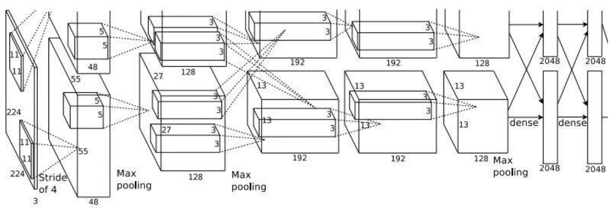
CS 2770: Computer Vision

Object Detection

Prof. Adriana Kovashka
University of Pittsburgh
March 16, 2021

So far: Image Classification



Vector: 4096

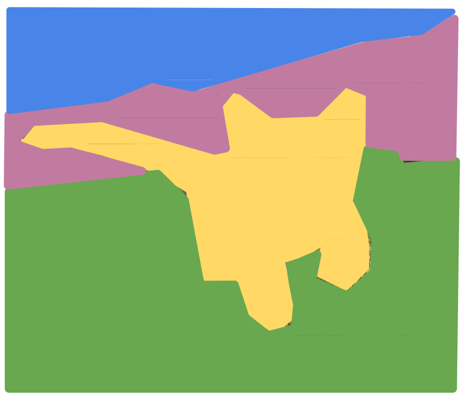
Fully-Connected: 4096 to 1000

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

Slide by: Justin Johnson

Other Computer Vision Tasks

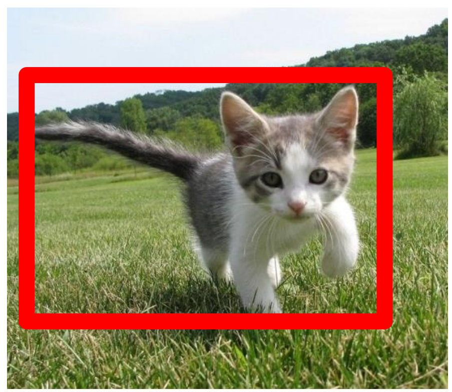
Semantic Segmentation



GRASS, CAT, TREE, SKY

No objects, just pixels

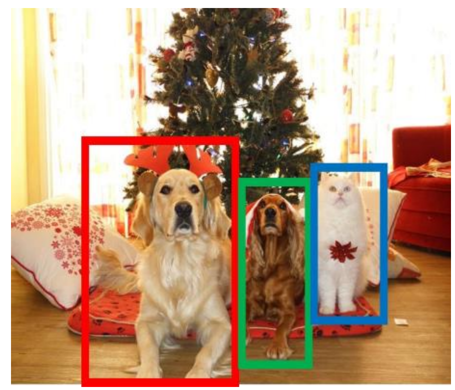
Classification + Localization



CAT

Single Object

Object Detection



DOG, DOG, CAT

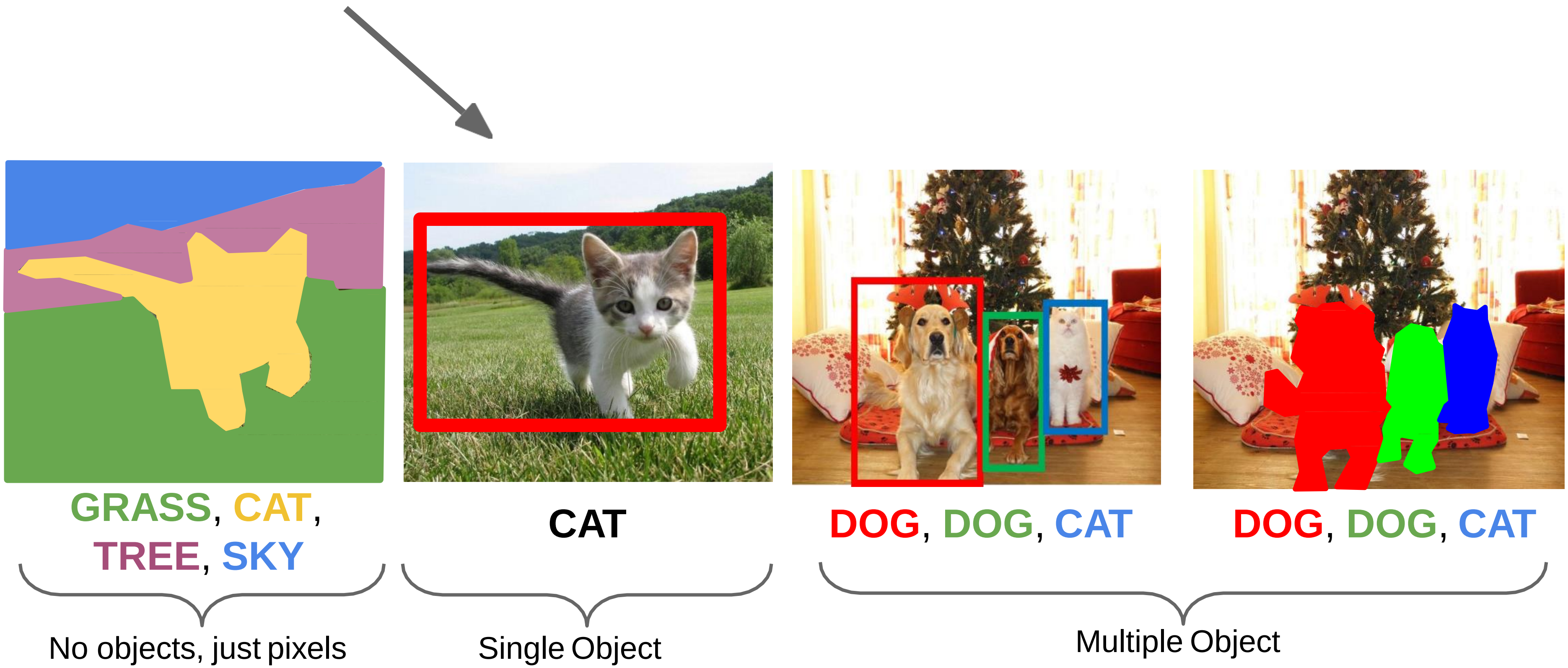
Instance Segmentation

DOG, DOG, CAT

Multiple Object

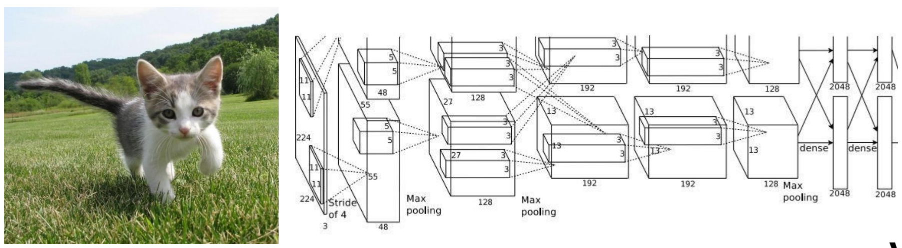
Slide by: Justin Johnson

Classification + Localization



Slide by: Justin Johnson

Classification + Localization



Fully Connected:
4096 to 1000

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

Vector:
4096

Fully Connected:
4096 to 4

Box Coordinates
(x, y, w, h)

Treat localization as a regression problem!

Slide by: Justin Johnson

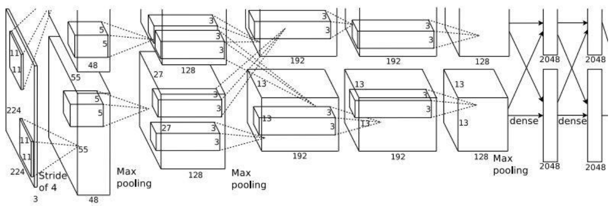
Classification + Localization

Correct label:
Cat

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

Softmax Loss

Fully Connected:
4096 to 1000



Vector:
4096

Fully Connected:
4096 to 4

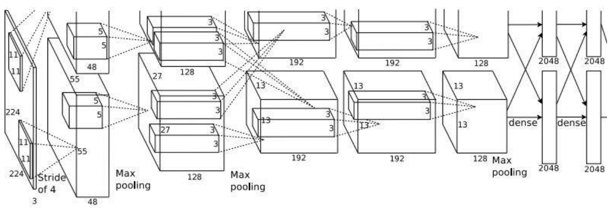
Box Coordinates
(x, y, w, h)

L2 Loss

Correct box:
(x’, y’, w’, h’)

Treat localization as a regression problem!

Classification + Localization



Vector: 4096

Fully Connected: 4096 to 1000

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

Correct label:
Cat

Softmax Loss

Fully Connected: 4096 to 4

Box Coordinates
(x, y, w, h)

L2 Loss

Correct box:
(x', y', w', h')

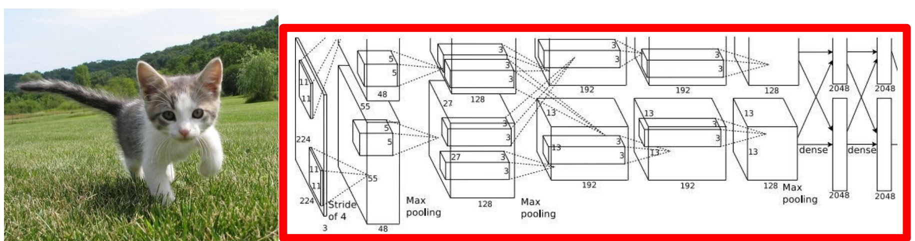
Multitask Loss

+ → **Loss**

Treat localization as a regression problem!

Slide by: Justin Johnson

Classification + Localization



Often pretrained on ImageNet
(Transfer learning)

Vector:
4096

Fully Connected:
4096 to 1000

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

Correct label:
Cat

Softmax Loss

Fully Connected:
4096 to 4

Box Coordinates
(x, y, w, h)

L2 Loss

Correct box:
(x', y', w', h')

+ **Loss**

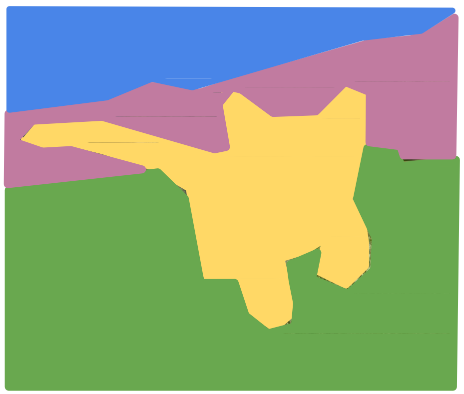
Treat localization as a regression problem!

Slide by: Justin Johnson

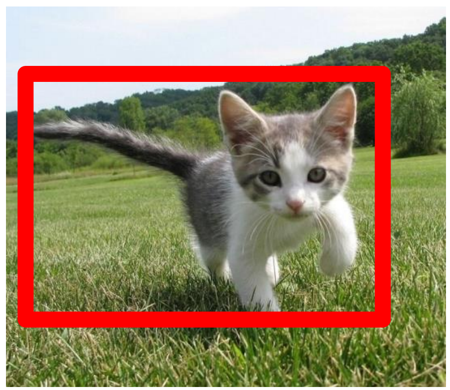
Plan for this lecture

- Fully supervised detection
  - Pre-CNN: Deformable part models
  - Detection with region proposals: R-CNN, Fast/er R-CNN
  - Detection without region proposals: YOLO
  - Semantic and instance segmentation: FCN, Mask R-CNN
- Weak or out-of-domain supervision
  - Weakly supervised object detection
  - Domain adaptation

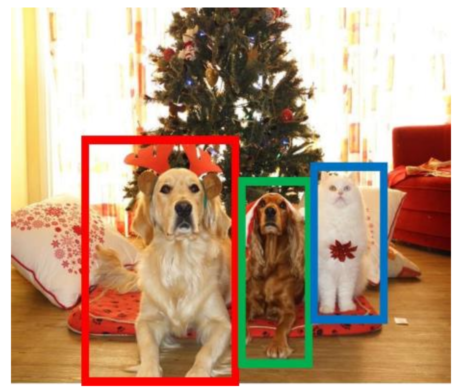
Object Detection



GRASS, CAT, TREE, SKY



CAT



DOG, DOG, CAT

DOG, DOG, CAT

No objects, just pixels

Single Object

Multiple Object

Slide by: Justin Johnson

Object detection: basic framework

- Build/train object model
- Generate candidate regions in new image
- Score the candidates

Adapted from Kristen Grauman

Window-template-based models
Building an object model

Given the representation, train a binary classifier

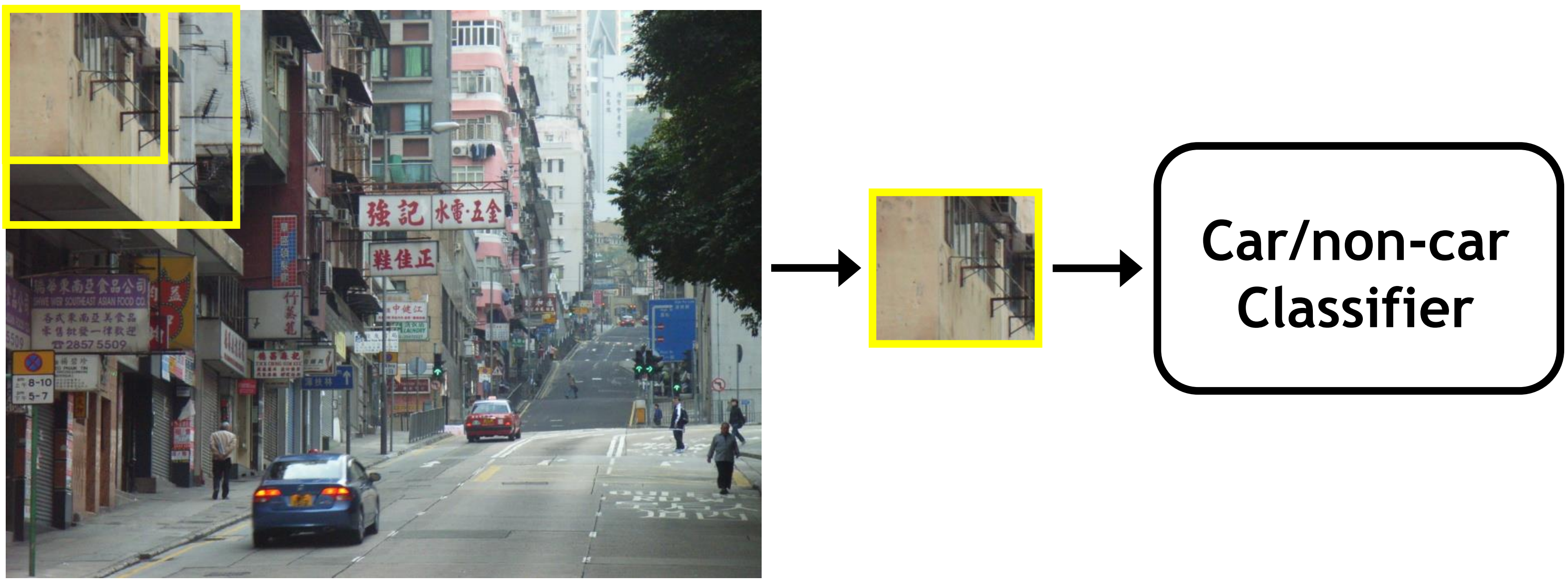
Car/non-car Classifier

No, not a car.

Yes, car.

