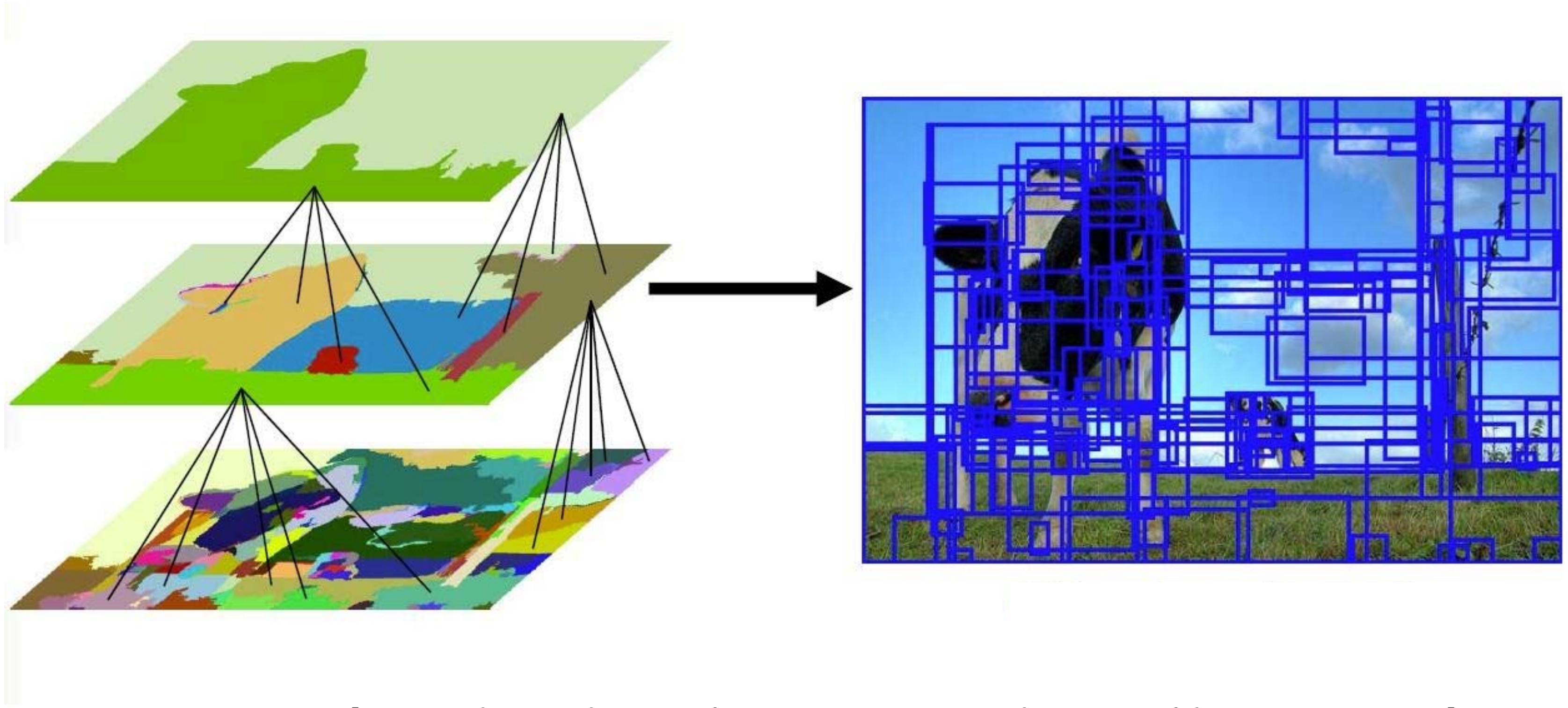
1. Merge two most similar regions based on *S.*
2. Update similarities between the new region and its neighbors.
3. Go back to step 1. until the whole image is a single region.



[K. van de Sande, J. Uijlings, T. Gevers, and A. Smeulders, ICCV 2011]

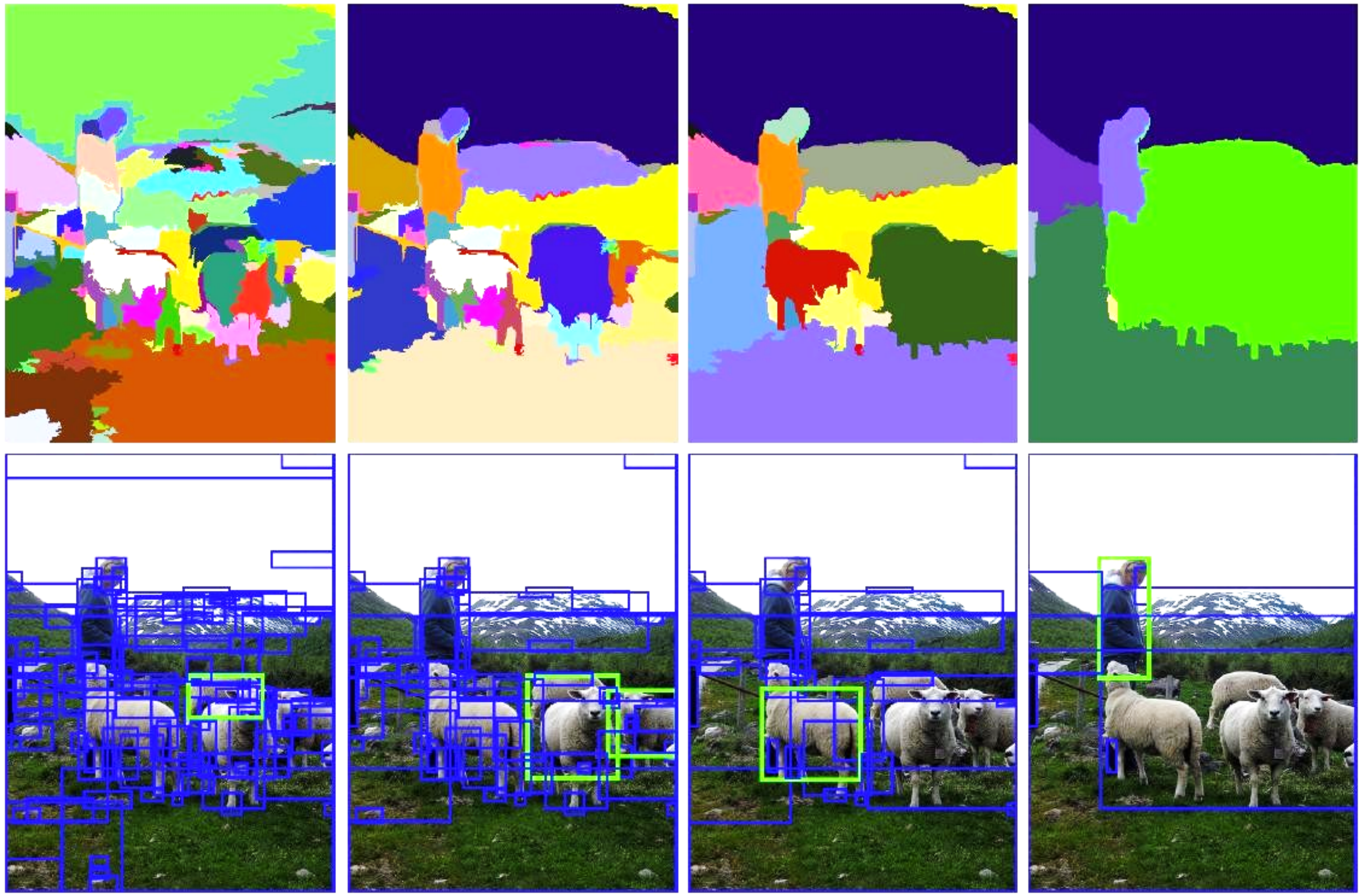
Region proposals: Selective Search

- Take bounding boxes of all generated regions and treat them as possible object locations.



[K. van de Sande, J. Uijlings, T. Gevers, and A. Smeulders, ICCV 2011]

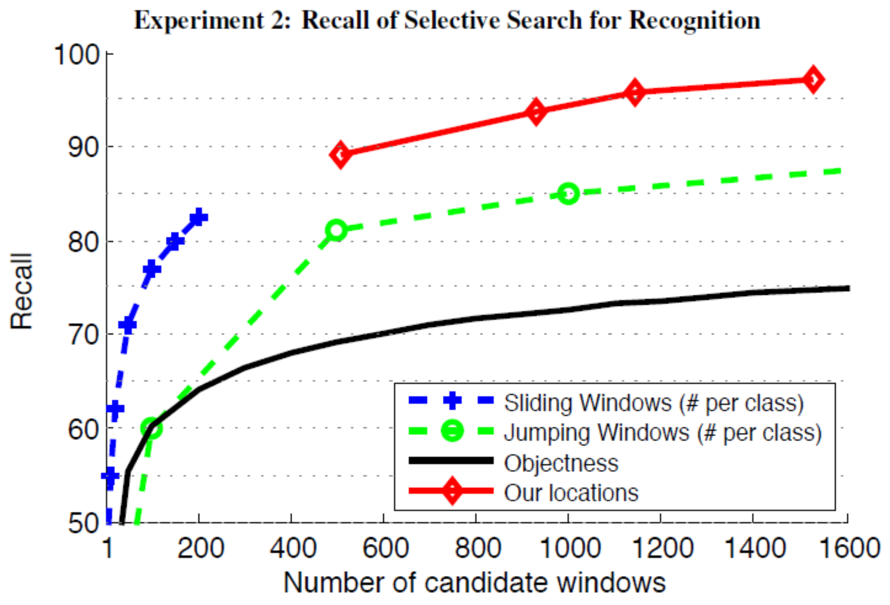
Region proposals: Selective Search



[K. van de Sande, J. Uijlings, T. Gevers, and A. Smeulders, ICCV 2011]

Selective Search: Comparison

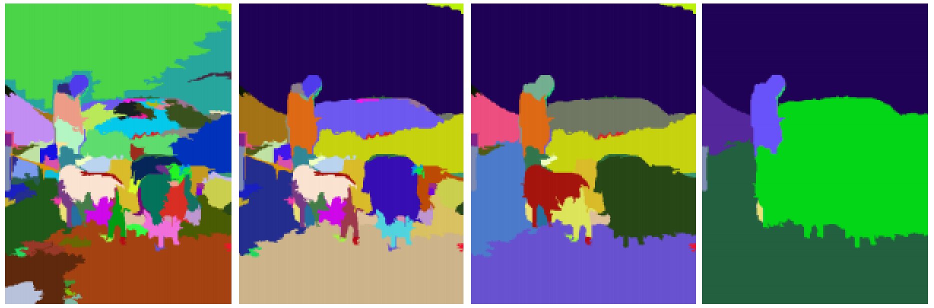
	Max. recall (%)	# windows
Sliding Windows [13]	83.0	200 per class
Jumping Windows [27]	94.0	10,000 per class
'Objectness' [1]	82.4	10,000
Our hypotheses	96.7	1,536



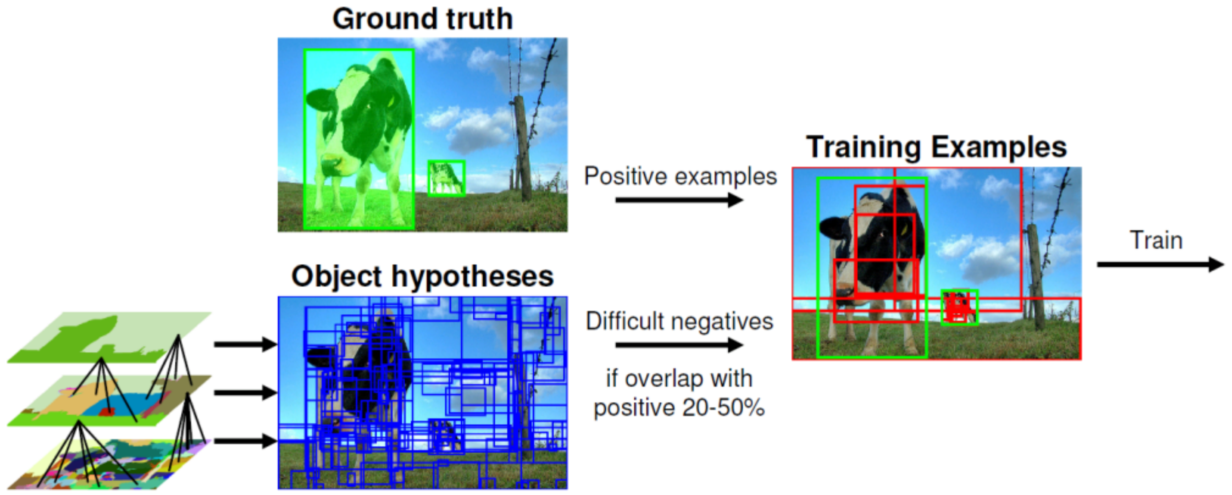
[K. van de Sande, J. Uijlings, T. Gevers, and A. Smeulders, ICCV 2011]

Selective search for object location [v.d.Sande et al. 11]

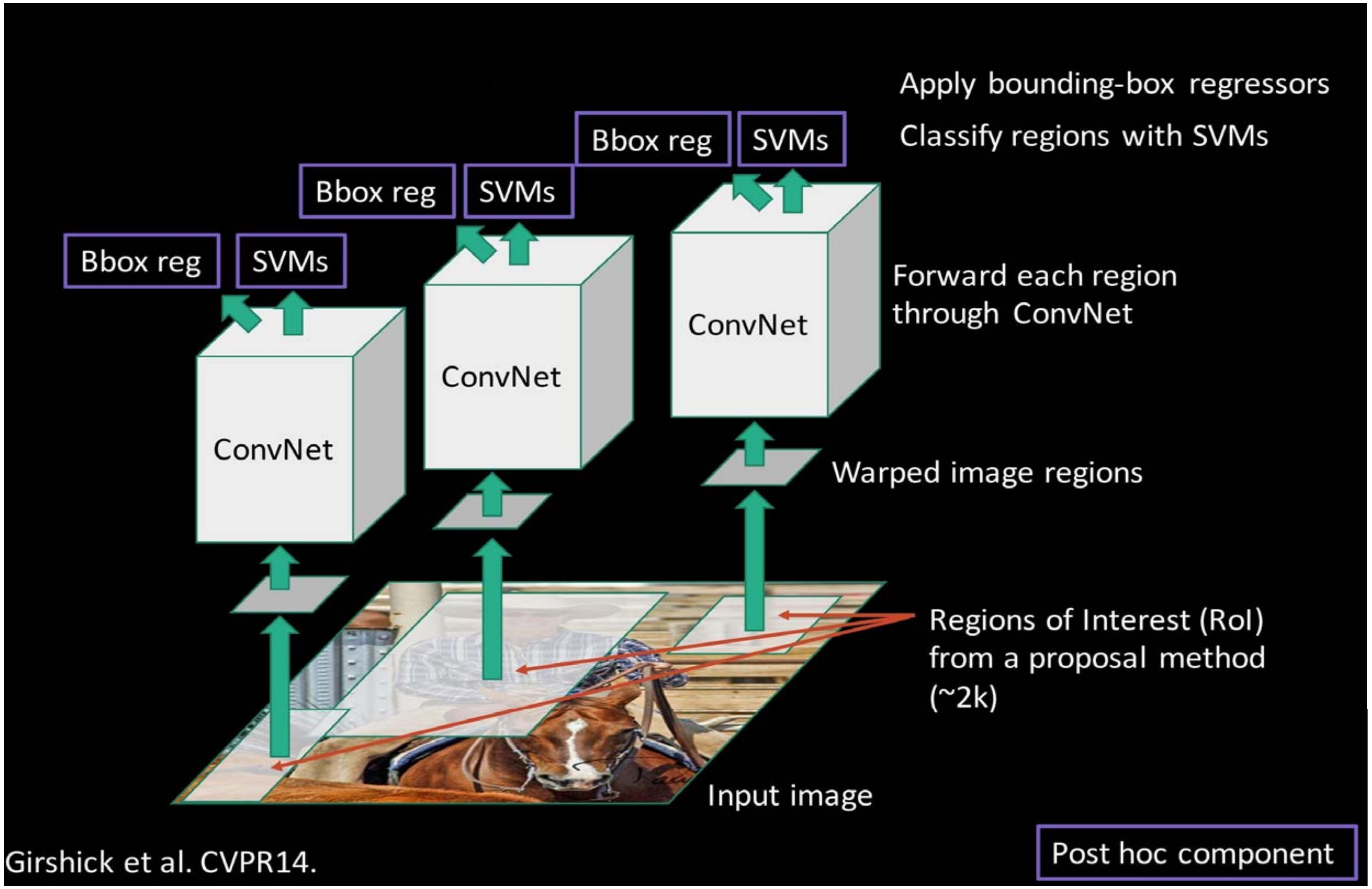
- Select *class-independent* candidate image windows with segmentation



- Local features + bag-of-words
- SVM classifier with histogram intersection kernel + hard negative mining



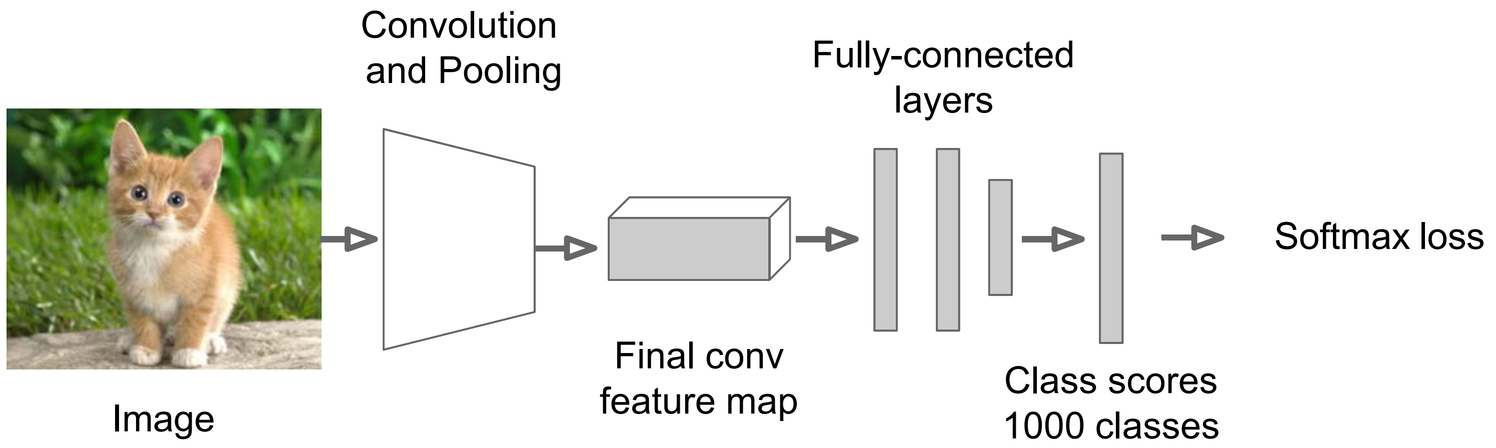
Selective search regions with CNN features: R-CNN


Apply bounding-box regressors
Classify regions with SVMs
Bbox reg
SVMs
Bbox reg
SVMs
Bbox reg
SVMs
ConvNet
ConvNet
ConvNet
Forward each region through ConvNet
Warped image regions
Regions of Interest (RoI) from a proposal method (~2k)
Input image
Girshick et al. CVPR14.
Post hoc component

Slide credit: Ross Girschick [Girschick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014]

R-CNN Training

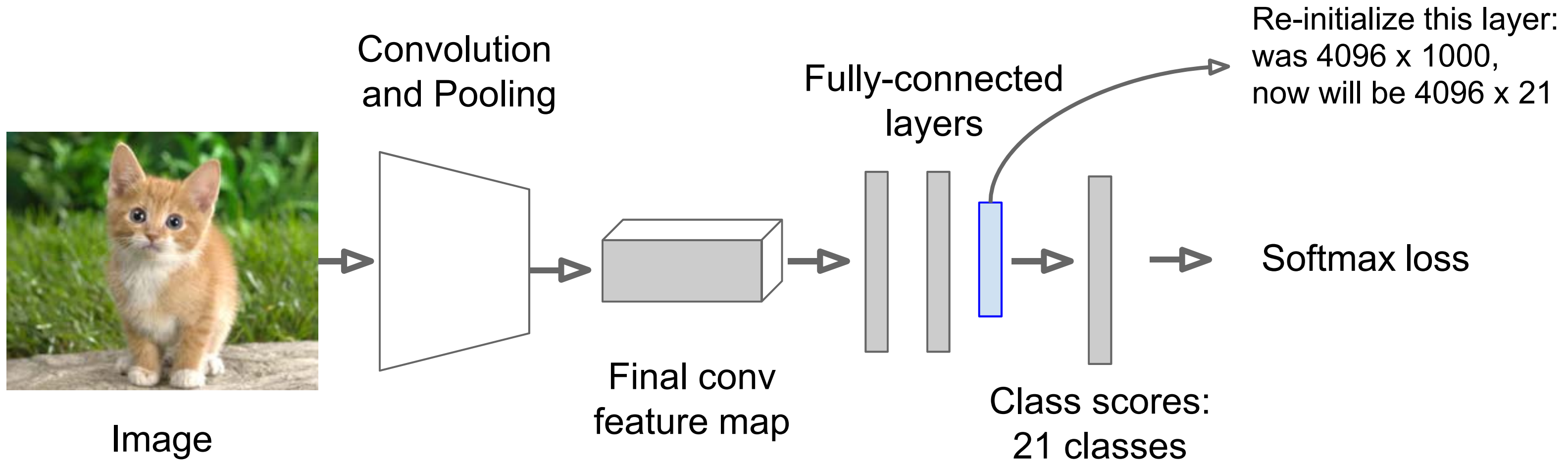
Step 1: Train (or download) a classification model for ImageNet (AlexNet)



R-CNN Training

Step 2: Fine-tune model for detection

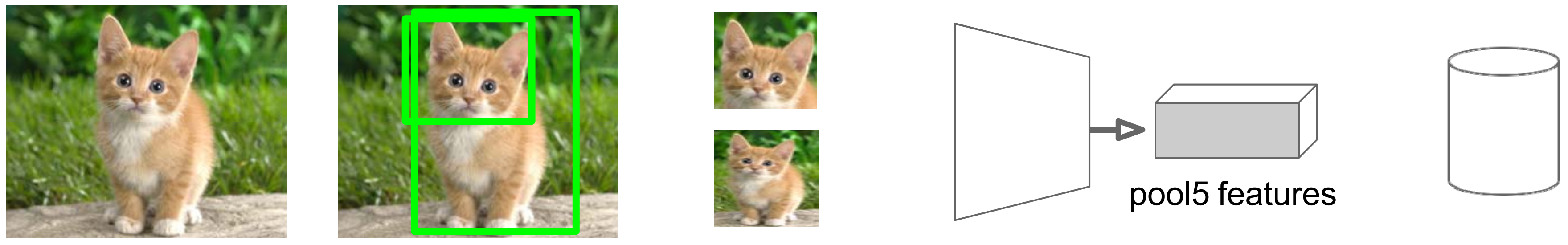
- Instead of 1000 ImageNet classes, want 20 object classes + background
- Throw away final fully-connected layer, reinitialize this layer from scratch
- Keep training model using positive / negative regions from detection images



R-CNN Training

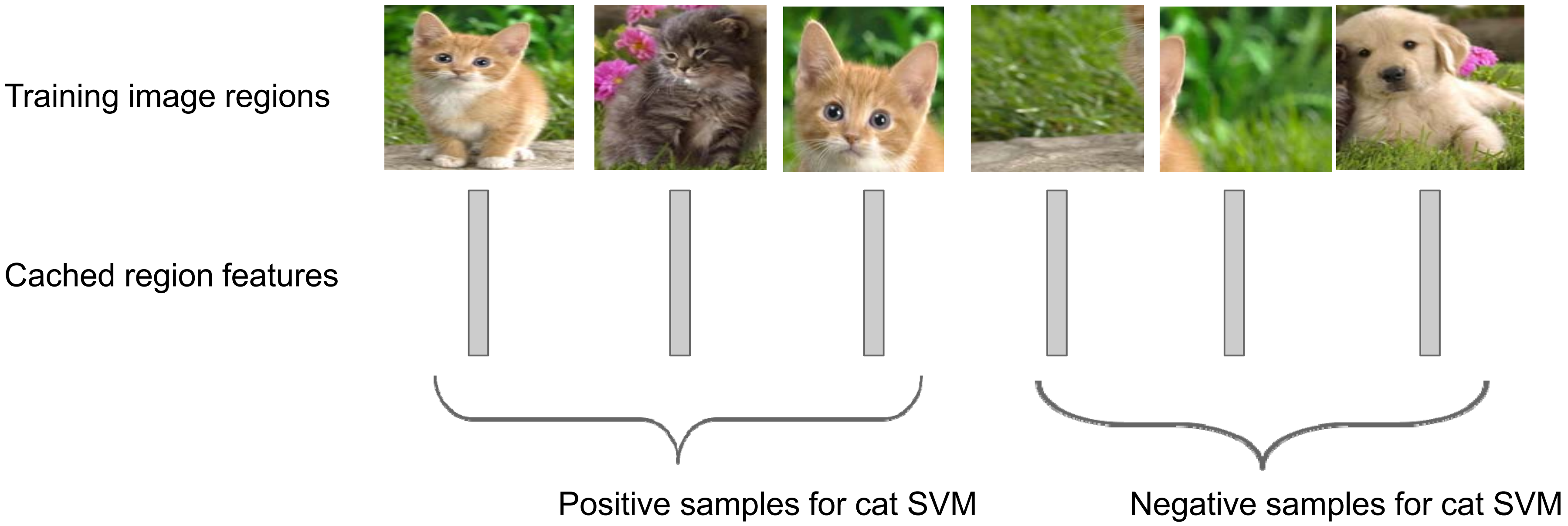
Step 3: Extract features
-Extract region proposals for all images
-For each region: warp to CNN input size, run forward through CNN, save pool5 features to disk
-Have a big hard drive: features are ~200GB for PASCAL dataset!