Kristen Grauman

Window-template-based models
Generating and scoring candidates

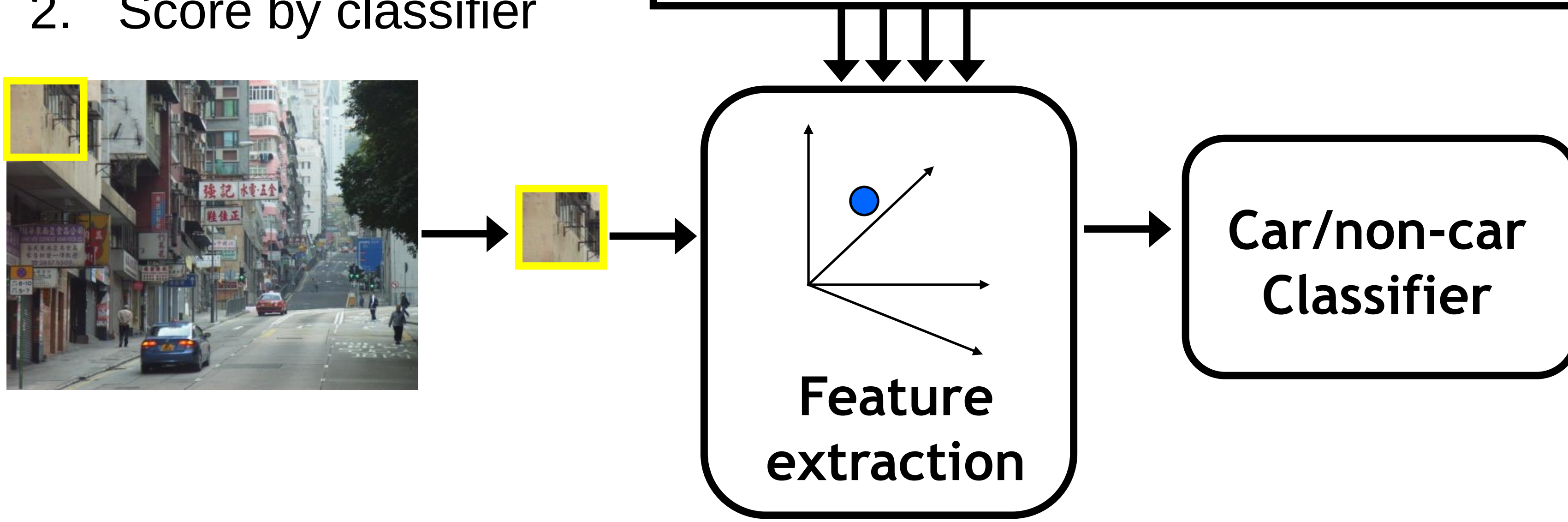


Kristen Grauman

Window-template-based object detection: recap

Training:
1. Obtain training data
2. Define features
3. Define classifier

Given new image:
1. Slide window
2. Score by classifier



Kristen Grauman

Dalal-Triggs pedestrian detector

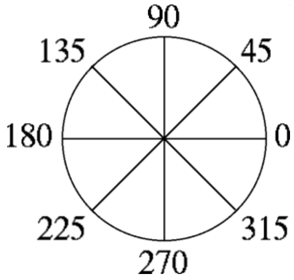
1. Extract fixed-sized (64x128 pixel) window at multiple positions and scales
2. Compute HOG (histogram of gradient) features within each window
3. Score the window with a linear SVM classifier
4. Perform non-maxima suppression to remove overlapping detections with lower scores

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

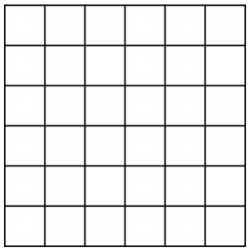
Histograms of oriented gradients (HOG)

Divide image into 8x8 regions

Orientation: 9 bins
(for unsigned angles)



Histograms in
8x8 pixel cells

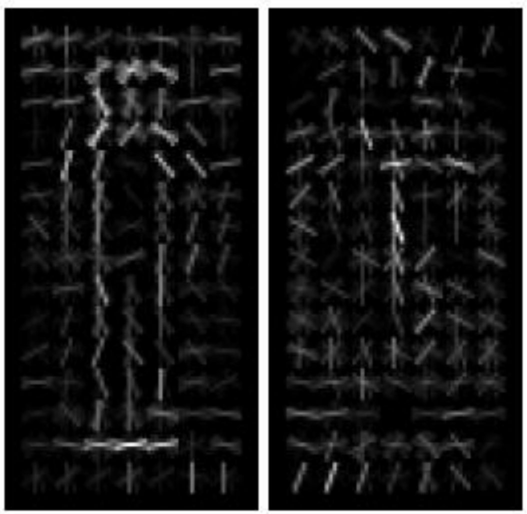


Votes weighted by magnitude

Adapted from Pete Barnum

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

Train SVM for pedestrian detection using HoG

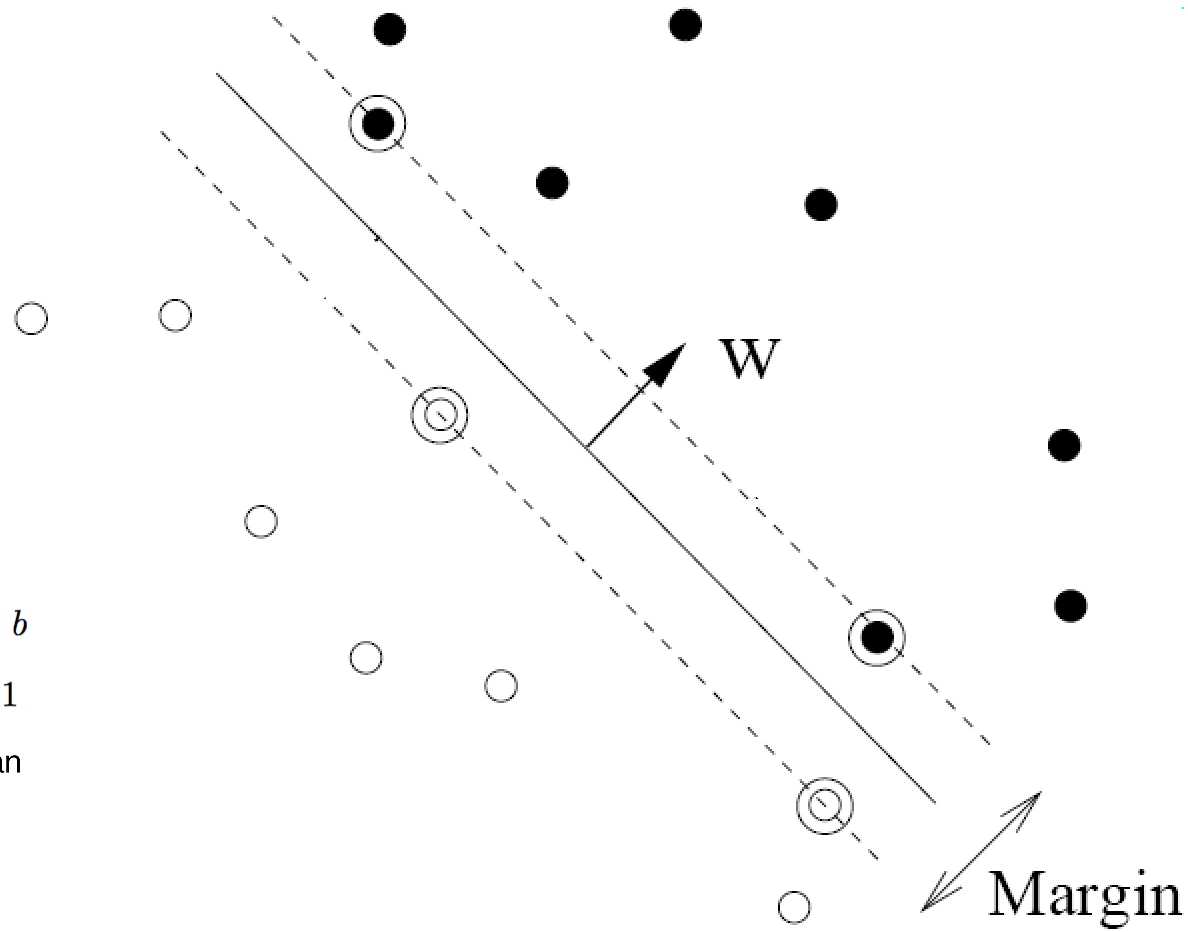


pos w   neg w

$$0.16 = w^T x + b$$

$$sign(0.16) = 1$$

=> pedestrian

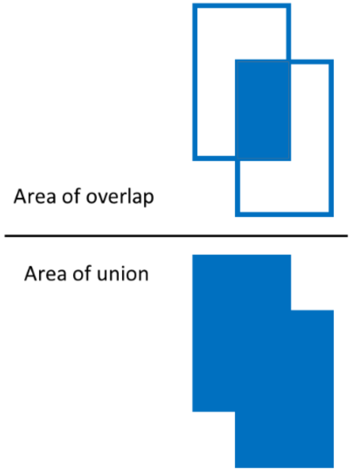


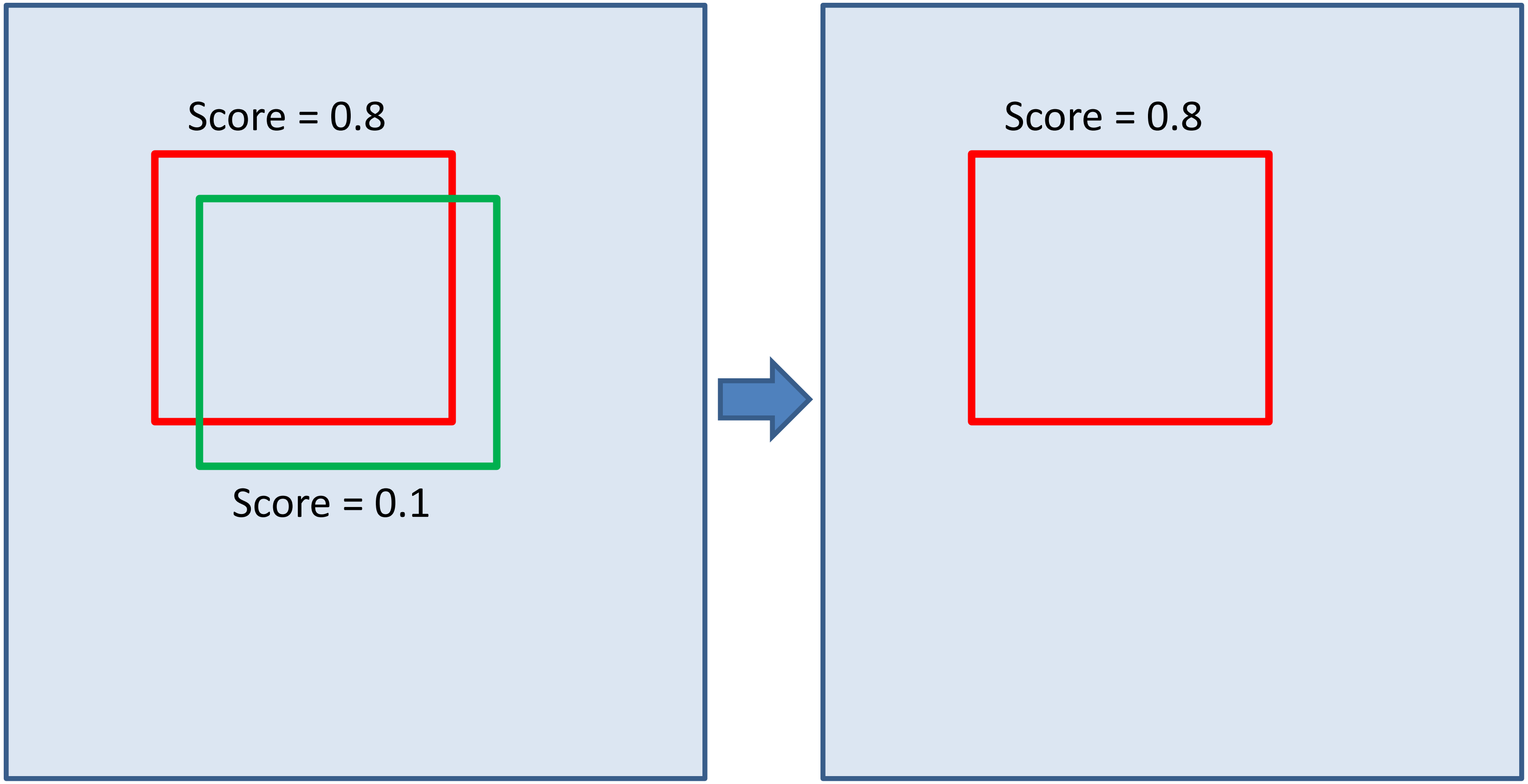
Adapted from Pete Barnum

Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

Remove overlapping detections

Non-max suppression





Adapted from Derek Hoiem

Are window templates enough?

- Many objects are articulated, or have parts that can vary in configuration

Images from Caltech-256, D. Ramanan

- Many object categories look very different from different viewpoints, or from instance to instance

Adapted from N. Snavely, D. Tran

Parts-based Models

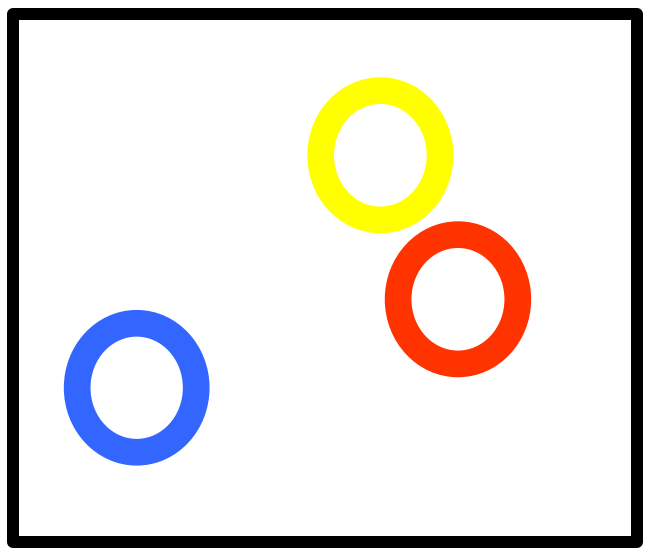
Define object by collection of parts modeled by

1. Appearance
2. Spatial configuration

Slide credit: Rob Fergus

How to model spatial relations?

- One extreme: fixed template



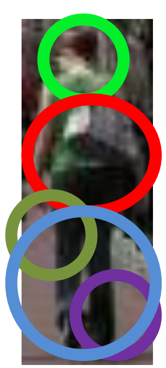
Derek Hoiem

Fixed part-based template

- Object model = sum of scores of features at fixed positions

$$+3 +2 \; -2 -1 \; -2.5 = -0.5 \overset{?}{>} 7.5$$

Non-object



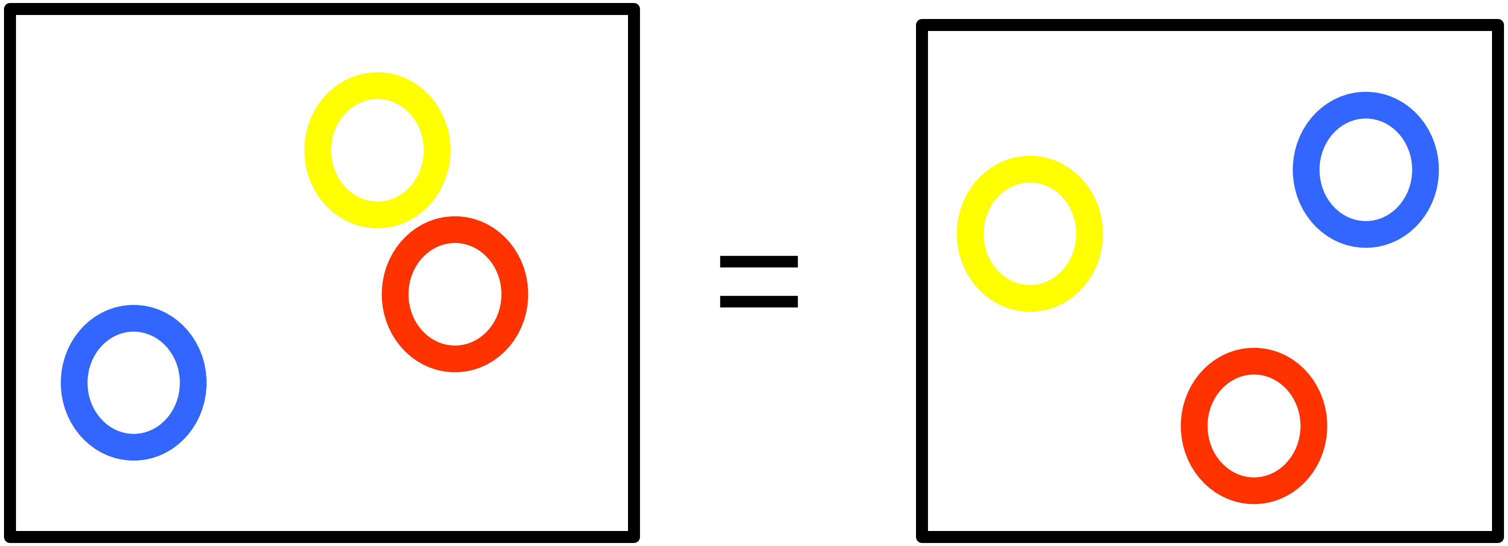
$$+4 +1 +0.5 \; +3 +0.5 = 10.5 \overset{?}{>} 7.5$$

Object

Derek Hoiem

How to model spatial relations?

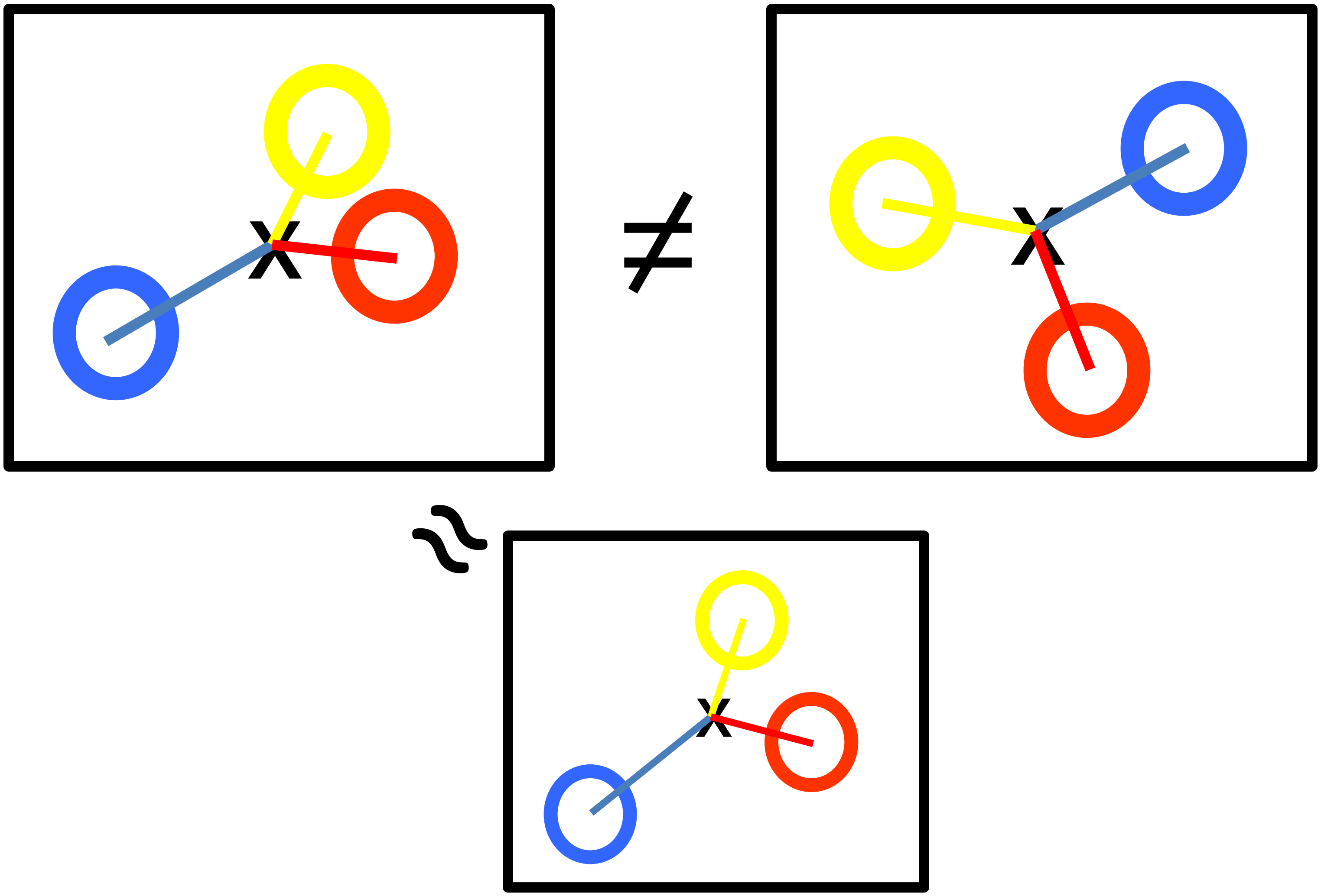
- Another extreme: bag of words



Derek Hoiem

How to model spatial relations?

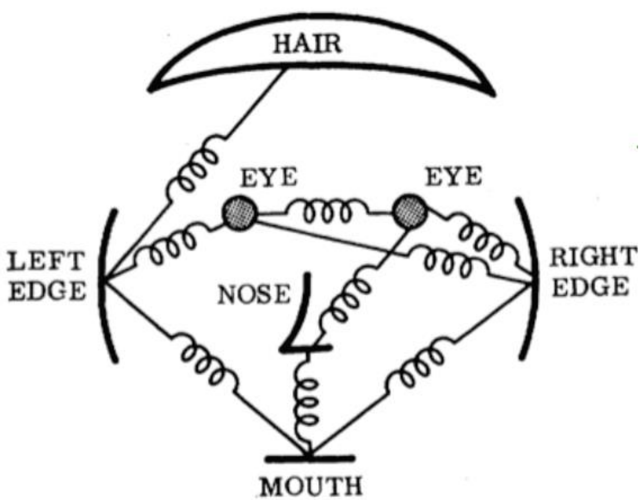
- Star-shaped model

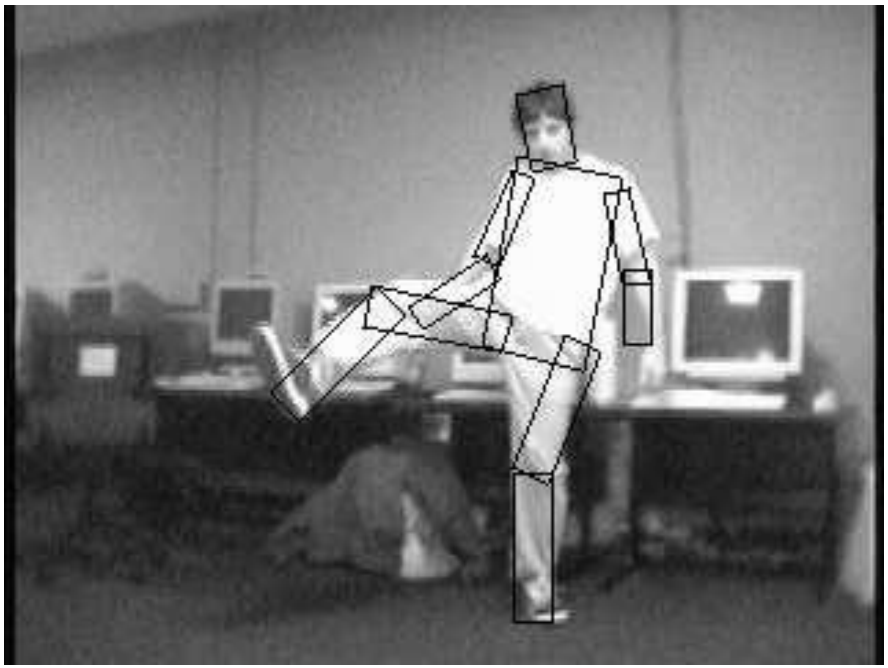


Derek Hoiem

Parts-based Models

- Articulated parts model
  - Object is configuration of parts
  - Each part is detectable and can move around

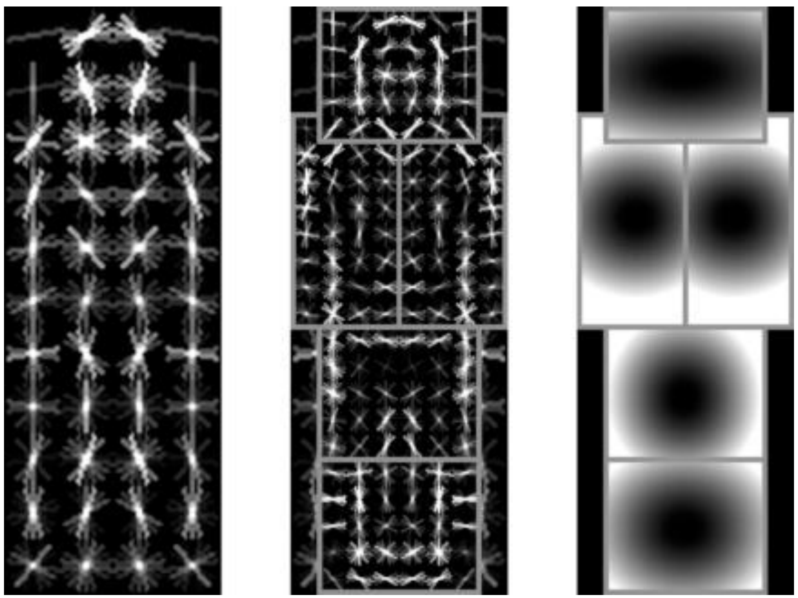


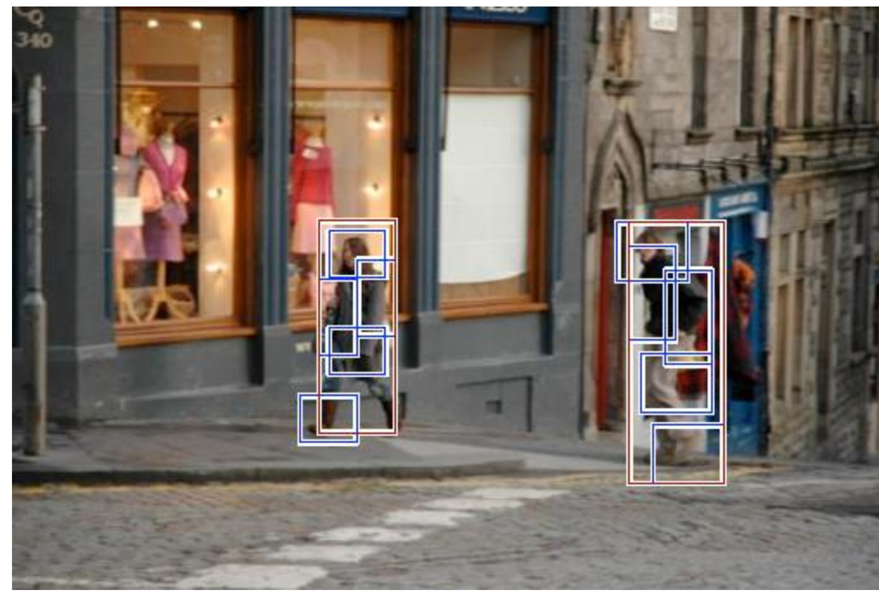


Adapted from Derek Hoiem, images from Felzenszwalb

Discriminative part-based models

Root filter | Part filters | Deformation weights





P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, [Object Detection with Discriminatively Trained Part Based Models](), PAMI 32(9), 2010

Lana Lazebnik

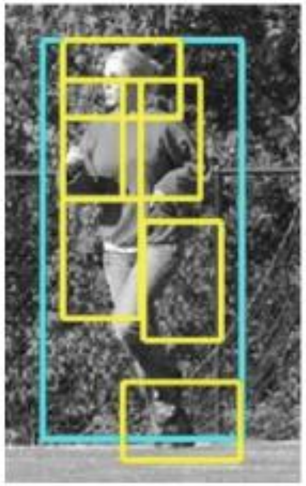
Scoring an object hypothesis

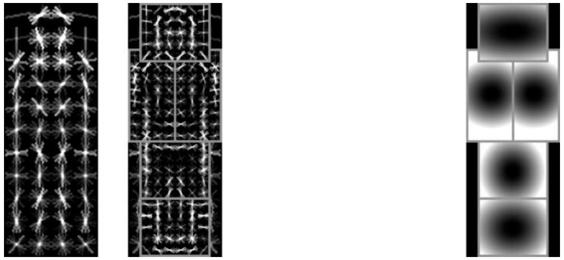
- The score of a hypothesis is the sum of appearance scores minus the sum of deformation costs

$$z = (p_0, \ldots, p_n)$$

p_0 : location of root

$p_1, \ldots, p_n$: location of parts





$$(dx_i, dy_i) = \overbrace{(x_i, y_i)}^{\text{part loc (where we see part)}} - \overbrace{(2(x_0, y_0) + v_i)}^{\text{anchor loc (where we } \textit{expect} \text{ to see part)}}$$

$$\text{score}(p_0, \ldots, p_n) =$$

$$\sum_{i=0}^{n} \boxed{F_i'} \cdot \boxed{\phi(H, p_i)} - \sum_{i=1}^{n} \boxed{d_i} \cdot \boxed{\phi_d(dx_i, dy_i)} + b$$

Appearance weights

Part features

Displacements

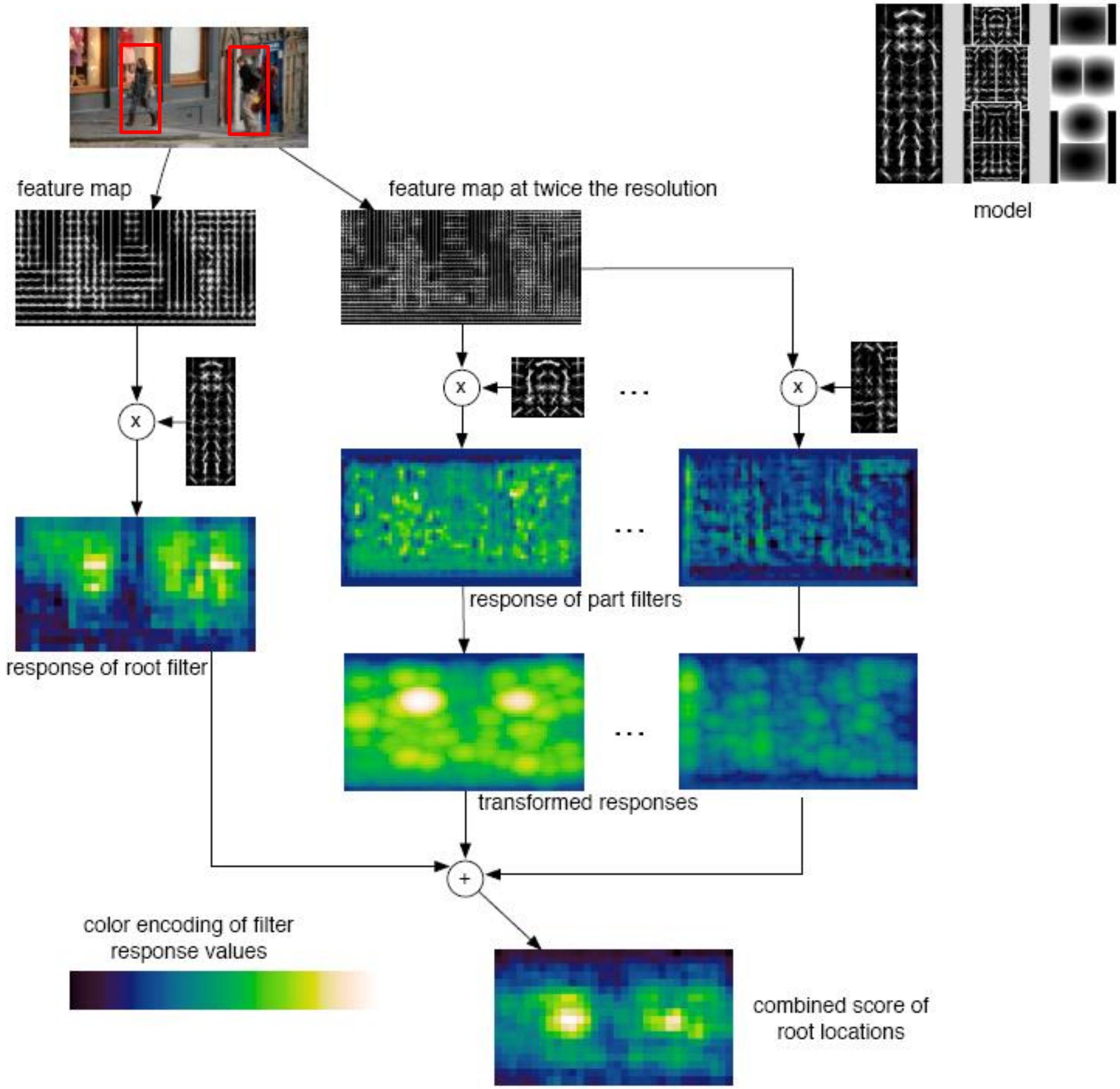
i.e. how much the part p_i moved from its expected anchor location in the x, y directions