Convolution and Pooling



R-CNN Training

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

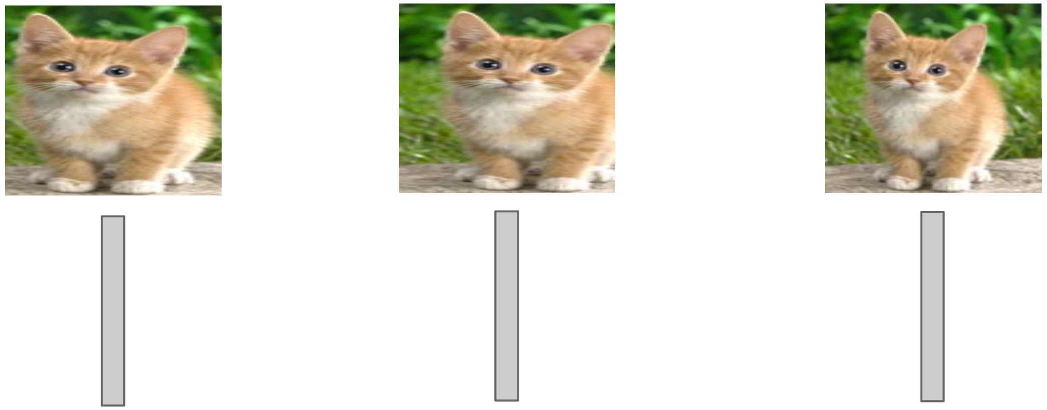


R-CNN Training

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



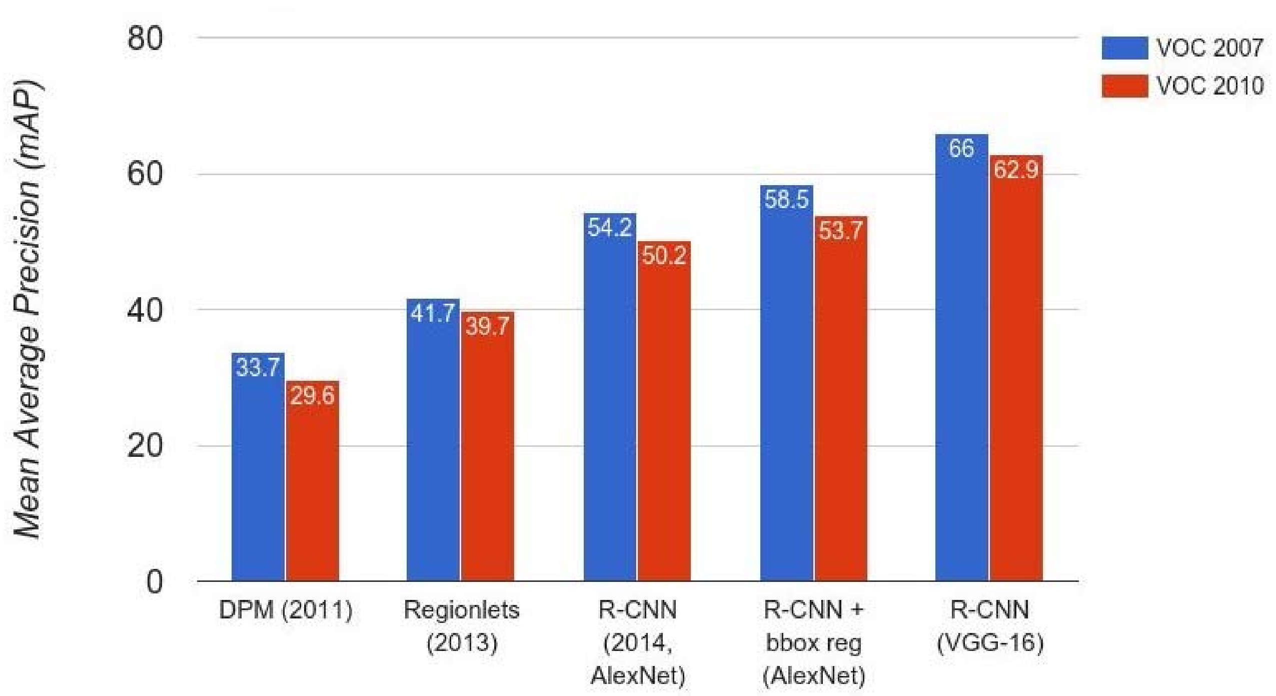
Regression targets
(dx, dy, dw, dh)
Normalized coordinates

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

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

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

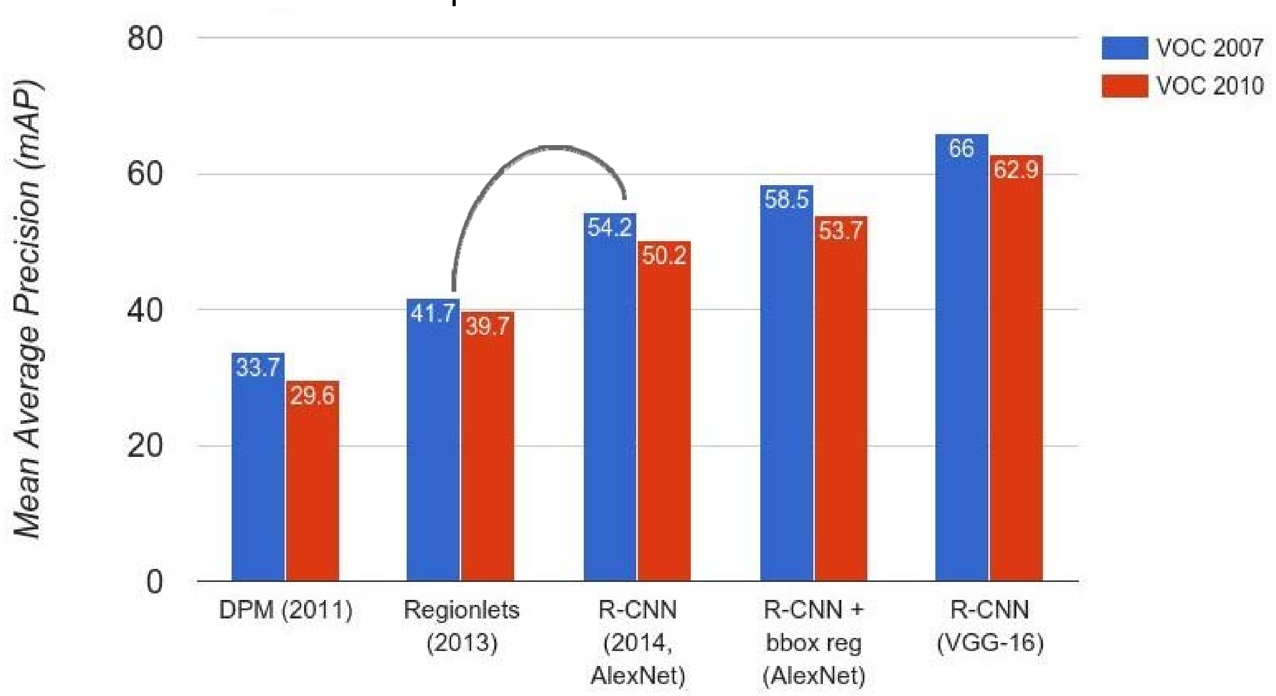
R-CNN Results



Regionlets for generic object detection, Wang et al., ICCV 2013
Object detection with discriminatively trained part based models, Felzenszwalb et al., PAMI 2011