Deformation weights

i.e. how much we'll penalize the part p_i for moving from its expected location

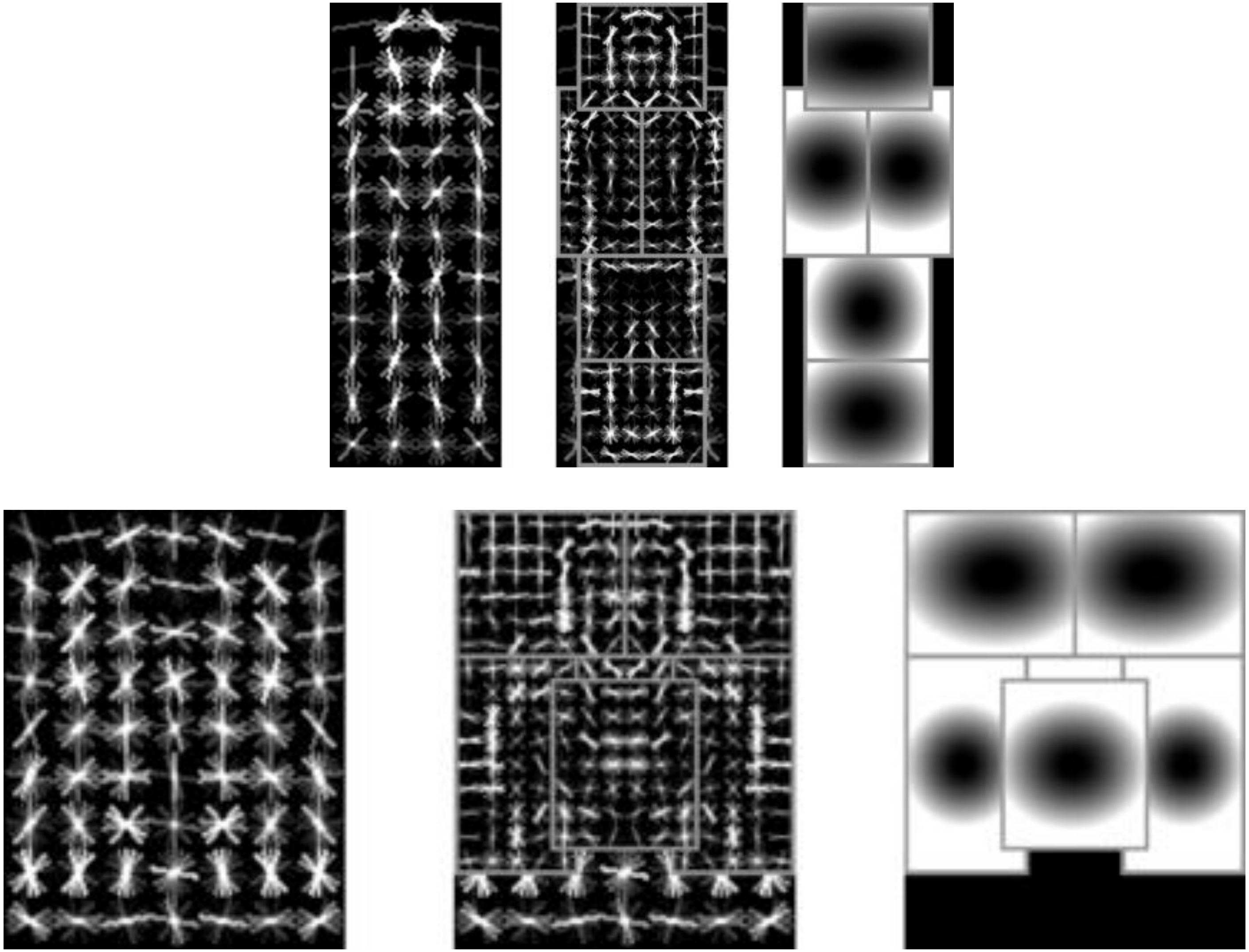
Felzenszwalb et al.

Detection



Felzenszwalb et al.

Person model



Lana Lazebnik

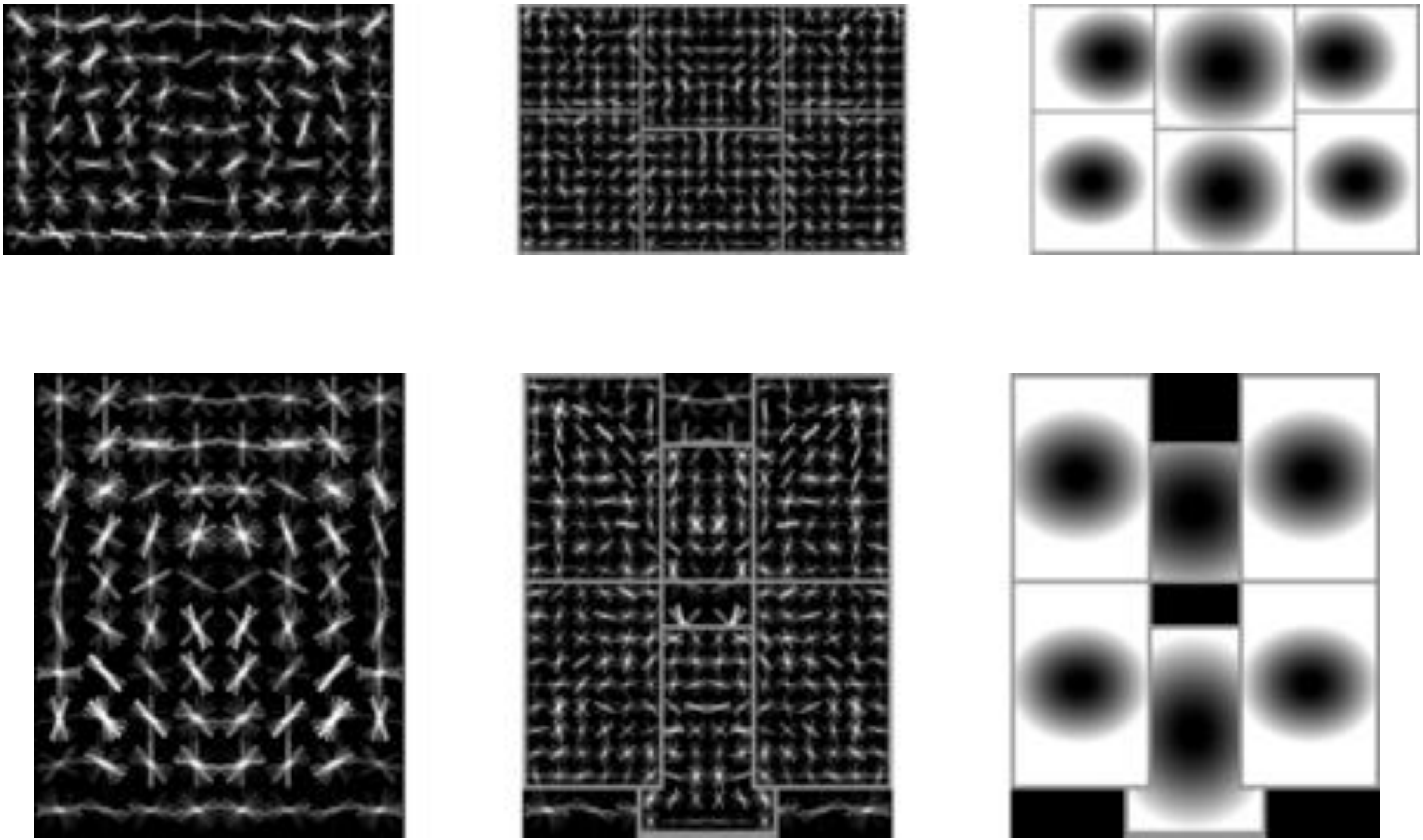
Person detections

high scoring true positives

high scoring false positives
(not enough overlap)

Lana Lazebnik

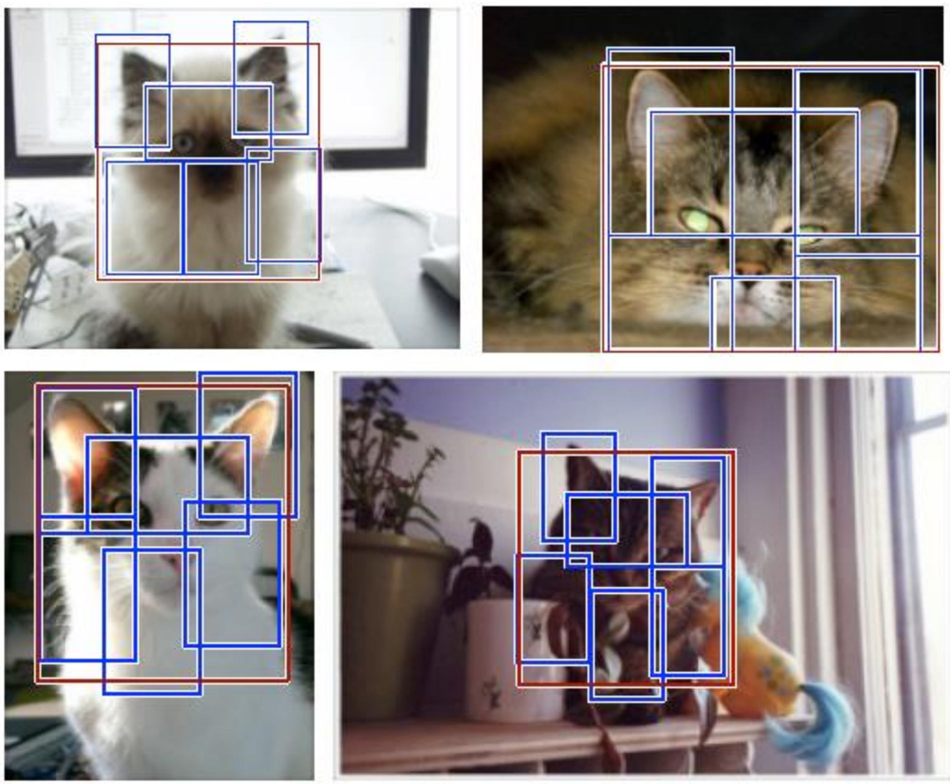
Cat model



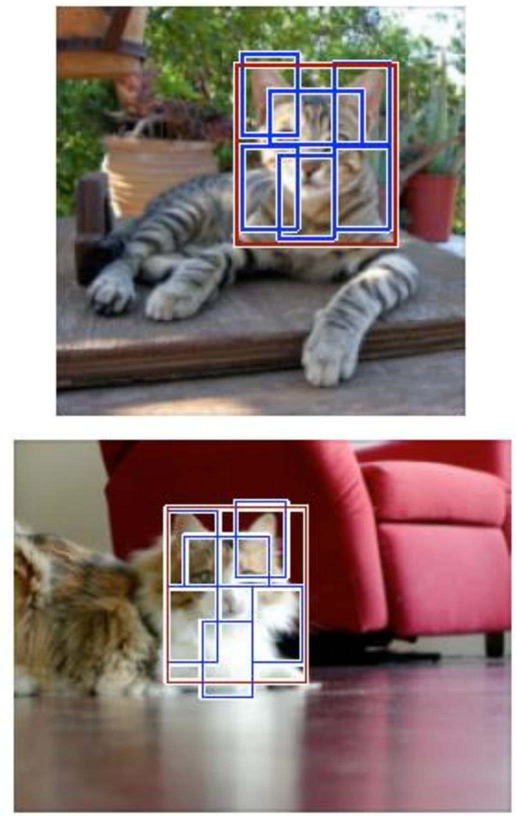
Lana Lazebnik

Cat detections

high scoring true positives



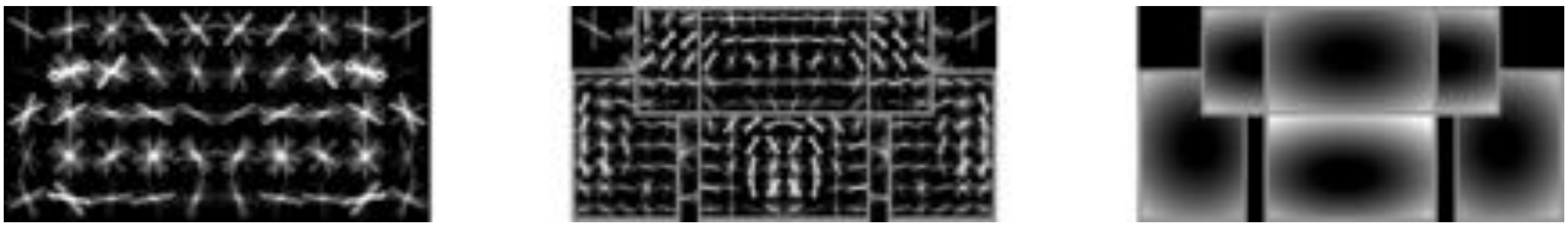
high scoring false positives
(not enough overlap)



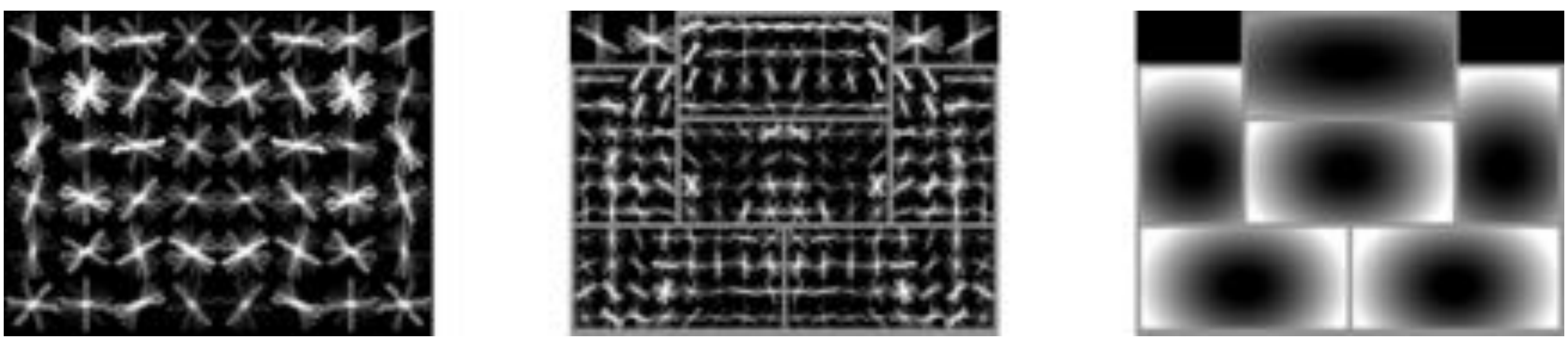
Lana Lazebnik

Car model

Component 1

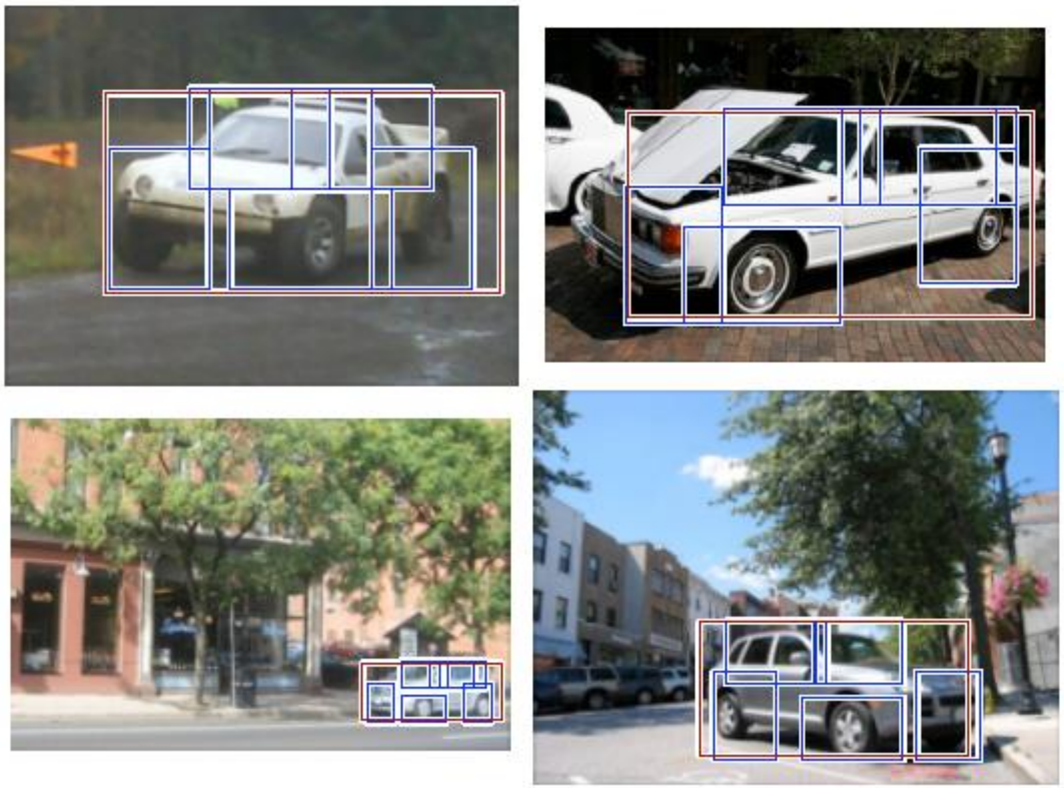


Component 2

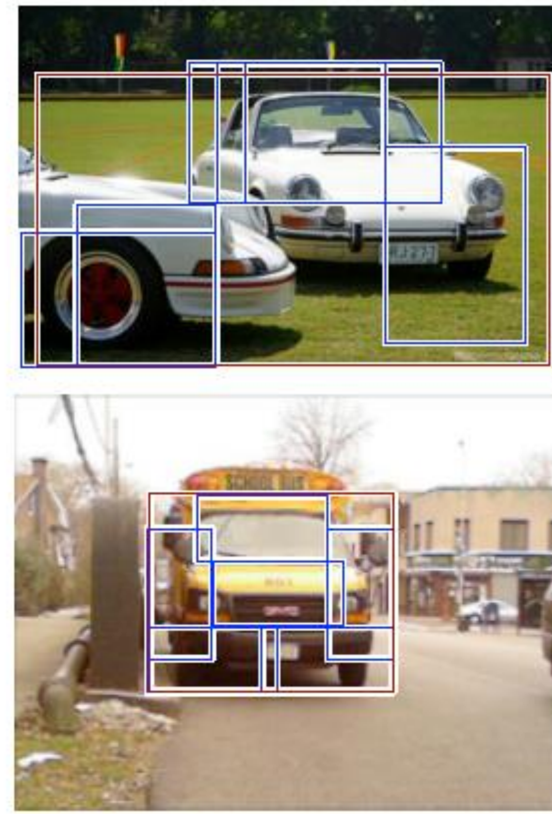


Car detections

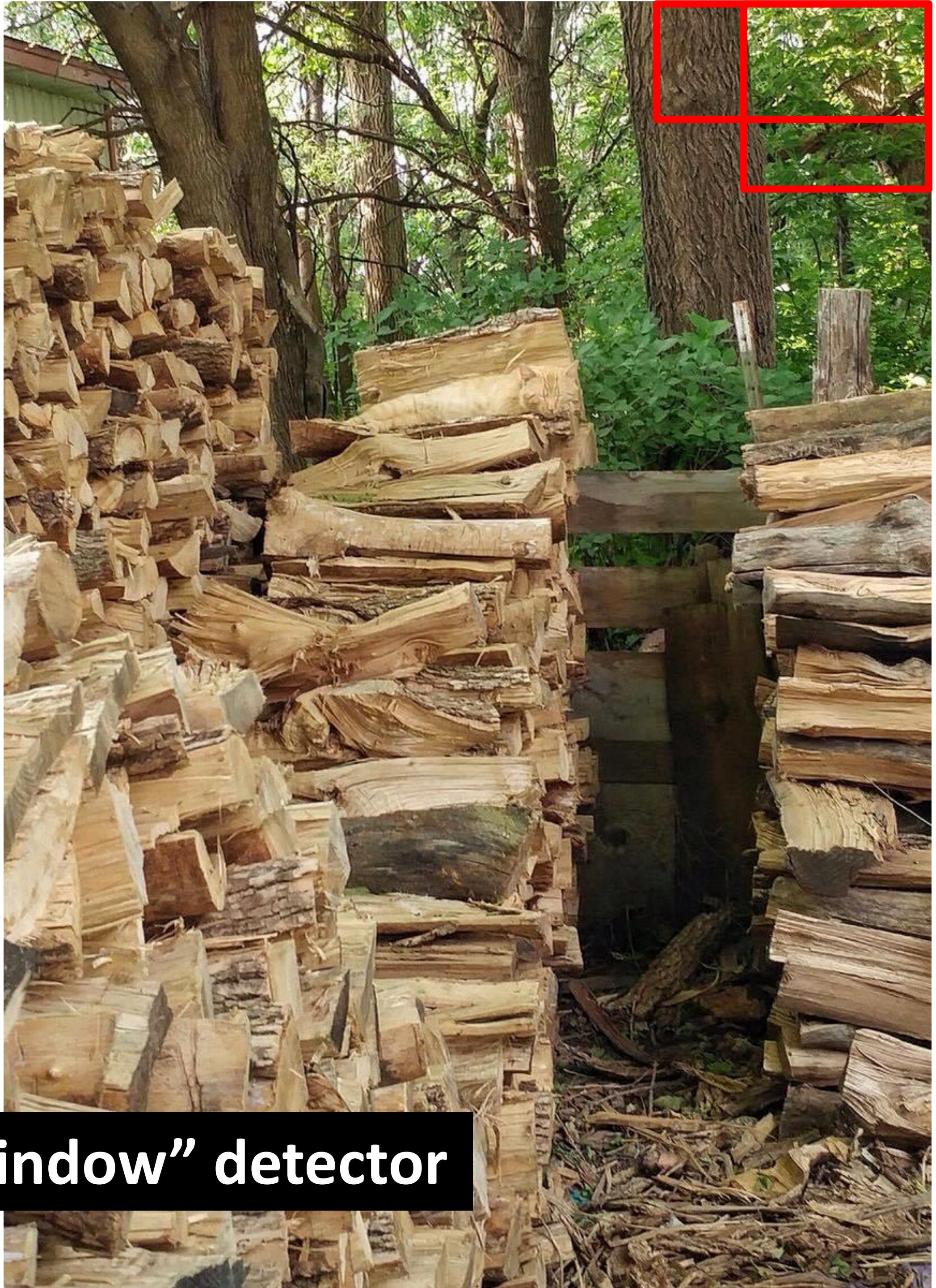
high scoring true positives



high scoring false positives



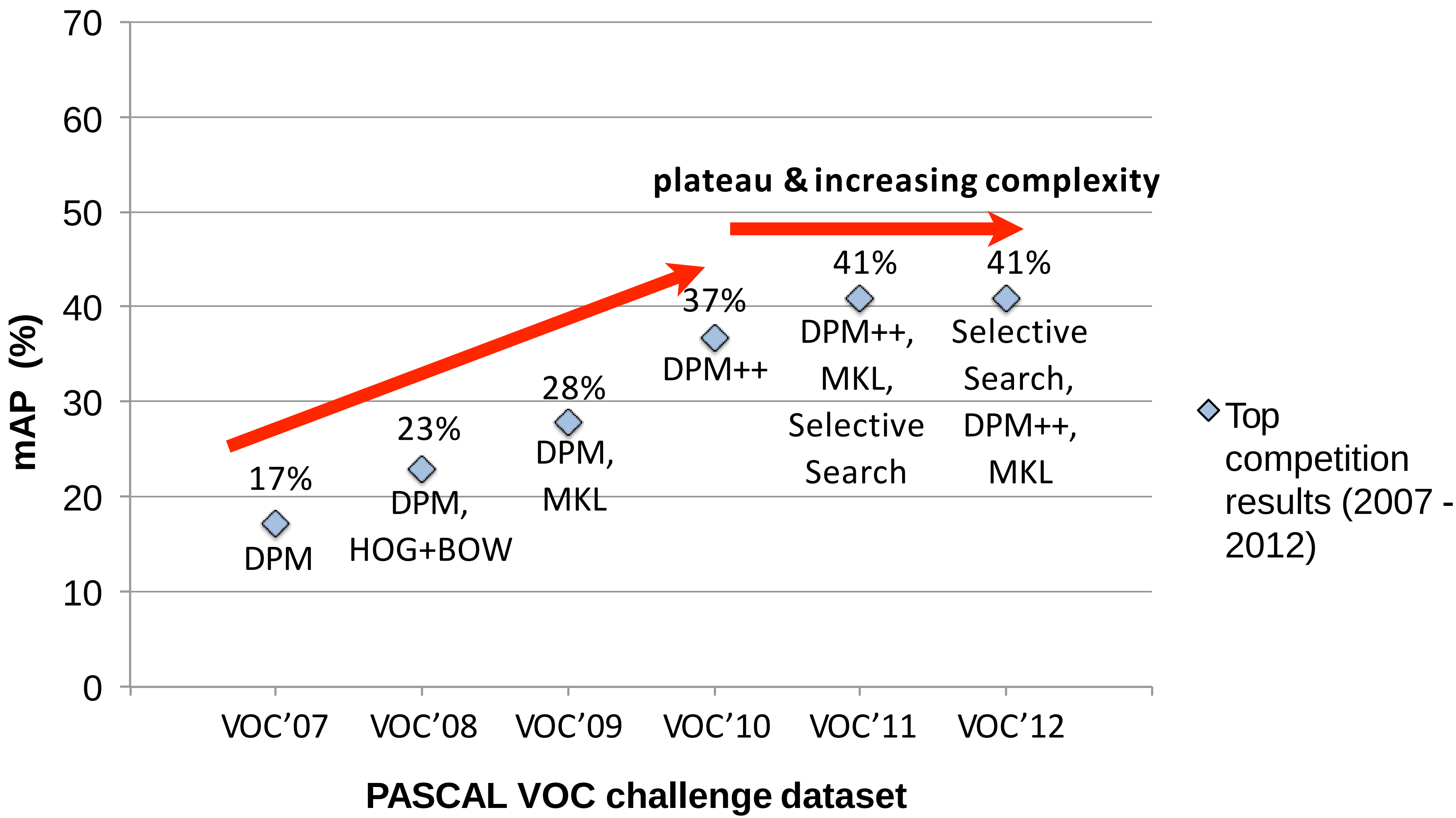
Lana Lazebnik



“Sliding window” detector

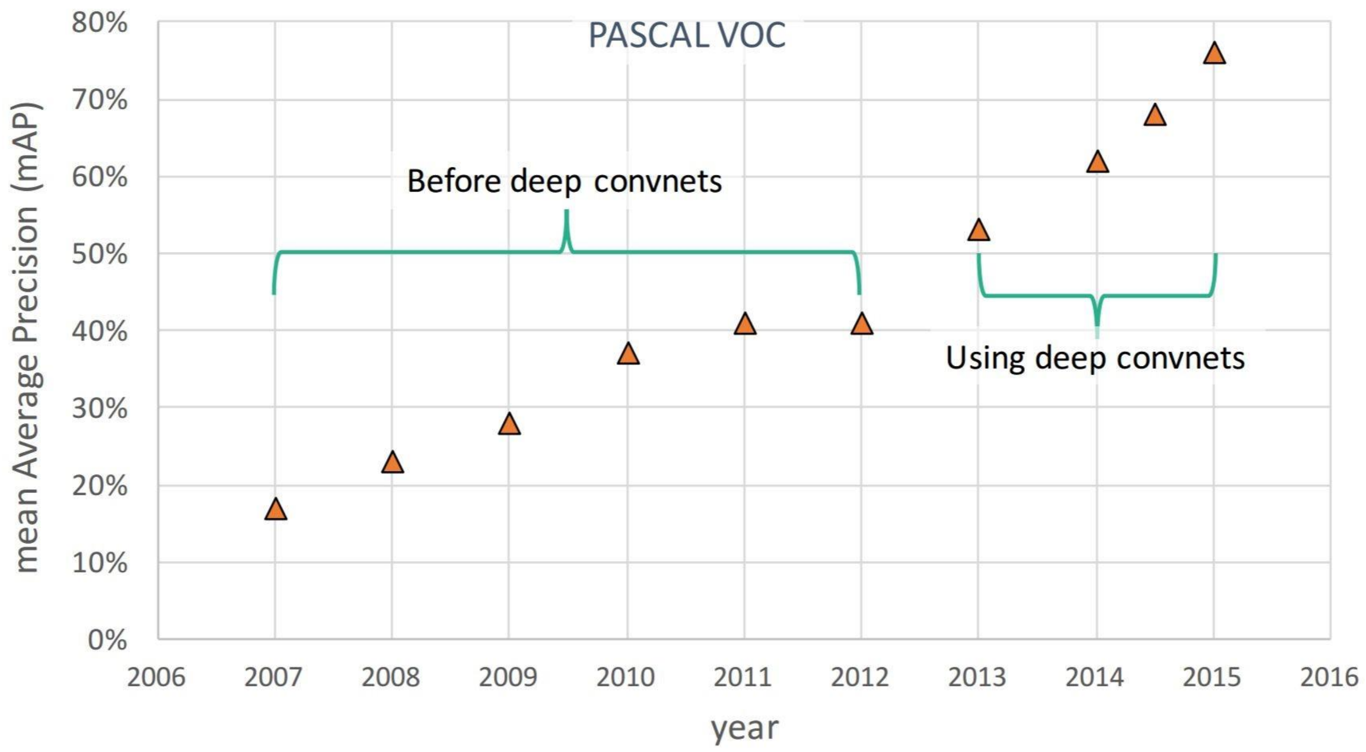
Complexity and the plateau

[Source: http://pascallin.ecs.soton.ac.uk/challenges/VOC/voc20{07,08,09,10,11,12}/results/index.html]



Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

Impact of Deep Learning

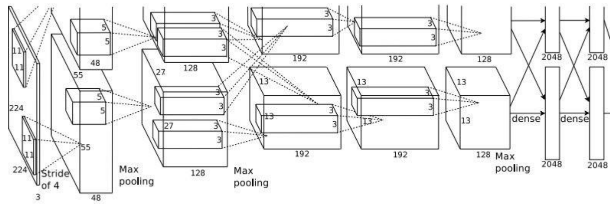


Slide by: Justin Johnson

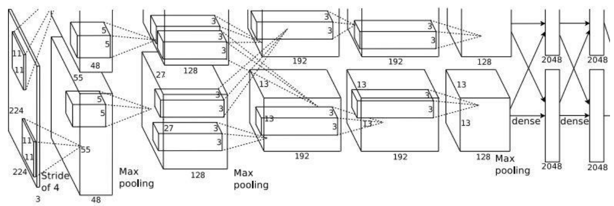
Plan for this lecture

- Fully supervised detection
  - Pre-CNN: Deformable part models
  - Detection with region proposals: R-CNN, Fast/er R-CNN
  - Detection without region proposals: YOLO
  - Semantic and instance segmentation: FCN, Mask R-CNN
- Weak or out-of-domain supervision
  - Weakly supervised object detection
  - Domain adaptation

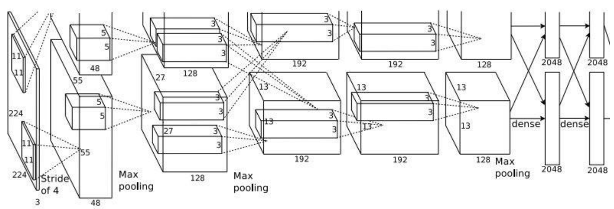
Object Detection as Regression?



CAT: (x, y, w, h)



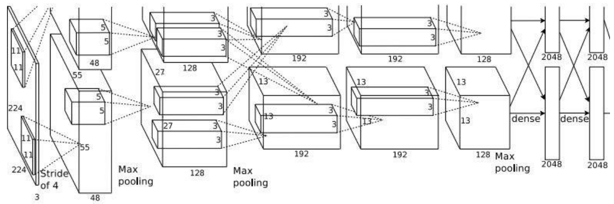
DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)



DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
….

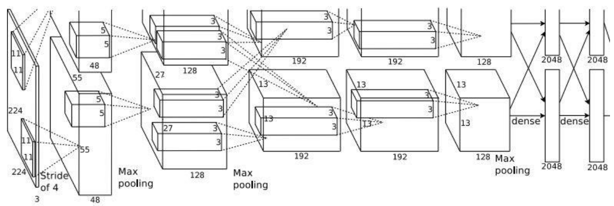
Slide by: Justin Johnson

Object Detection as Regression?



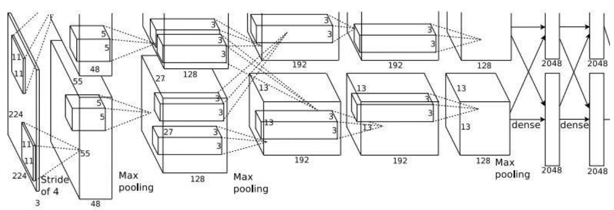
CAT: (x, y, w, h)

4 numbers



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

16 numbers



DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
….

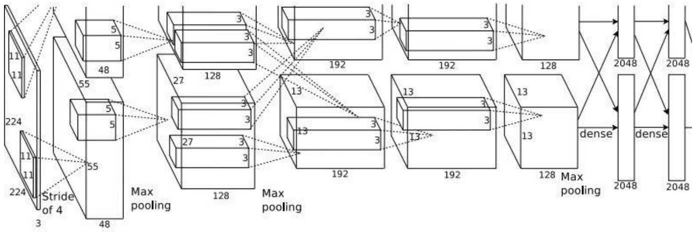
Many numbers!