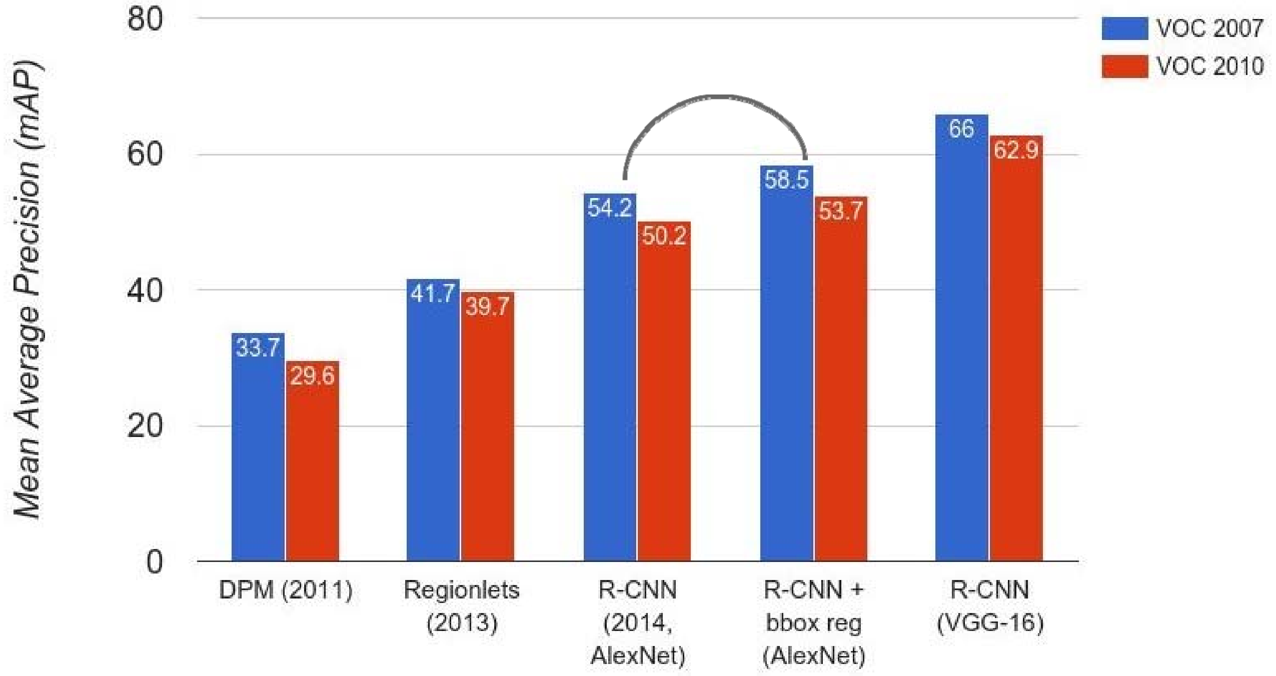
R-CNN Results

Big improvement compared
to pre-CNN methods



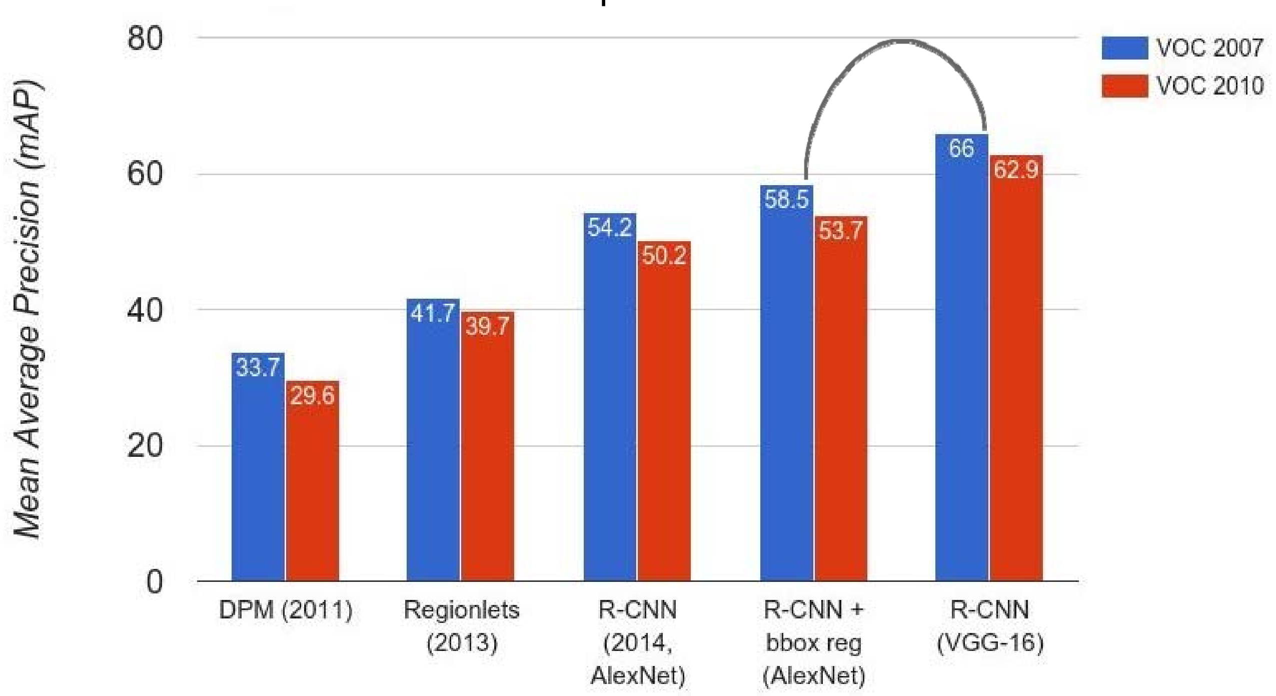
R-CNN Results

Bounding box regression helps a bit

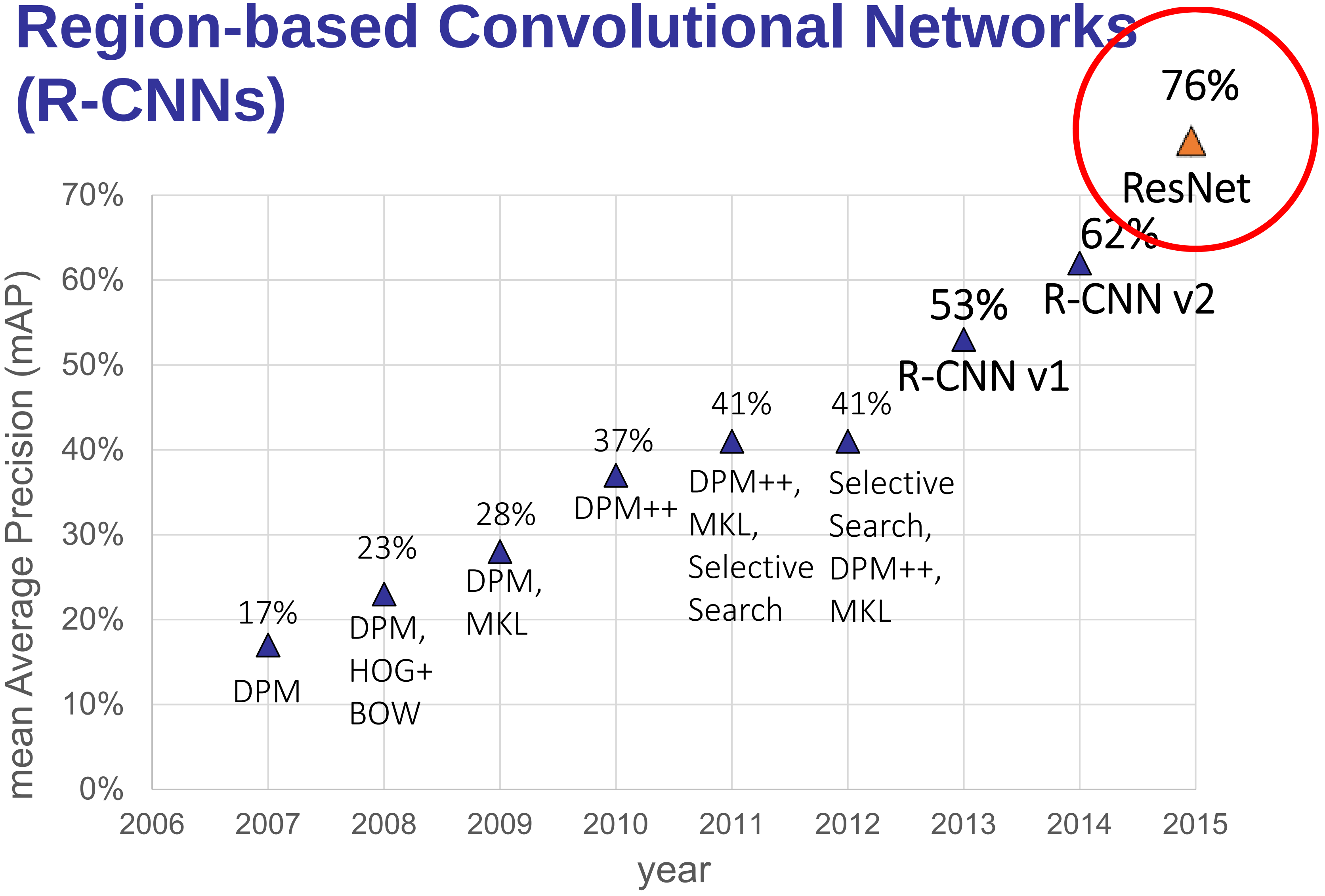


R-CNN Results

Features from a deeper network help a lot



Region-based Convolutional Networks (R-CNNs)



[R-CNN. Girshick et al. CVPR 2014]

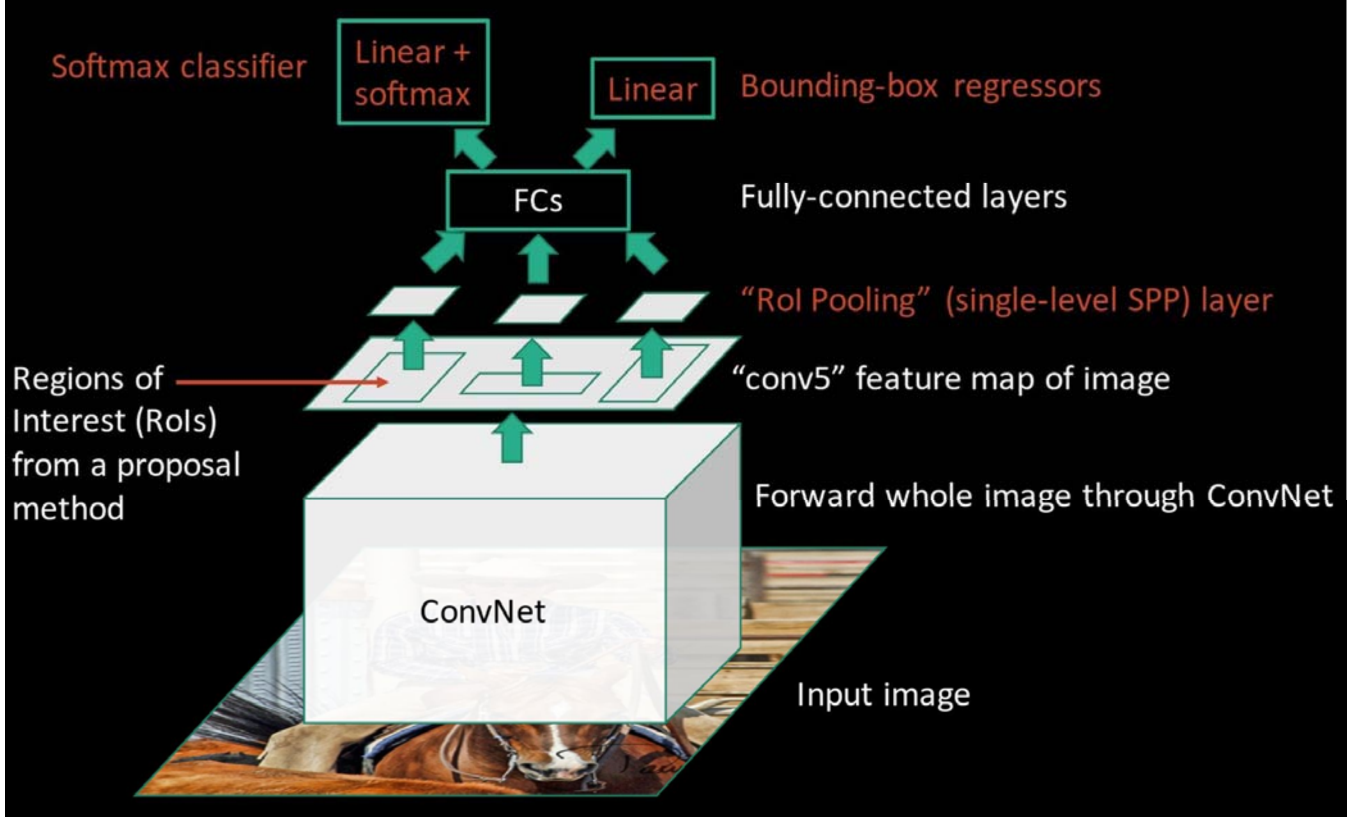
R-CNN Problems

1. Slow at test-time: need to run full forward pass of CNN for each region proposal

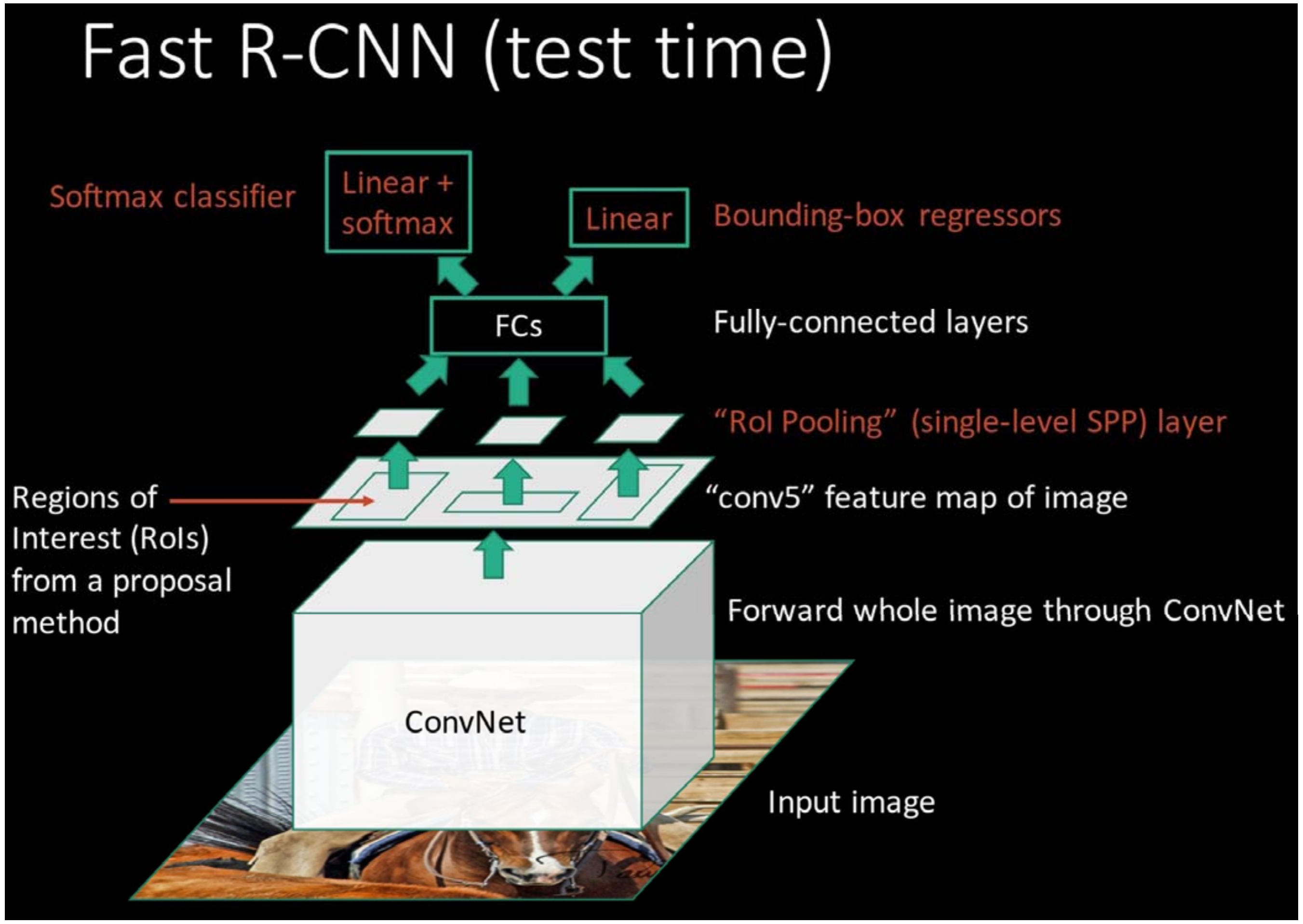
2. SVMs and regressors are post-hoc: CNN features not updated in response to SVMs and regressors

3. Complex multistage training pipeline

Fast R-CNN (test time)

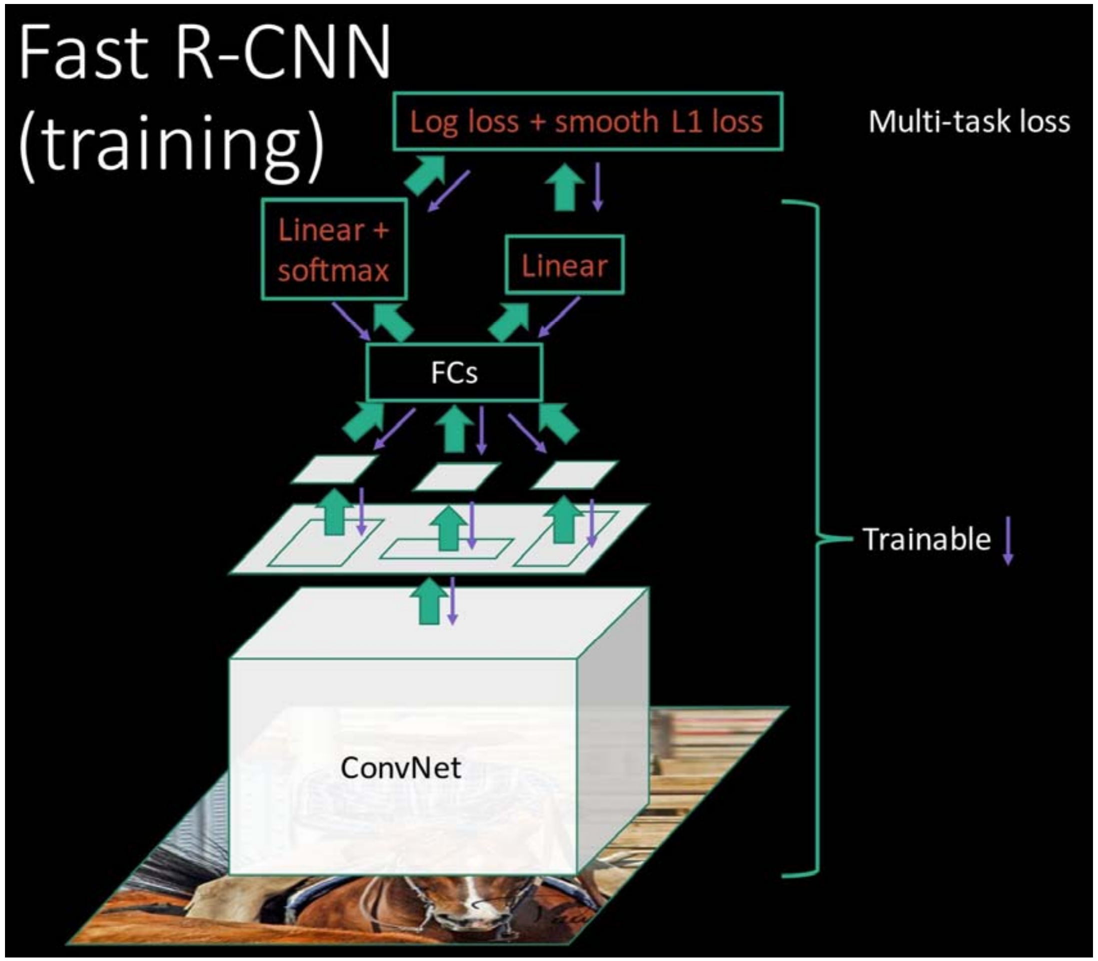

Softmax classifier
Linear + softmax
Linear
Bounding-box regressors
FCs
Fully-connected layers
"RoI Pooling" (single-level SPP) layer
Regions of Interest (RoIs) from a proposal method
"conv5" feature map of image
Forward whole image through ConvNet
ConvNet
Input image

[Girschick, "Fast R-CNN", ICCV 2015]


Fast R-CNN (test time)
Softmax classifier
Linear + softmax
Linear
Bounding-box regressors
FCs
Fully-connected layers
"RoI Pooling" (single-level SPP) layer
"conv5" feature map of image
Regions of Interest (RoIs) from a proposal method
Forward whole image through ConvNet
ConvNet
Input image

R-CNN Problem #1: Slow at test-time due to independent forward passes of the CNN

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



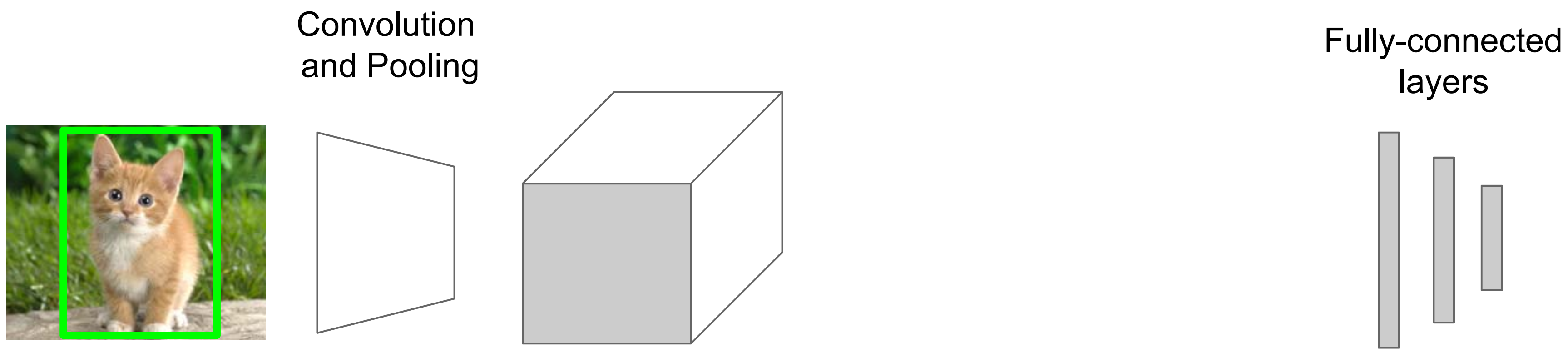
R-CNN Problem #2:
Post-hoc training: CNN not updated in response to final classifiers and regressors

R-CNN Problem #3:
Complex training pipeline

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

Slide credit: Ross Girschick

Fast R-CNN: Region of Interest Pooling

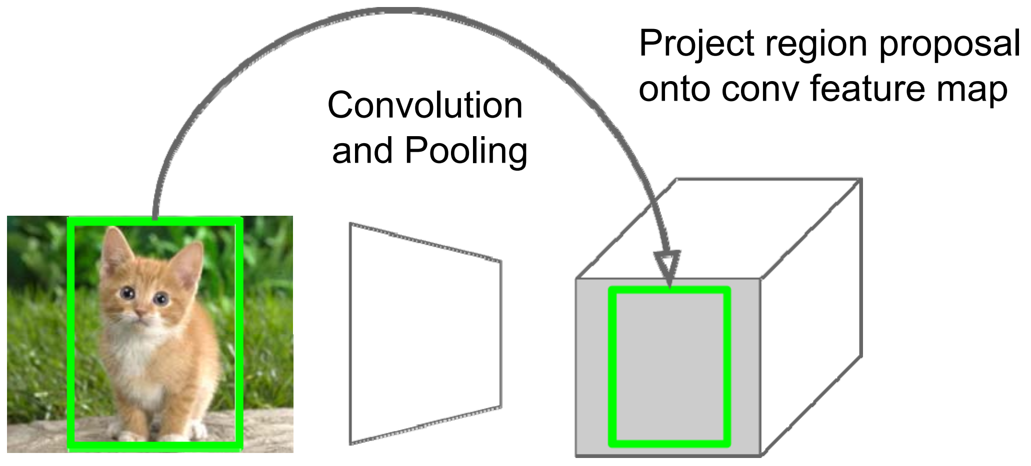


Hi-res input image:
3 x 800 x 600
with region
proposal

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

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

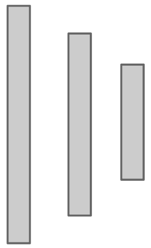
Fast R-CNN: Region of Interest Pooling



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

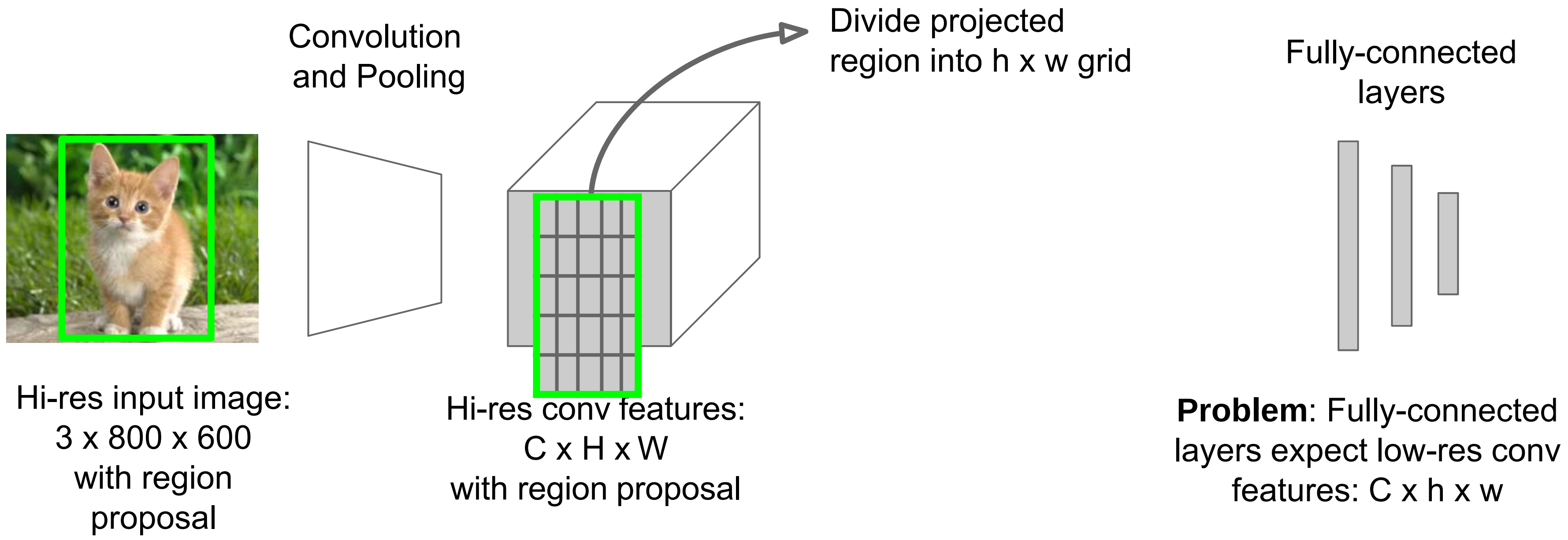
Hi-res conv features:
C x H x W
with region proposal

Fully-connected
layers

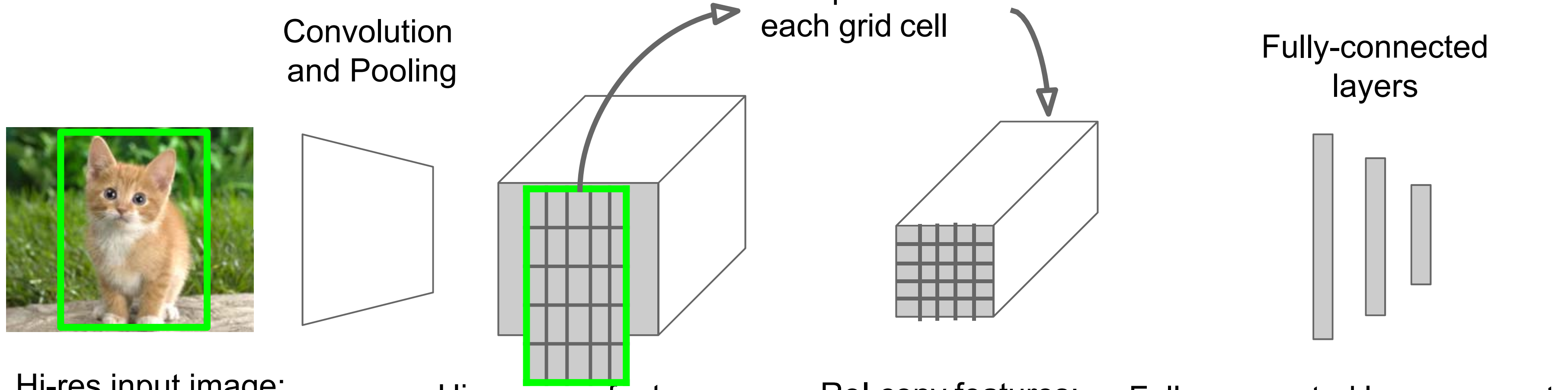


Problem: Fully-connected layers expect low-res conv features: C x h x w

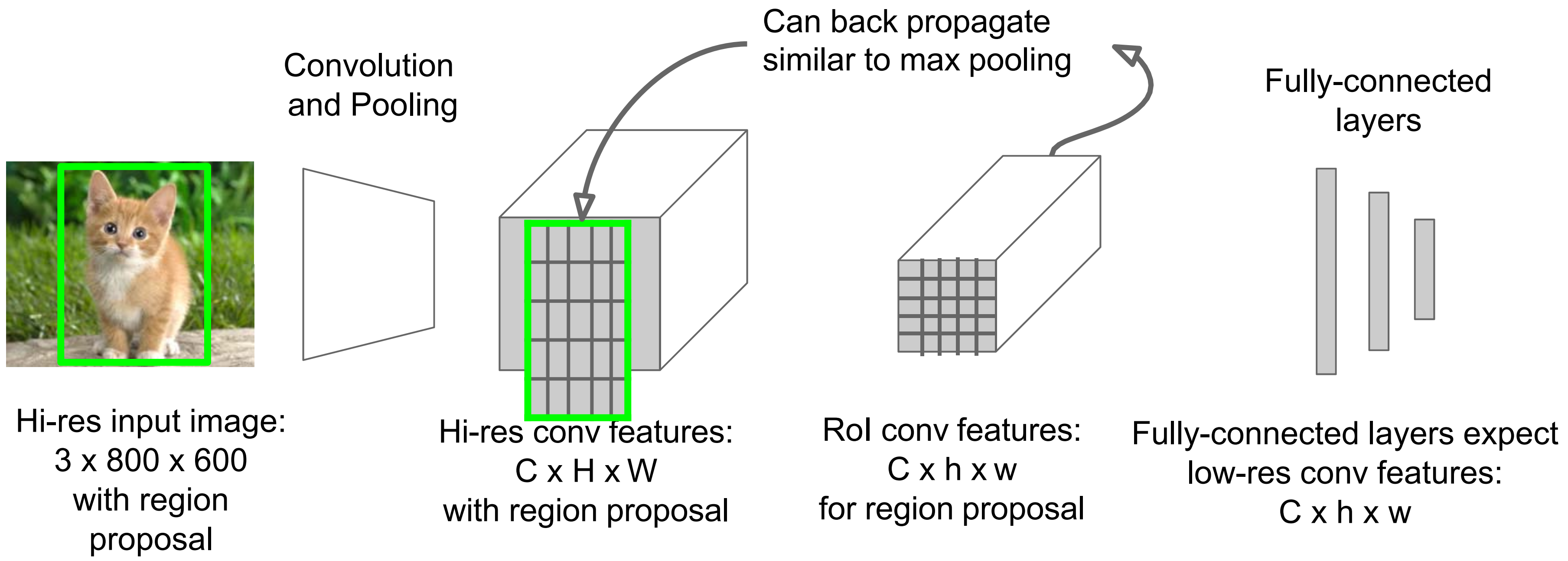
Fast R-CNN: Region of Interest Pooling


Convolution and Pooling
Divide projected region into h x w grid
Fully-connected layers
Hi-res input image: 3 x 800 x 600 with region proposal
Hi-res conv features: C x H x W with region proposal
Problem: Fully-connected layers expect low-res conv features: C x h x w