Each image needs a different number of outputs!

Slide by: Justin Johnson

Object Detection as Classification: Sliding Window

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

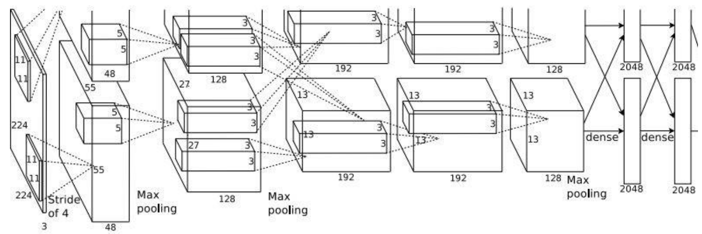


Dog? NO
Cat? NO
Background? YES

Slide by: Justin Johnson

Object Detection as Classification: Sliding Window

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

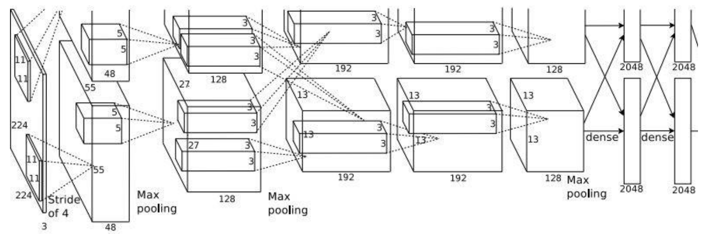


Dog? YES
Cat? NO
Background? NO

Slide by: Justin Johnson

Object Detection as Classification: Sliding Window

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

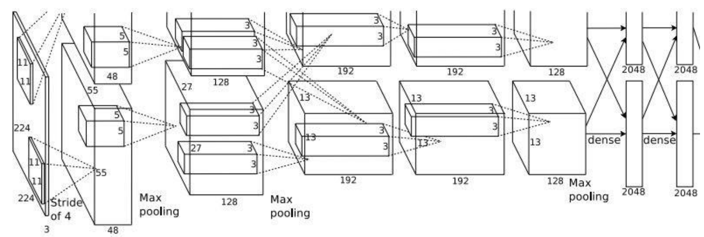


Dog? YES
Cat? NO
Background? NO

Slide by: Justin Johnson

Object Detection as Classification: Sliding Window

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

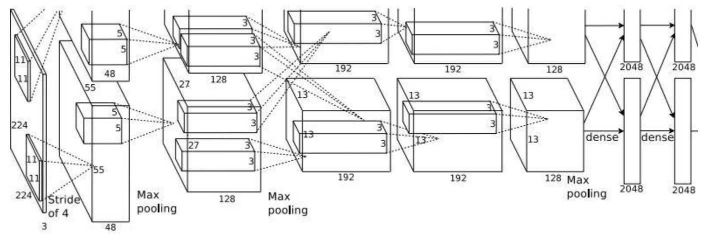


Dog? NO
Cat? YES
Background? NO

Slide by: Justin Johnson

Object Detection as Classification: Sliding Window

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

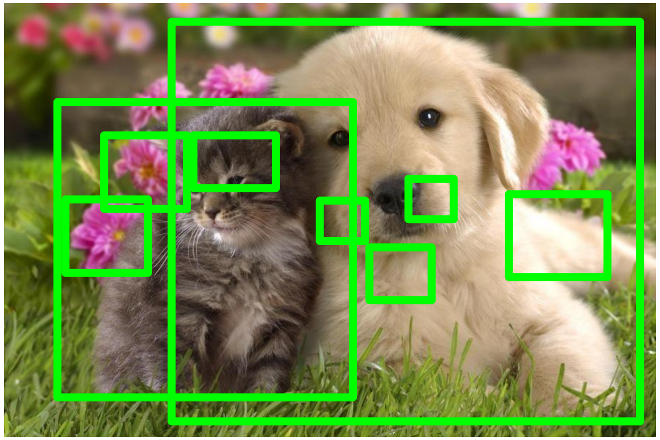


Dog? NO
Cat? YES
Background? NO

Problem: Need to apply CNN to huge number of locations and scales, very computationally expensive!

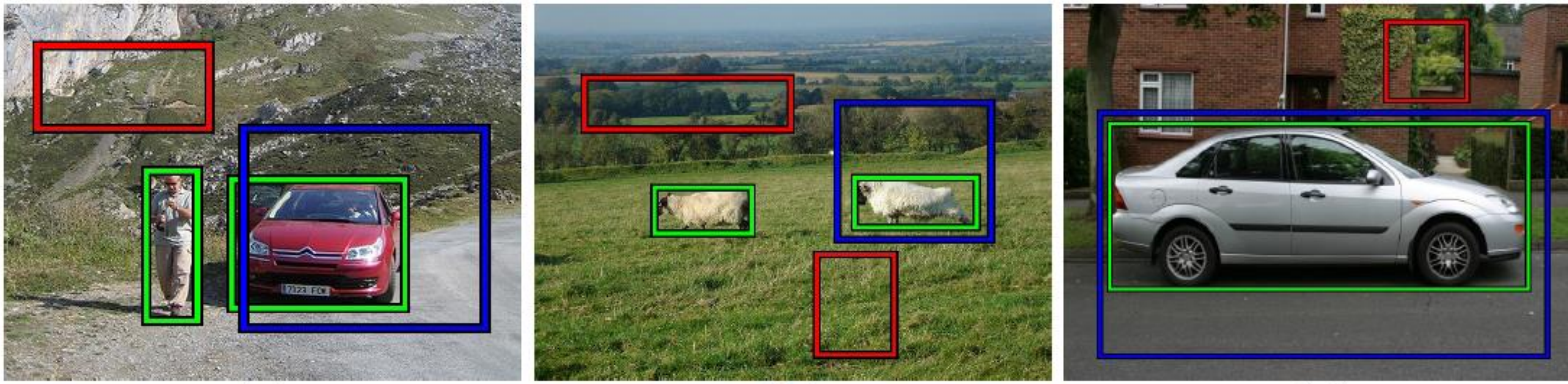
Region Proposals

- Find “blobby” image regions that are likely to contain objects
- Relatively fast to run; e.g. Selective Search gives 1000 region proposals in a few seconds on CPU



Alexe et al, “Measuring the objectness of image windows”, TPAMI 2012
Uijlings et al, “Selective Search for Object Recognition”, IJCV 2013
Cheng et al, “BING: Binarized normed gradients for objectness estimation at 300fps”, CVPR 2014
Zitnick and Dollar, “Edge boxes: Locating object proposals from edges”, ECCV 2014

Speeding up detection: Restrict set of windows we pass through SVM to those w/ high "objectness"



(a) (b) (c)

Fig. 1: **Desired behavior of an objectness measure.** *The desired objectness measure should score the blue windows, partially covering the objects, lower than the ground truth windows (green), and score even lower the red windows containing only stuff or small parts of objects.*

Alexe et al., "Measuring the objectness of image windows", PAMI 2012 and CVPR 2010

Proposals cue: color contrast at boundary

(a) (b) (c)

Fig. 3: **CC success and failure. Success:** *the windows containing the objects (cyan) have high color contrast with their surrounding ring (yellow) in images (a) and (b).* **Failure:** *the color contrast for windows in cyan in image (c) is much lower.*

Alexe et al., "Measuring the objectness of image windows", PAMI 2012 and CVPR 2010

Proposals cue: no segments “straddling” the object box

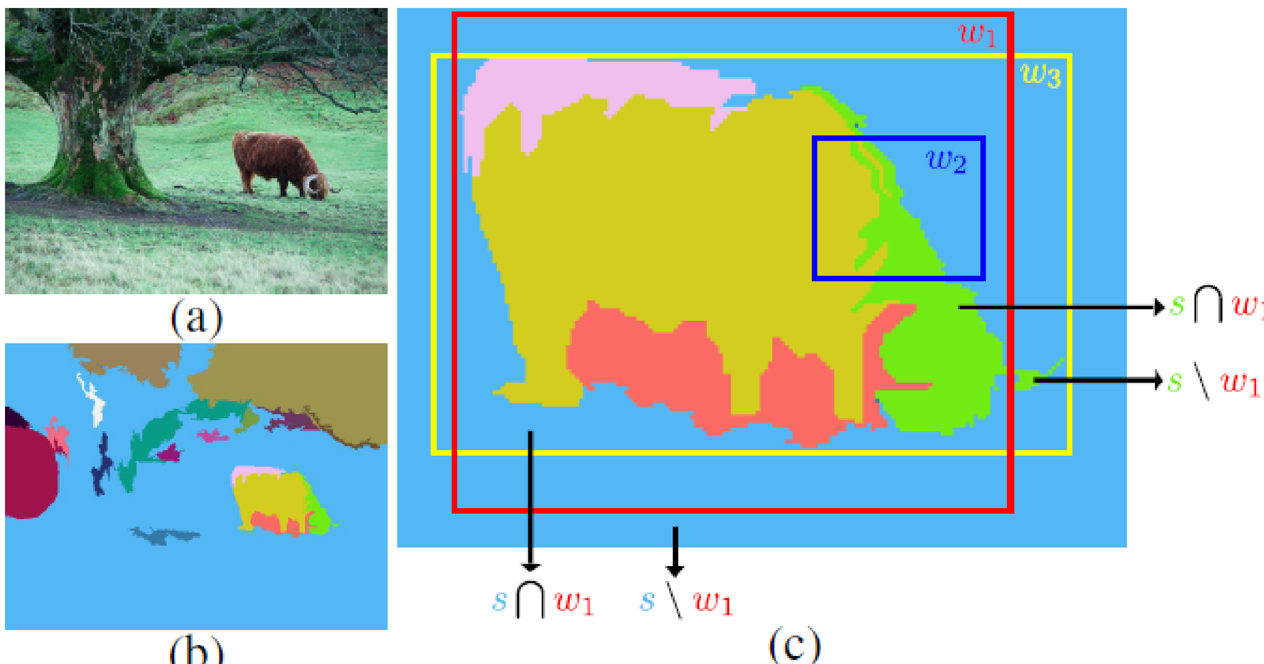
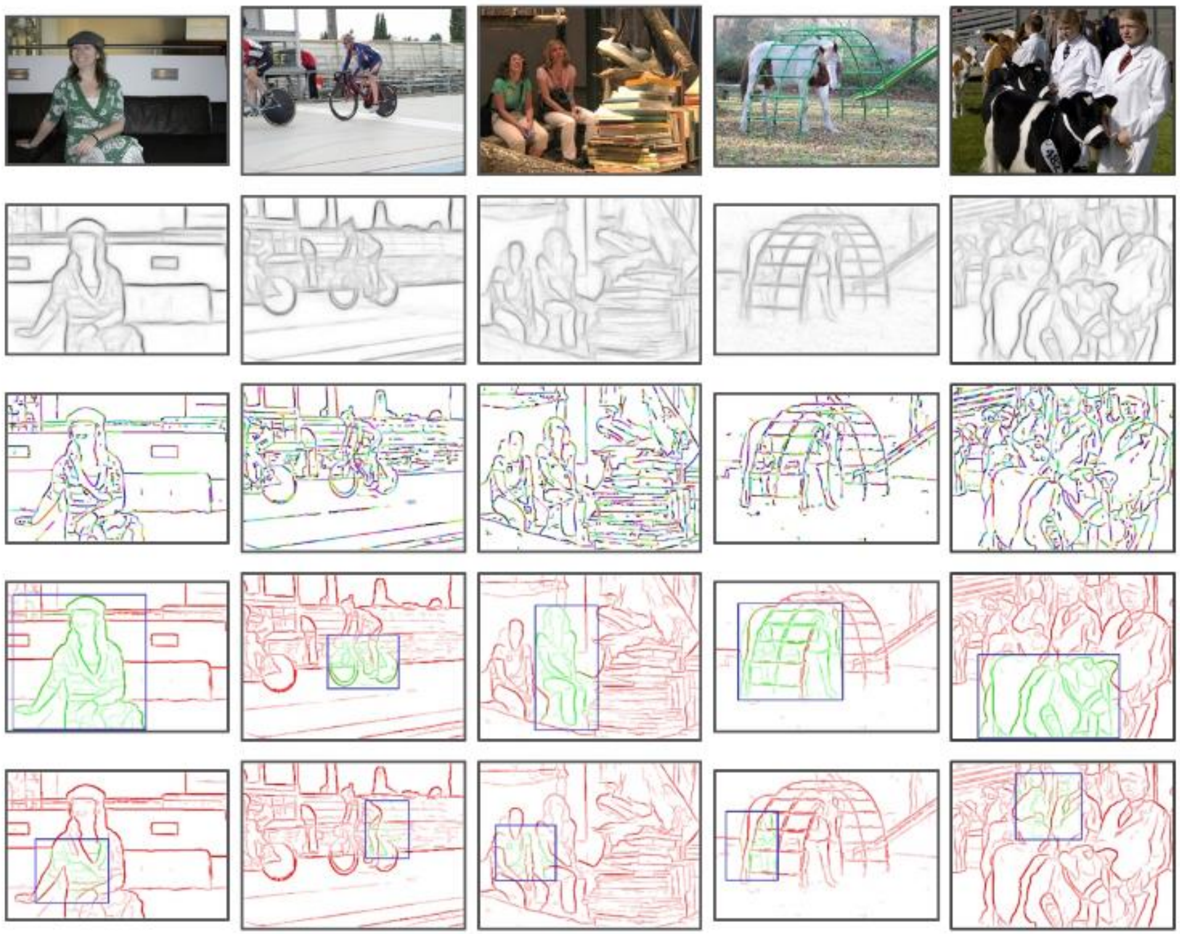


Fig. 5: **The SS cue.** *Given the segmentation (b) of image (a), for a window w we compute* $\mathrm{SS}(\mathrm{w}, \theta_{\mathrm{SS}})$ *(eq. 4). In (c), most of the surface of w_1 is covered by superpixels contained almost entirely inside it. Instead, all superpixels passing by w_2 continue largely outside it. Therefore, w_1 has a higher* SS *score than w_2. The window w_3 has an even higher score as it fits the object tightly.*

Alexe et al., “Measuring the objectness of image windows”, PAMI 2012 and CVPR 2010

Proposals cue:
many edges wholly contained inside box



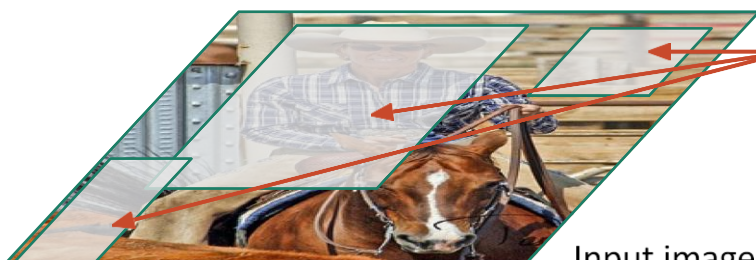
Zitnick and Dollar, “Edge Boxes: Locating Object Proposals from Edges”, ECCV 2014

R-CNN

Input image

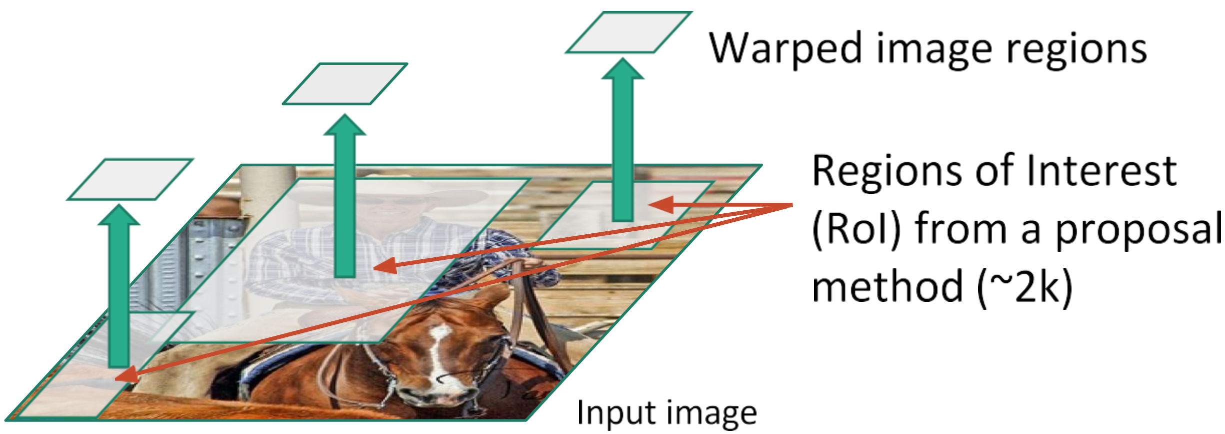
Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