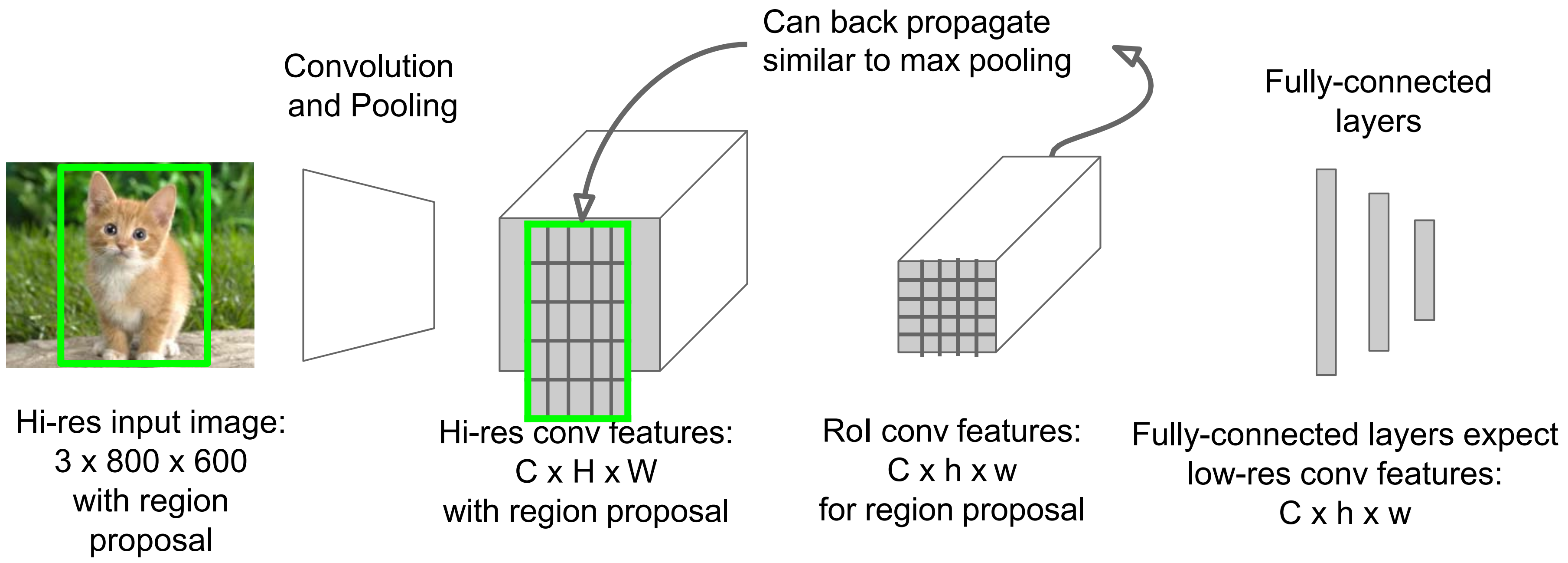
Fast R-CNN: Region of Interest Pooling


Convolution and Pooling
Max-pool within each grid cell
Fully-connected layers
Hi-res input image: 3 x 800 x 600 with region proposal
Hi-res conv features: C x H x W with region proposal
RoI conv features: C x h x w for region proposal
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


Convolution and Pooling
Can back propagate similar to max pooling
Fully-connected layers
Hi-res input image: 3 x 800 x 600 with region proposal
Hi-res conv features: C x H x W with region proposal
RoI conv features: C x h x w for region proposal
Fully-connected layers expect low-res conv features: C x h x w

Multi-task loss: $L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1] L_{\text{loc}}(t^u, v)$

Classification:

$$L_{\text{cls}}(p, u) = -\log p_u$$

Localization:

$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{\text{x,y,w,h}\}} \text{smooth}_{L_1}(t_i^u - v_i)$$

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

Fast R-CNN Results

	R-CNN	**Fast R-CNN**
Training Time:	84 hours	**9.5 hours**
(Speedup)	1x	**8.8x**

Faster!

Using VGG-16 CNN on Pascal VOC 2007 dataset

Fast R-CNN Results

	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**

Faster!

FASTER!

Using VGG-16 CNN on Pascal VOC 2007 dataset

Fast R-CNN Results

	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**

Faster!

FASTER!

Better!

Using VGG-16 CNN on Pascal VOC 2007 dataset

Fast R-CNN Problem:

Test-time speeds don't include region proposals

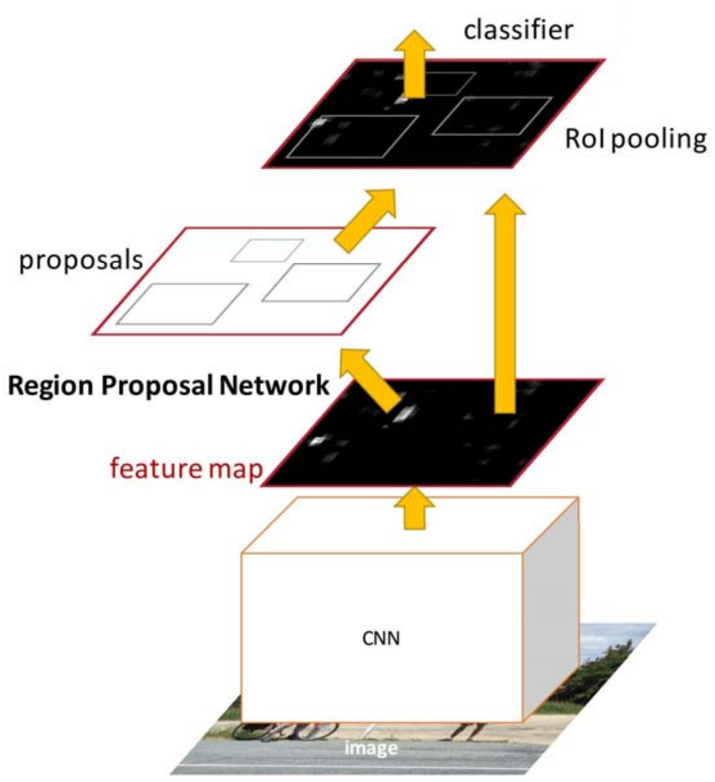
	R-CNN	**Fast R-CNN**
Test time per image	47 seconds	**0.32 seconds**
(Speedup)	1x	**146x**
Test time per image with Selective Search	50 seconds	**2 seconds**
(Speedup)	1x	**25x**

Fast R-CNN Problem Solution:

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

	R-CNN	**Fast R-CNN**
Test time per image	47 seconds	**0.32 seconds**
(Speedup)	1x	**146x**
Test time per image with Selective Search	50 seconds	**2 seconds**
(Speedup)	1x	**25x**

Faster R-CNN:



Insert a **Region Proposal Network (RPN)** after the last convolutional layer

RPN trained to produce region proposals directly; no need for external region proposals!

After RPN, use RoI Pooling and an upstream classifier and bbox regressor just like Fast R-CNN

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

Slide credit: Ross Girschick

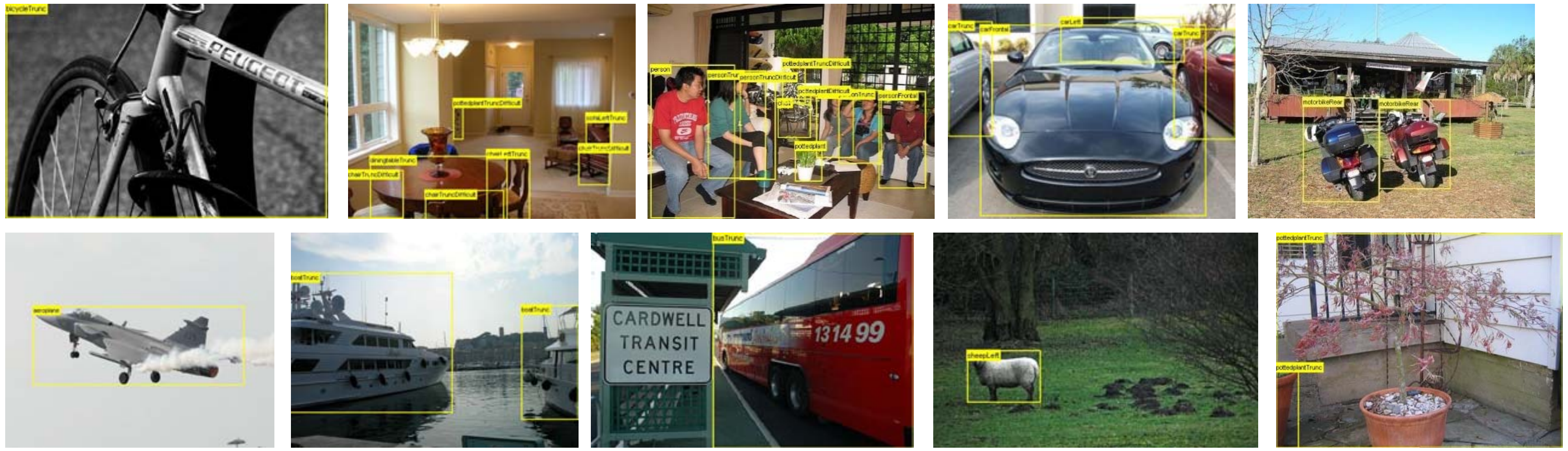
Outline

1. Sliding window detectors
2. Features and adding spatial information
3. HOG + linear SVM classifier
4. State of the art algorithms
5. *PASCAL VOC and MSR Coco*

PASCAL VOC dataset - Content

- 20 classes: aeroplane, bicycle, boat, bottle, bus, car, cat, chair, cow, dining table, dog, horse, motorbike, person, potted plant, sheep, train, TV

- Real images downloaded from flickr, not filtered for “quality”



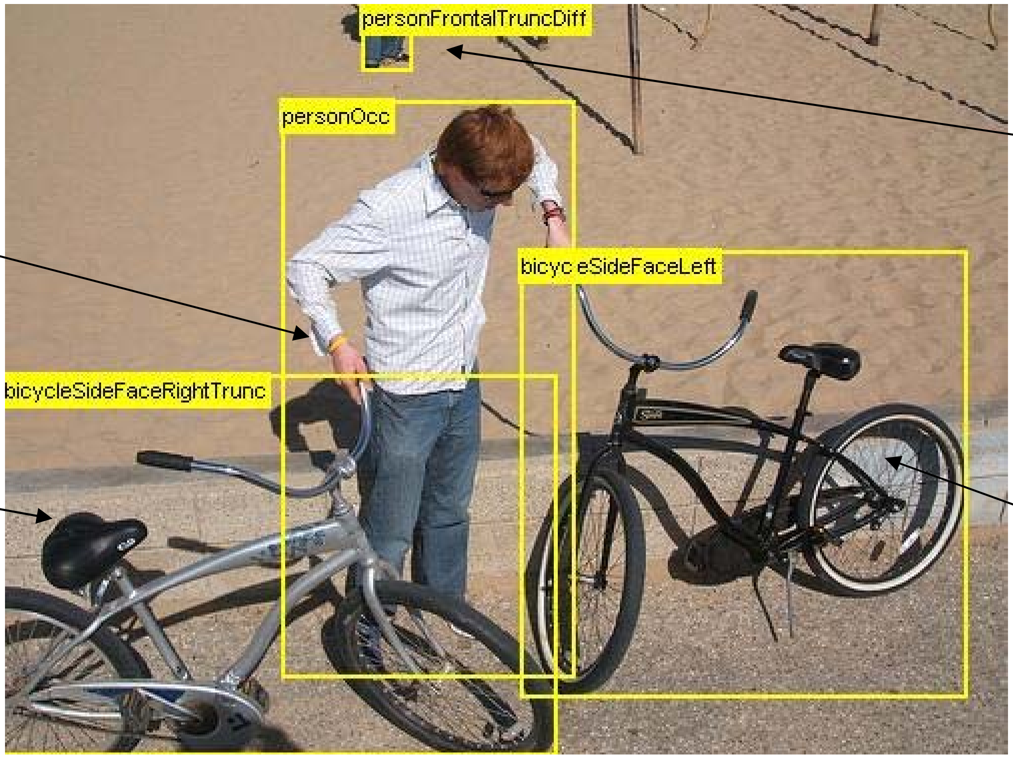
- Complex scenes, scale, pose, lighting, occlusion, ...