R-CNN



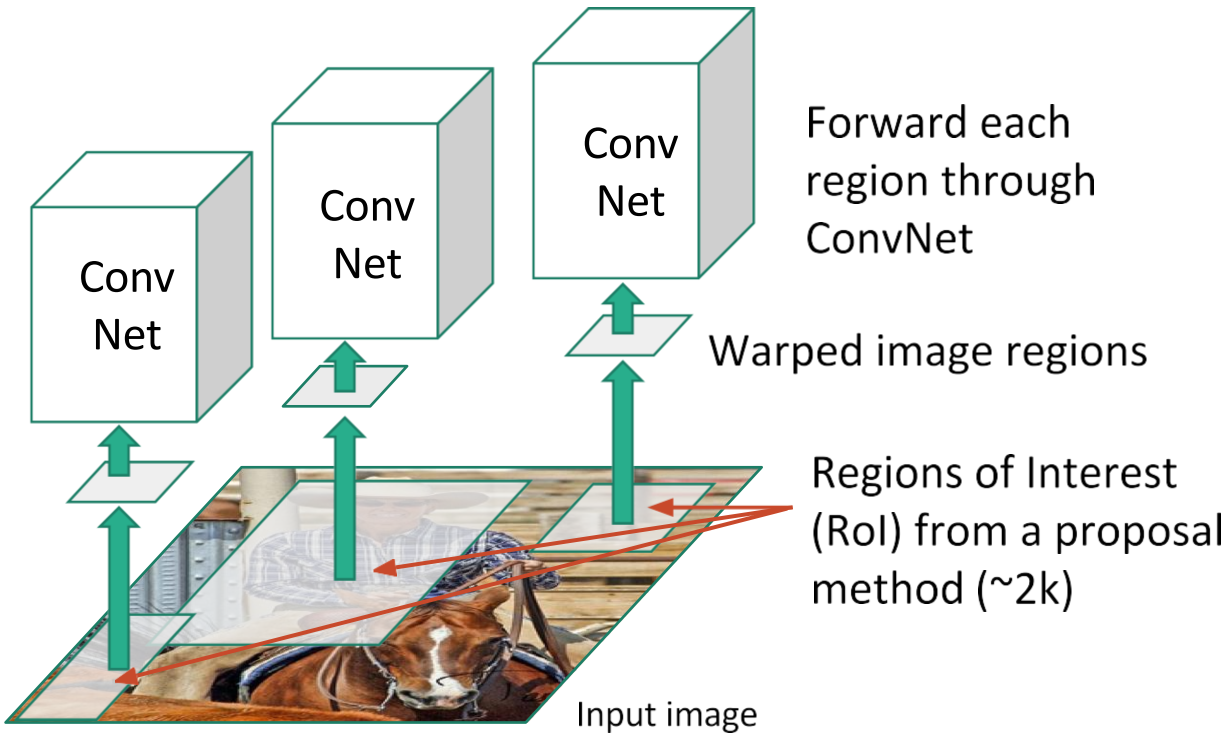
Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

R-CNN



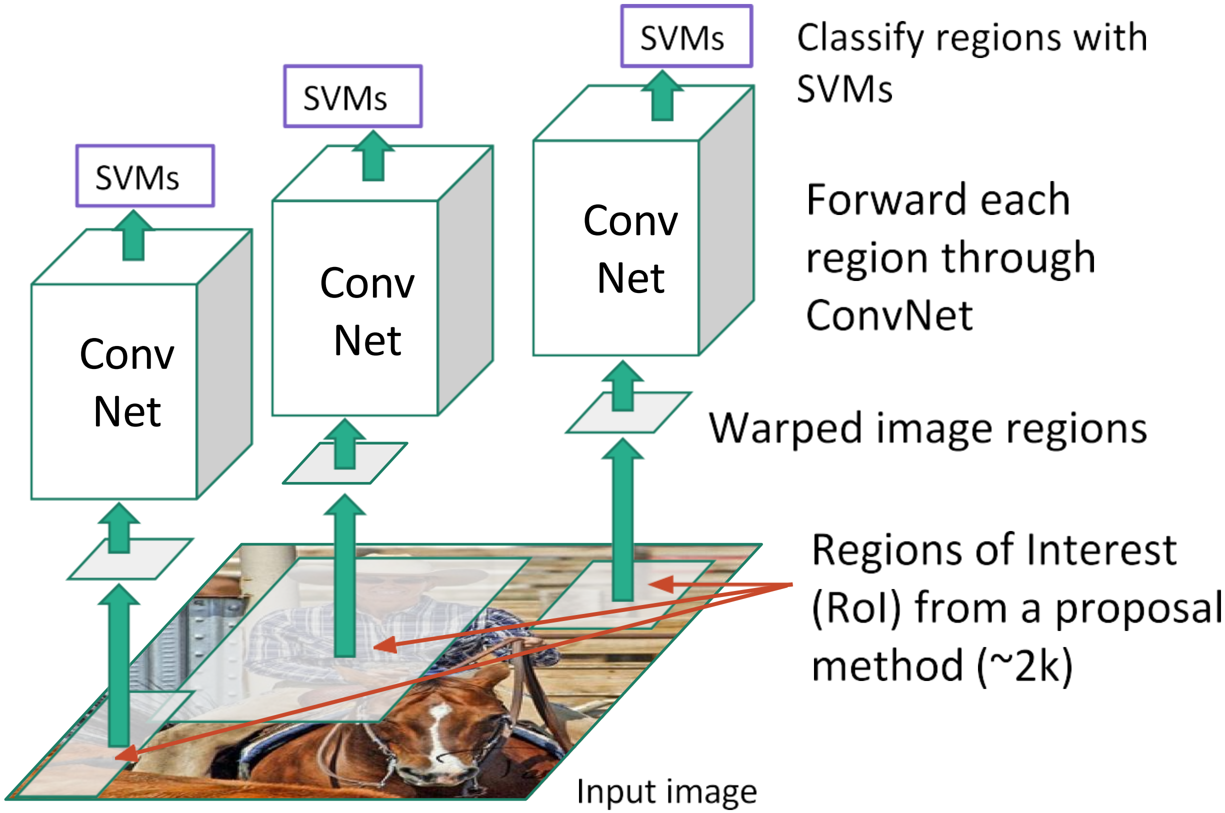
Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

R-CNN



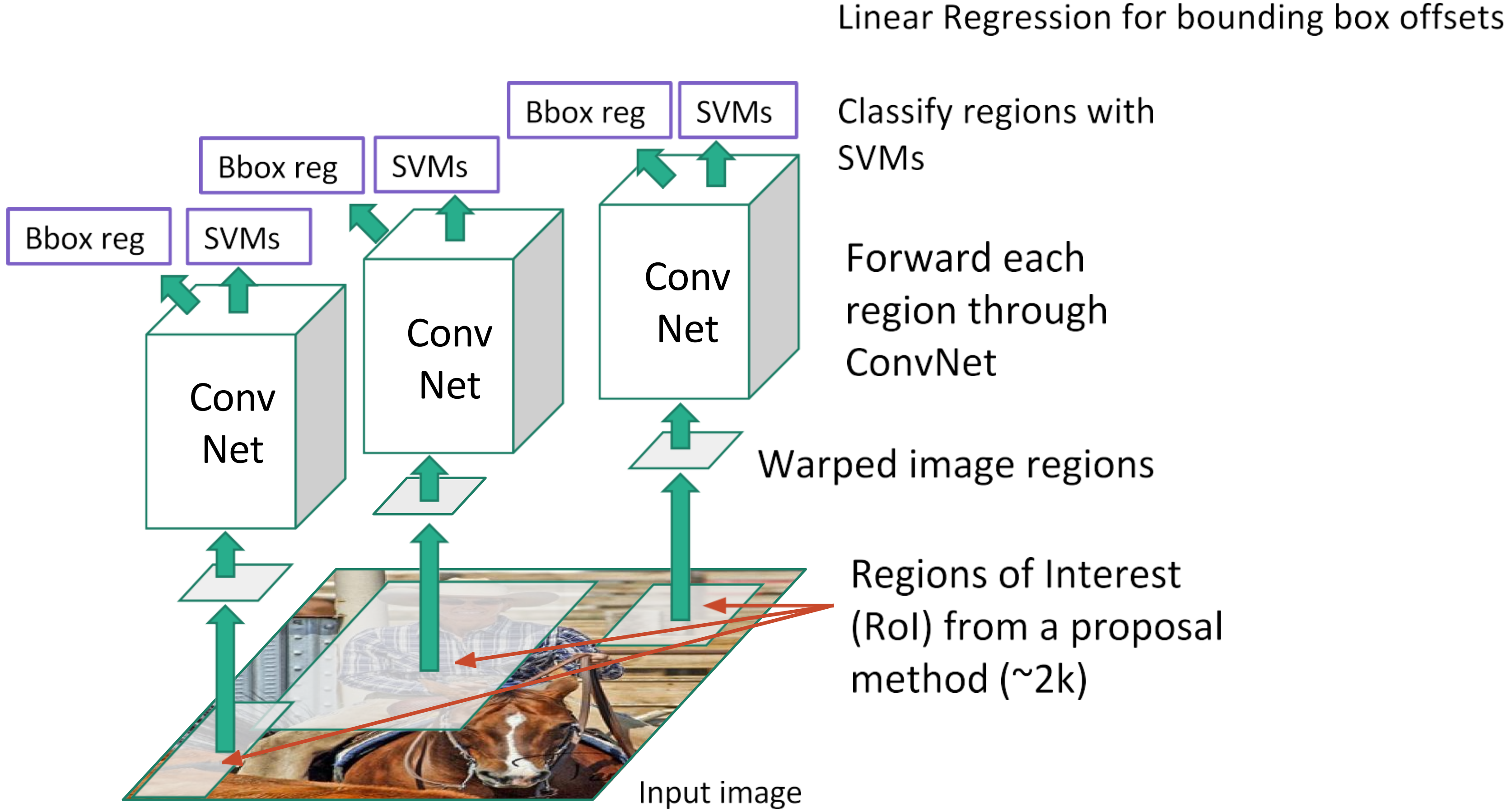
Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

R-CNN



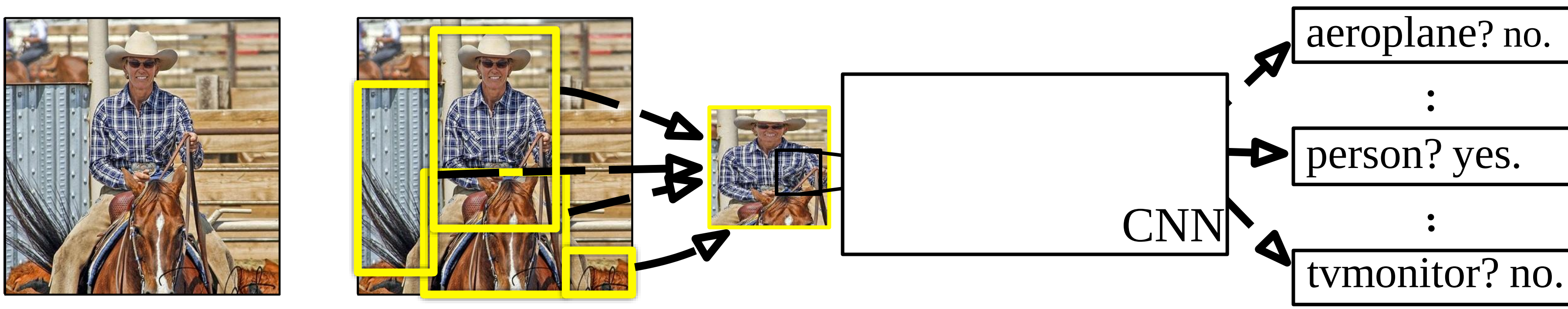
Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

R-CNN



Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

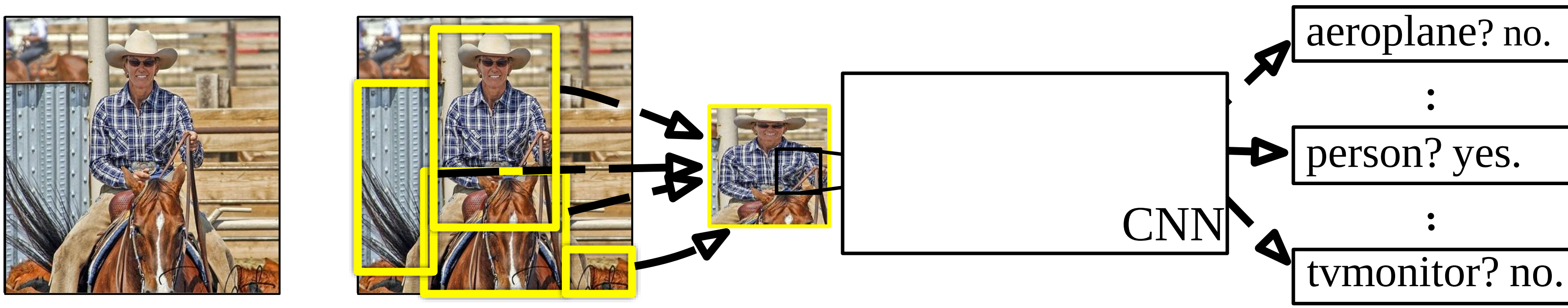
R-CNN: Regions with CNN features



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

Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

R-CNN at test time: Step 1



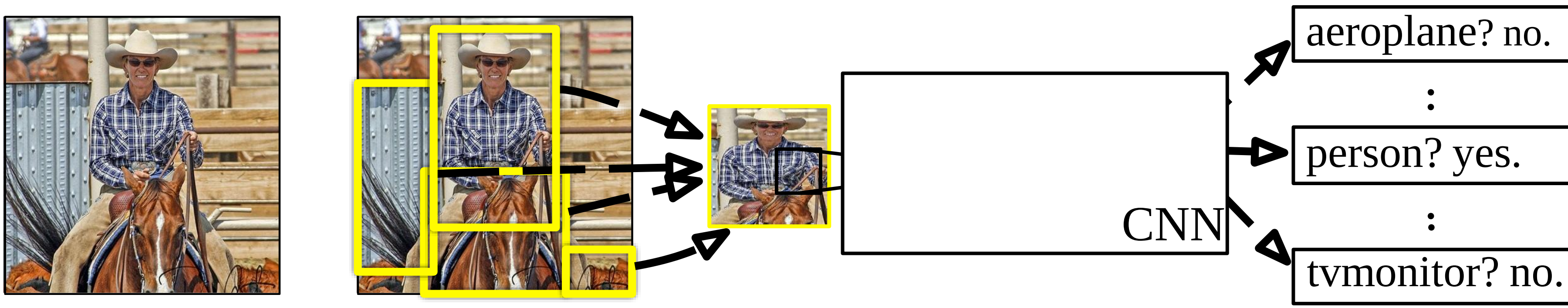
Input image → Extract region proposals (~2k / image)

Proposal-method agnostic, many choices

- Selective Search [van de Sande, Uijlings et al.] (Used in this work)
- Objectness [Alexe etal.]
- Category independent object proposals [Endres & Hoiem]
- CPMC [Carreira & Sminchisescu]

Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

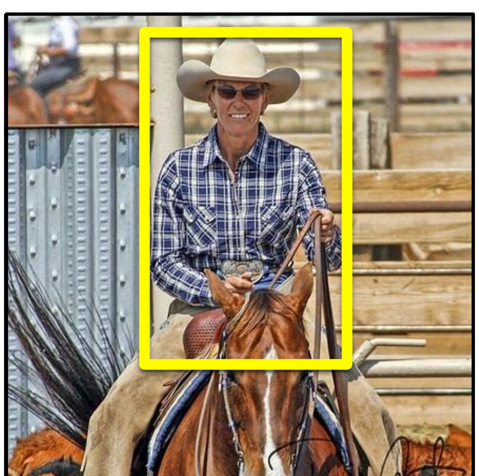
R-CNN at test time: Step 2



Input image

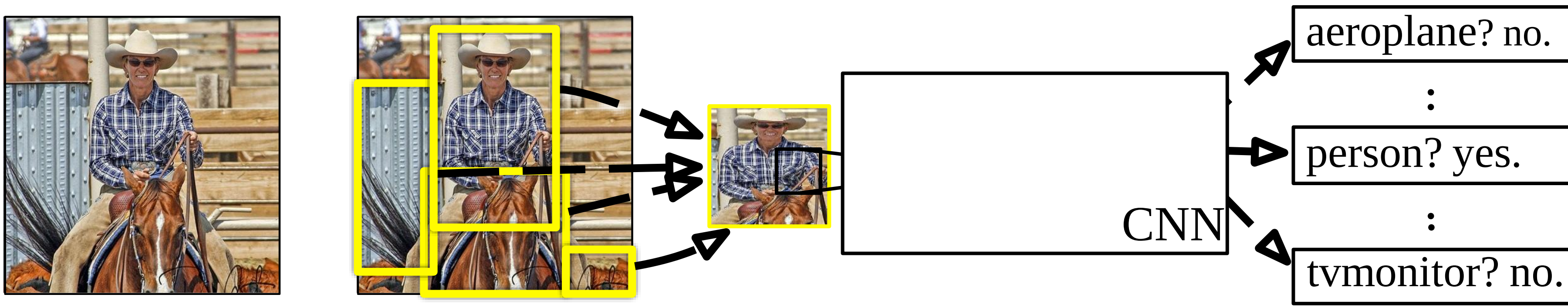
Extract region proposals (~2k / image)