Annotation

- Complete annotation of all objects

Occluded
Object is significantly occluded within BB

Difficult
Not scored in evaluation



Truncated
Object extends beyond BB

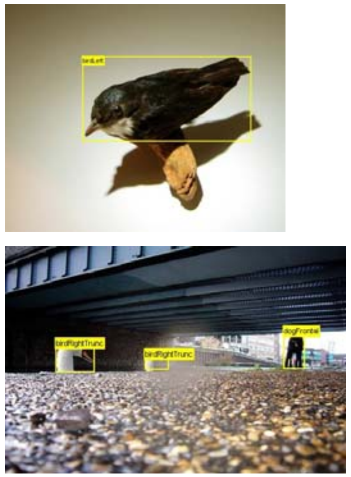
Pose
Facing left

Examples

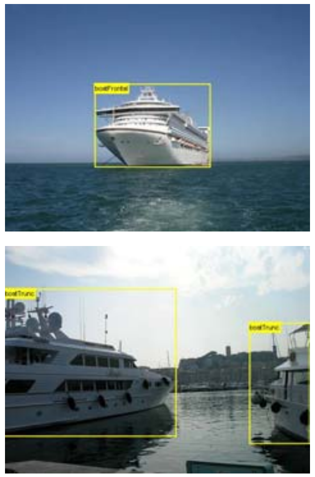
Aeroplane

Bicycle

Bird



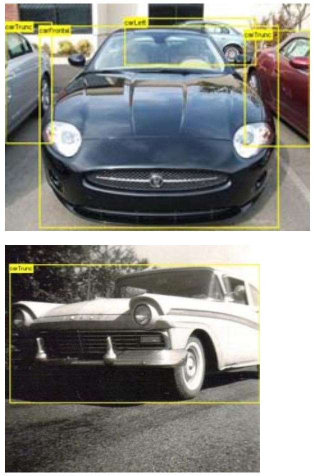
Boat



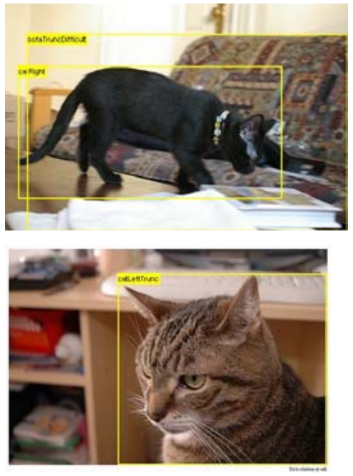
Bottle

Bus

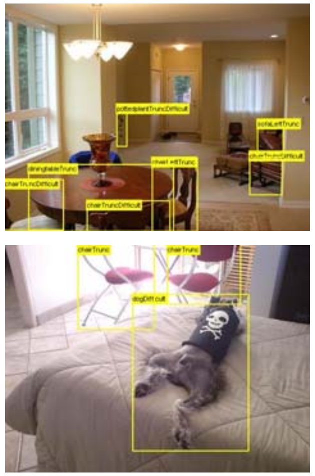
Car



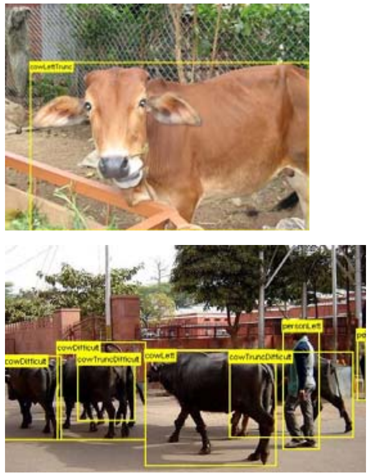
Cat



Chair

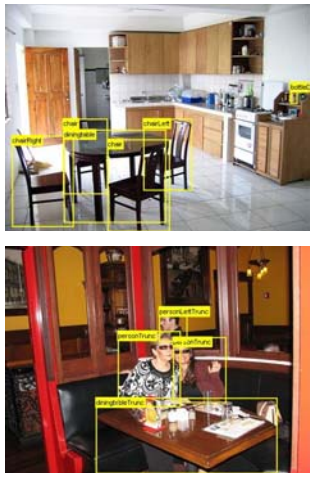


Cow

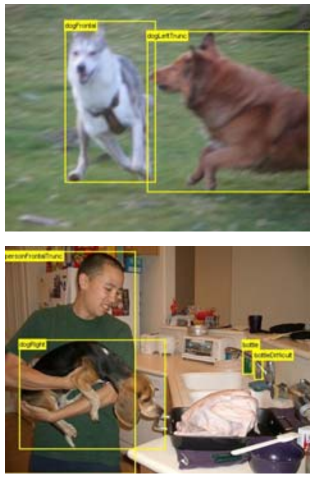


Examples

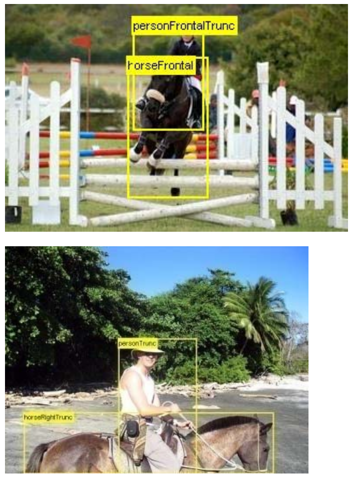
Dining Table



Dog

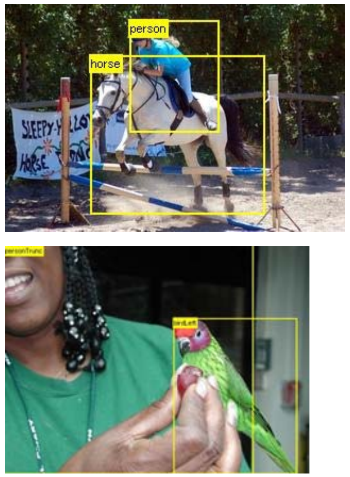


Horse



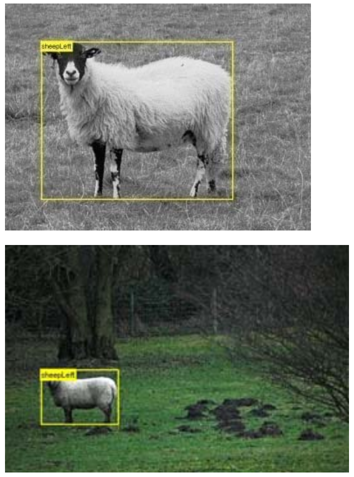
Motorbike

Person

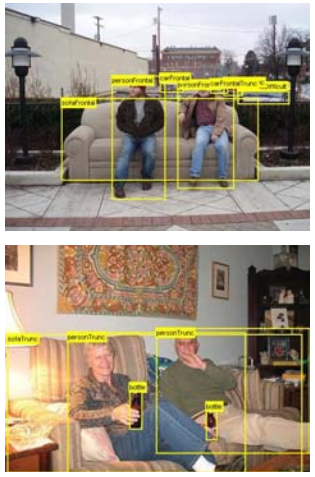


Potted Plant

Sheep



Sofa

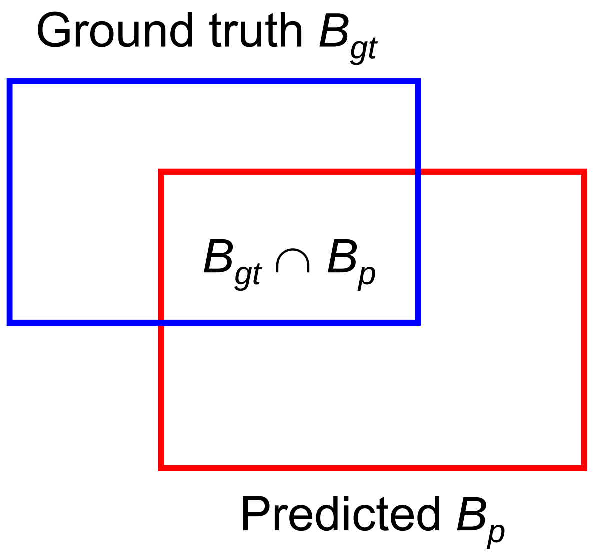


Train

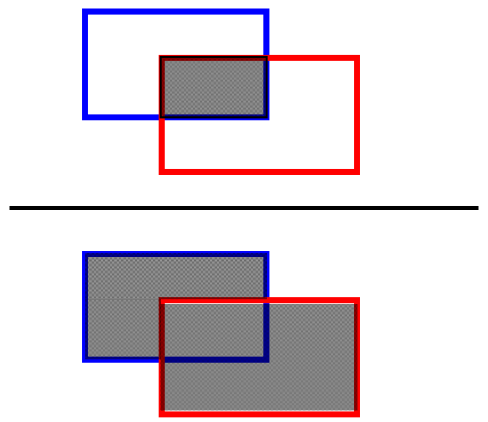
TV/Monitor

Detection: Evaluation of Bounding Boxes

- Area of Overlap (AO) Measure



$$AO(B_{gt}, B_p) = \frac{|B_{gt} \bigcap B_p|}{|B_{gt} \bigcup B_p|}$$

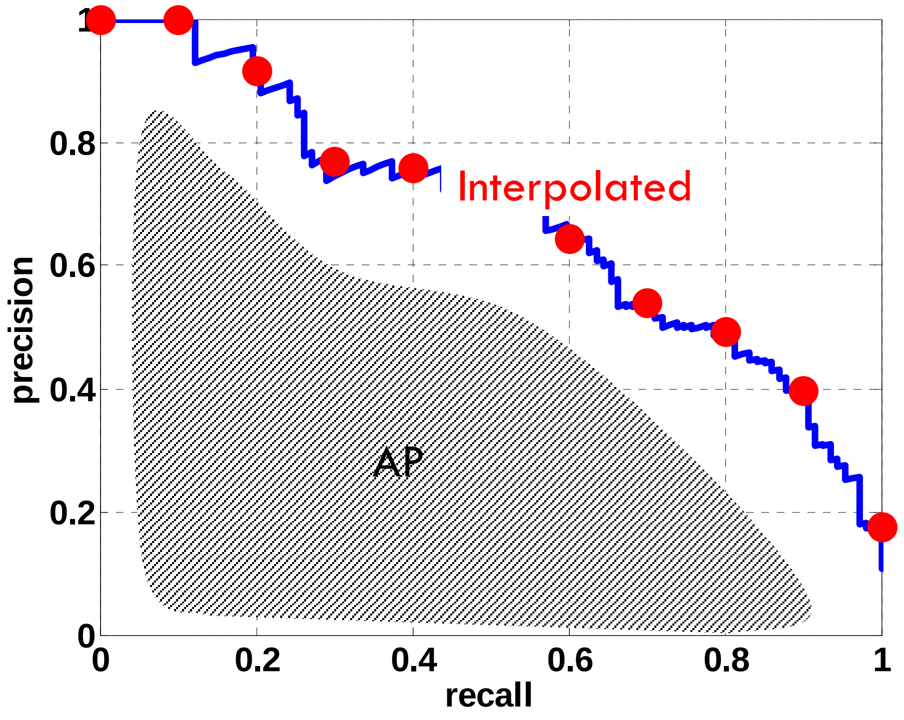


Detection if (figure) > Threshold

50%

Classification/Detection Evaluation

- Average Precision [TREC] averages precision over the entire range of recall

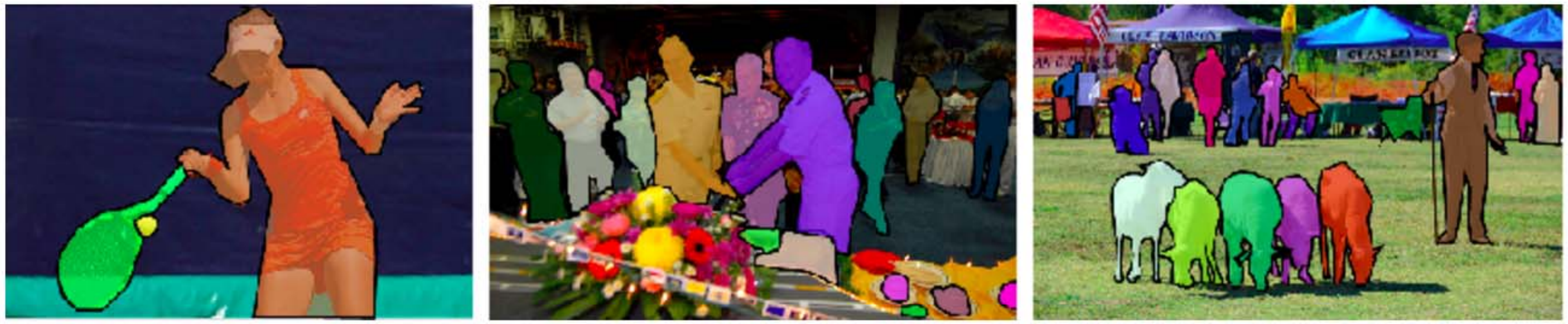


- A good score requires both high recall and high precision
- Application-independent
- Penalizes methods giving high precision but low recall

From Pascal to COCO: Common objects in context dataset

Welcome to the MS COCO Detection Challenge!

Winners to be announced at ICCV 2015



1. Overview

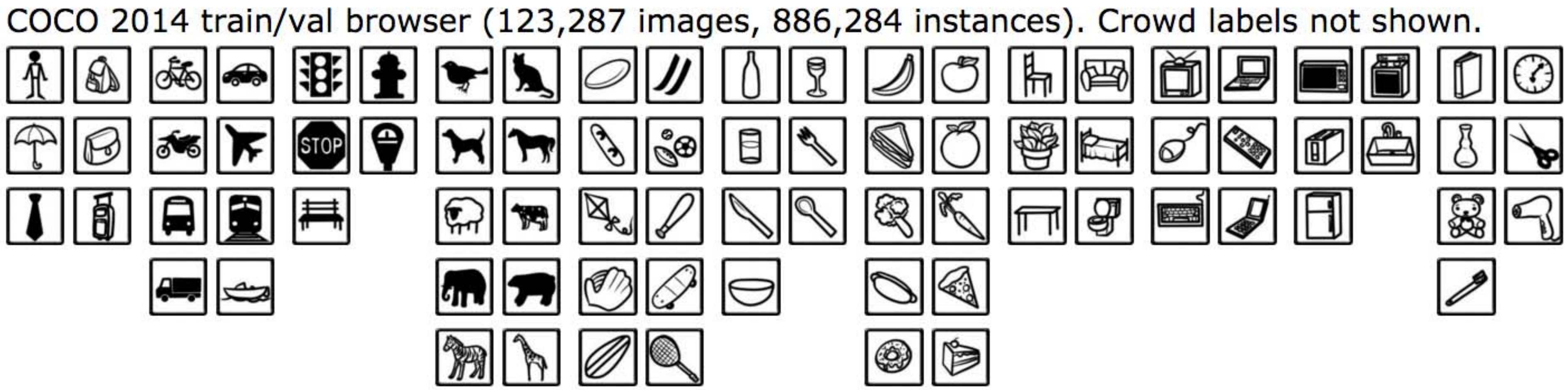
We are pleased to announce the MS COCO 2015 Detection Challenge. This competition is designed to push the state of the art in object detection forward. Teams are encouraged to compete in either (or both) of two object detection challenges: using bounding box output or object segmentation output.

The MS COCO train, validation, and test sets, containing more than 200,000 images and 80 object categories, are available on the download page. All object instance are annotated with a detailed segmentation mask. Annotations on the training and validation sets (with over 500,000 object instances segmented) are publicly available.

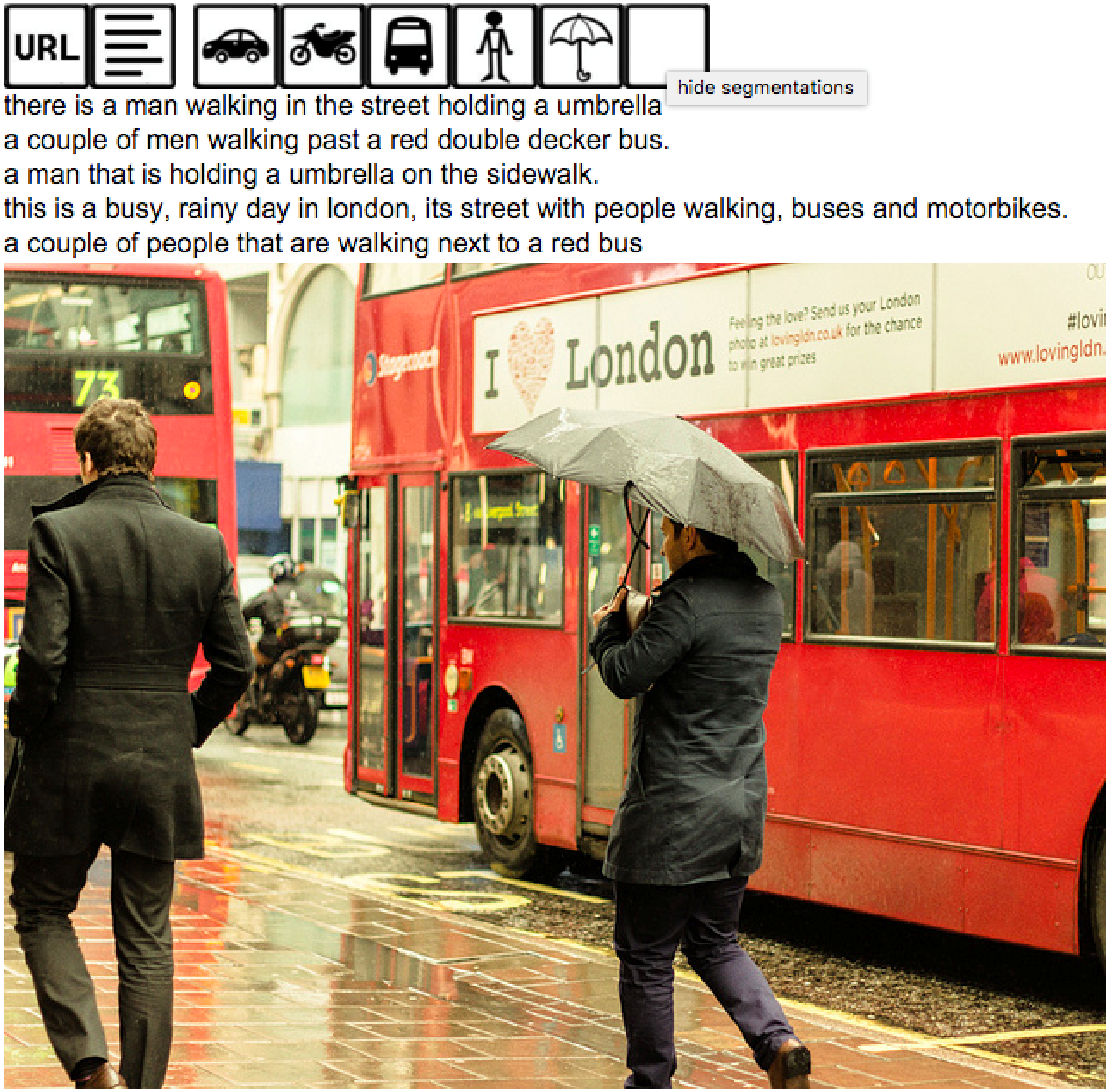
[Lin et al., 2015] http://mscoco.org/

Dataset statistics

- 80 object classes


COCO 2014 train/val browser (123,287 images, 886,284 instances). Crowd labels not shown.

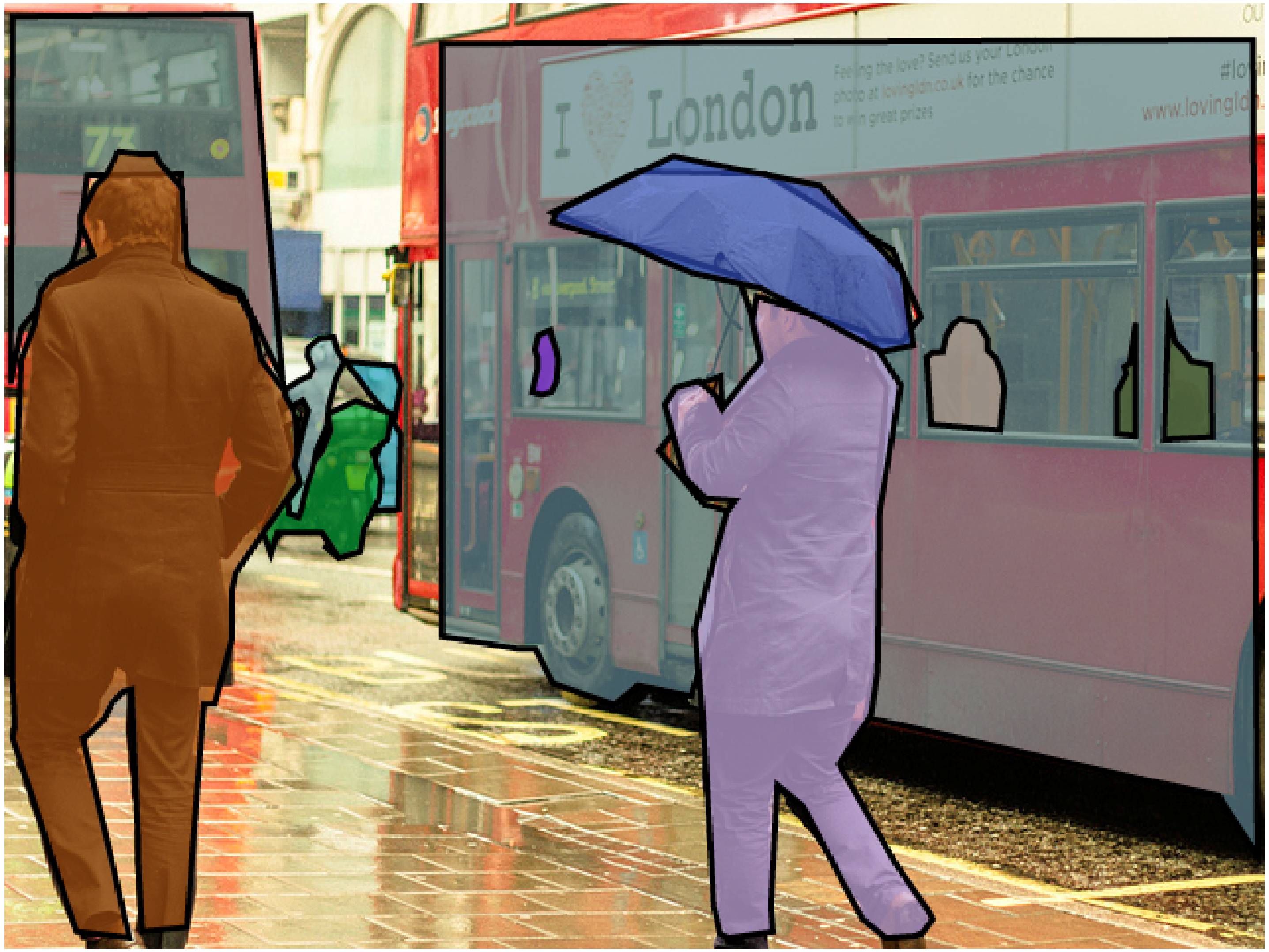
- 80k training images
- 40k validation images
- 80k testing images


URL
hide segmentations
73
Stagecoach
I London
Feeling the love? Send us your London photo at lovingldn.co.uk for the chance to win great prizes
#lovi
www.lovingldn.

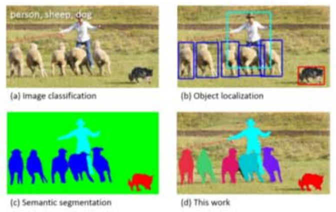
there is a man walking in the street holding a umbrella
a couple of men walking past a red double decker bus.
a man that is holding a umbrella on the sidewalk.
this is a busy, rainy day in london, its street with people walking, buses and motorbikes.
a couple of people that are walking next to a red bus

URL

there is a man walking in the street holding a umbrella
a couple of men walking past a red double decker bus.
a man that is holding a umbrella on the sidewalk.
this is a busy, rainy day in london, its street with people walking, buses and motorbikes.
a couple of people that are walking next to a red bus

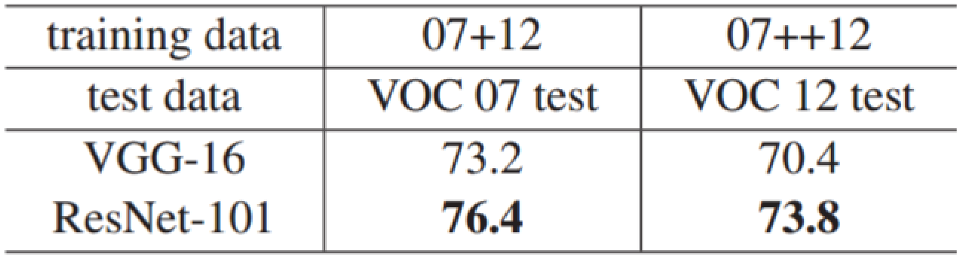

77
I London
Feeling the love? Send us your London photo at lovingldn.co.uk for the chance to win great prizes
www.lovingld

Towards object instance segmentation



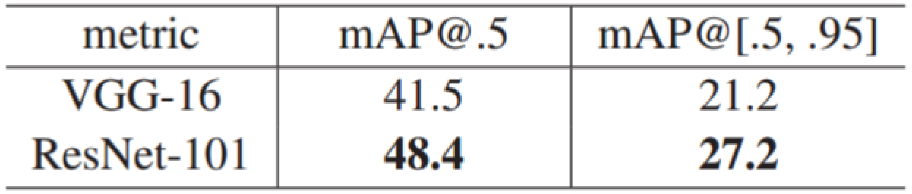
(a) Image classification
(b) Object localization
(c) Semantic segmentation
(d) This work

Object Detection State-of-the-art: ResNet 101 + Faster R-CNN + some extras



training data	07+12	07++12
test data	VOC 07 test	VOC 12 test
VGG-16	73.2	70.4
ResNet-101	**76.4**	**73.8**

AP (%) for Pascal VOC test sets (20 object classes)



metric	mAP@.5	mAP@[.5, .95]
VGG-16	41.5	21.2
ResNet-101	**48.4**	**27.2**

AP (%) for COCO validation set (80 object classes)

[He et. al, “Deep Residual Learning for Image Recognition”, CVPR 2016]
CVPR 2016 Best Paper Award

Summary of object detection

- Basic idea: train a sliding window classifier from training data
- Histogram of oriented gradients (HOG) features + linear SVM
  - Jittering, hard negative mining improve accuracy
- Region proposals using selective search
- R-CNN: combine region proposals and CNN features
- Fast(er) R-CNN: end-to-end training
  - Region proposals and object classification can be trained jointly
  - Deeper networks (ResNet101) improve accuracy