Compute CNN features



Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

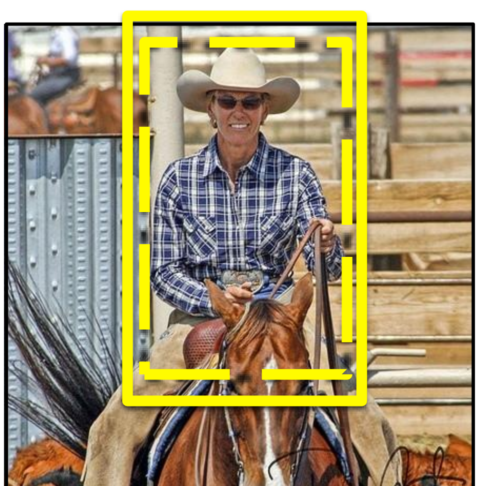
R-CNN at test time: Step 2



Input image

Extract region proposals (~2k / image)

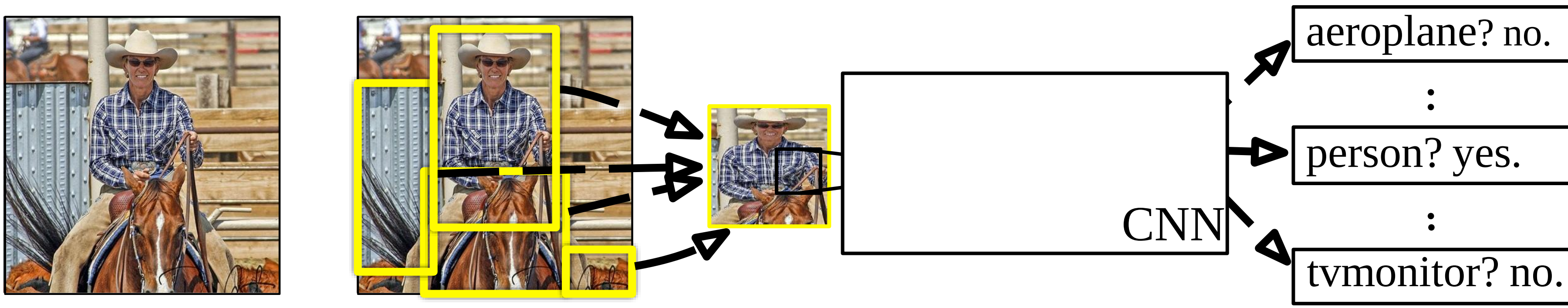
Compute CNN features



Dilate proposal

Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

R-CNN at test time: Step 2



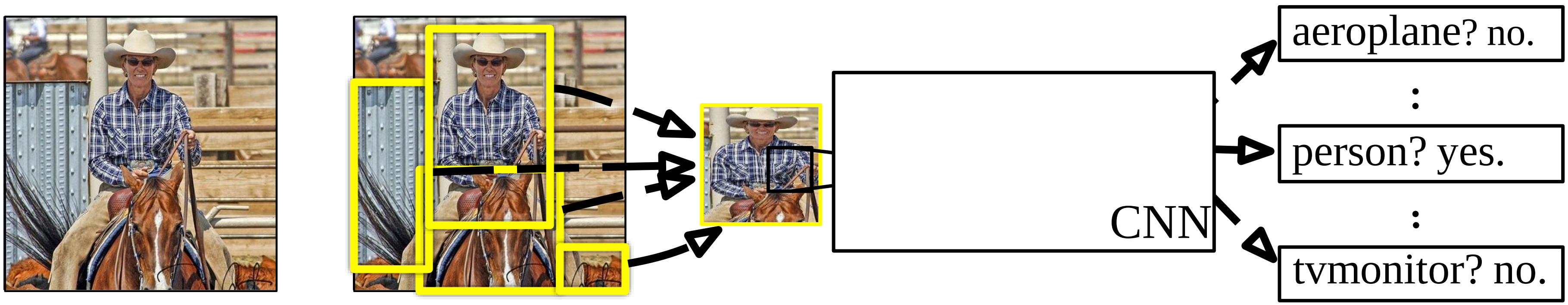
Input image

Extract region proposals (~2k / image) → Compute CNN features

a. Crop

Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

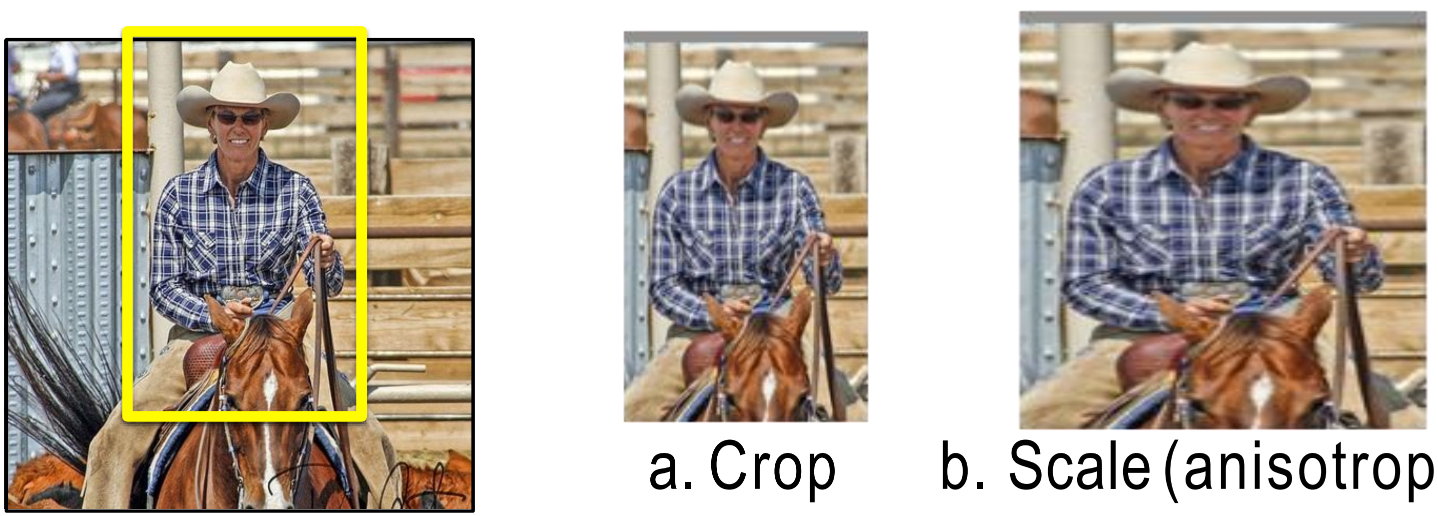
R-CNN at test time: Step 2



Input image

Extract region proposals (~2k / image)

Compute CNN features



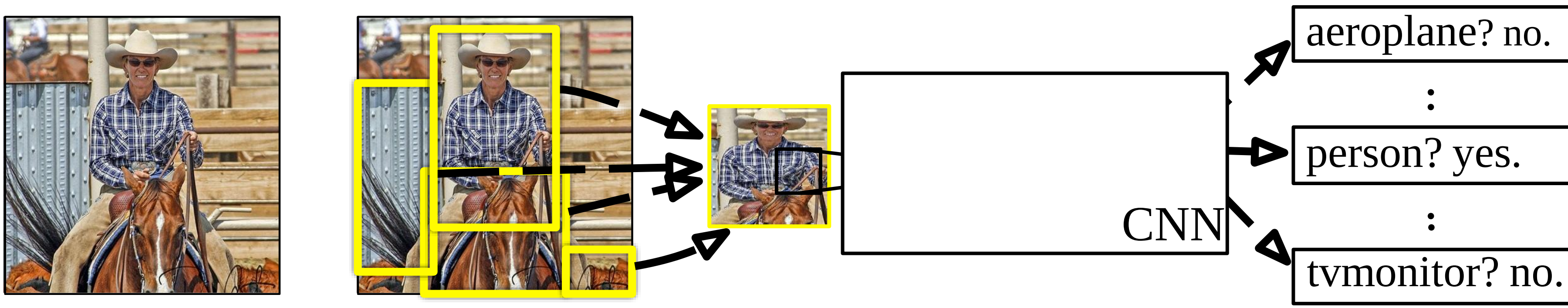
227 x 227

a. Crop

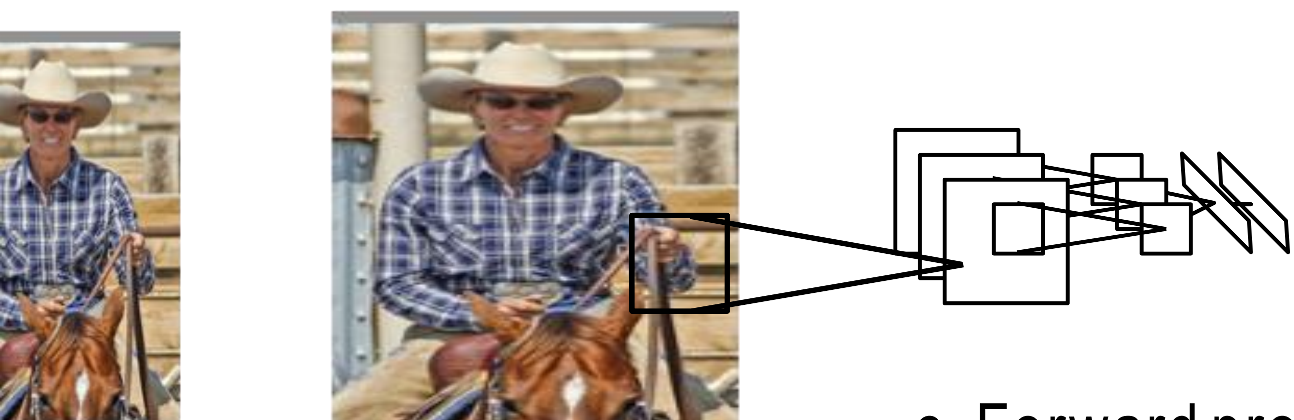
b. Scale (anisotropic)

Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

R-CNN at test time: Step 2



Input image | Extract region proposals (~2k / image) | Compute CNN features

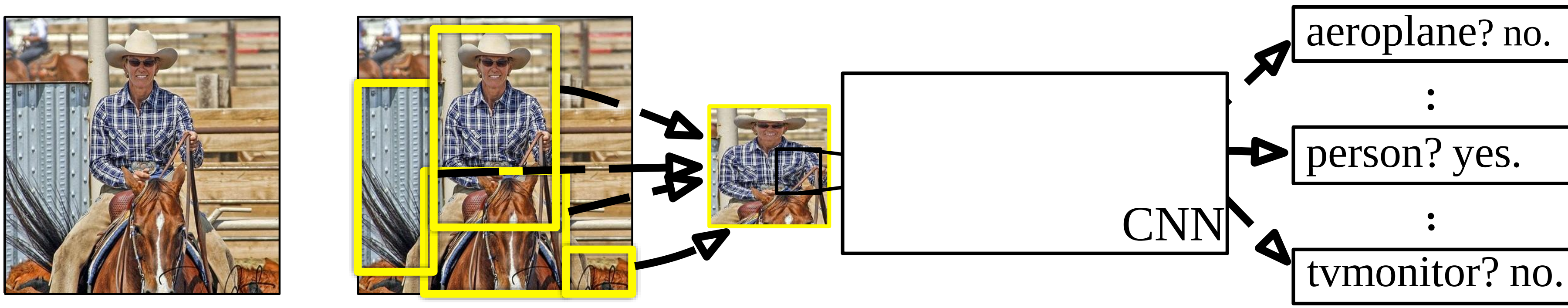


a. Crop | b. Scale (anisotropic) | c. Forward propagate

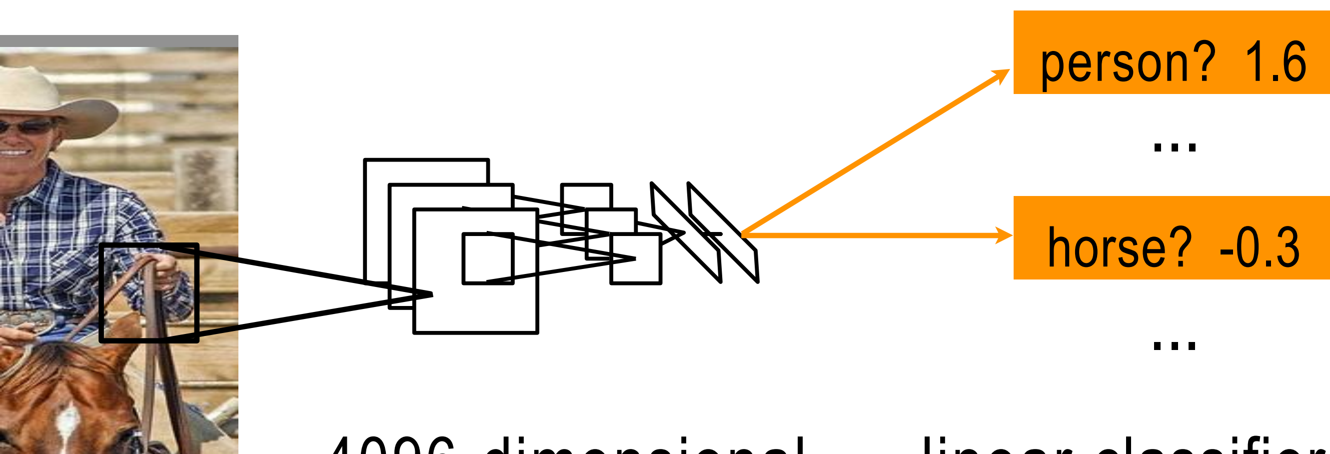
Output: "fc_7" features

Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

R-CNN at test time: Step 3



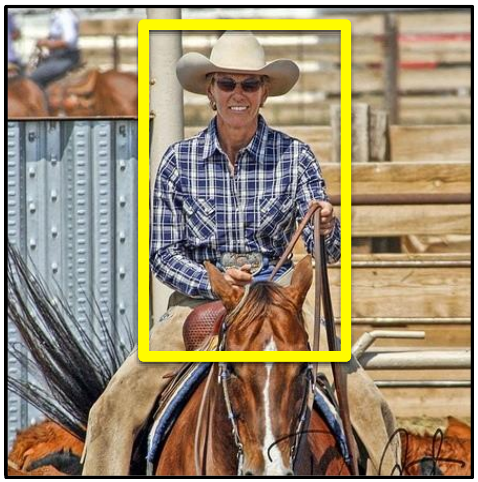
Input image | Extract region proposals (~2k / image) | Compute CNN features | Classify regions



proposal | 4096-dimensional fc_7 feature vector | linear classifiers (SVM or softmax)

Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

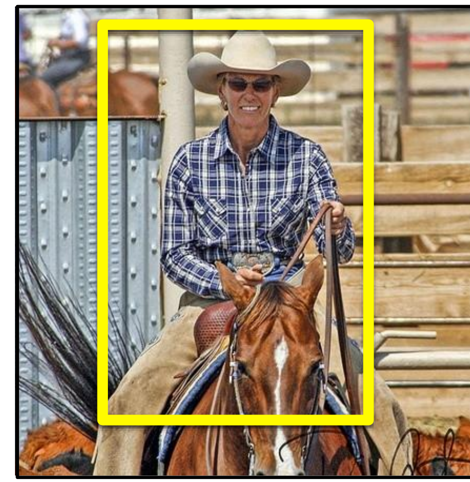
Step 4: Object proposal refinement



Original
proposal

Linear regression

on CNN features

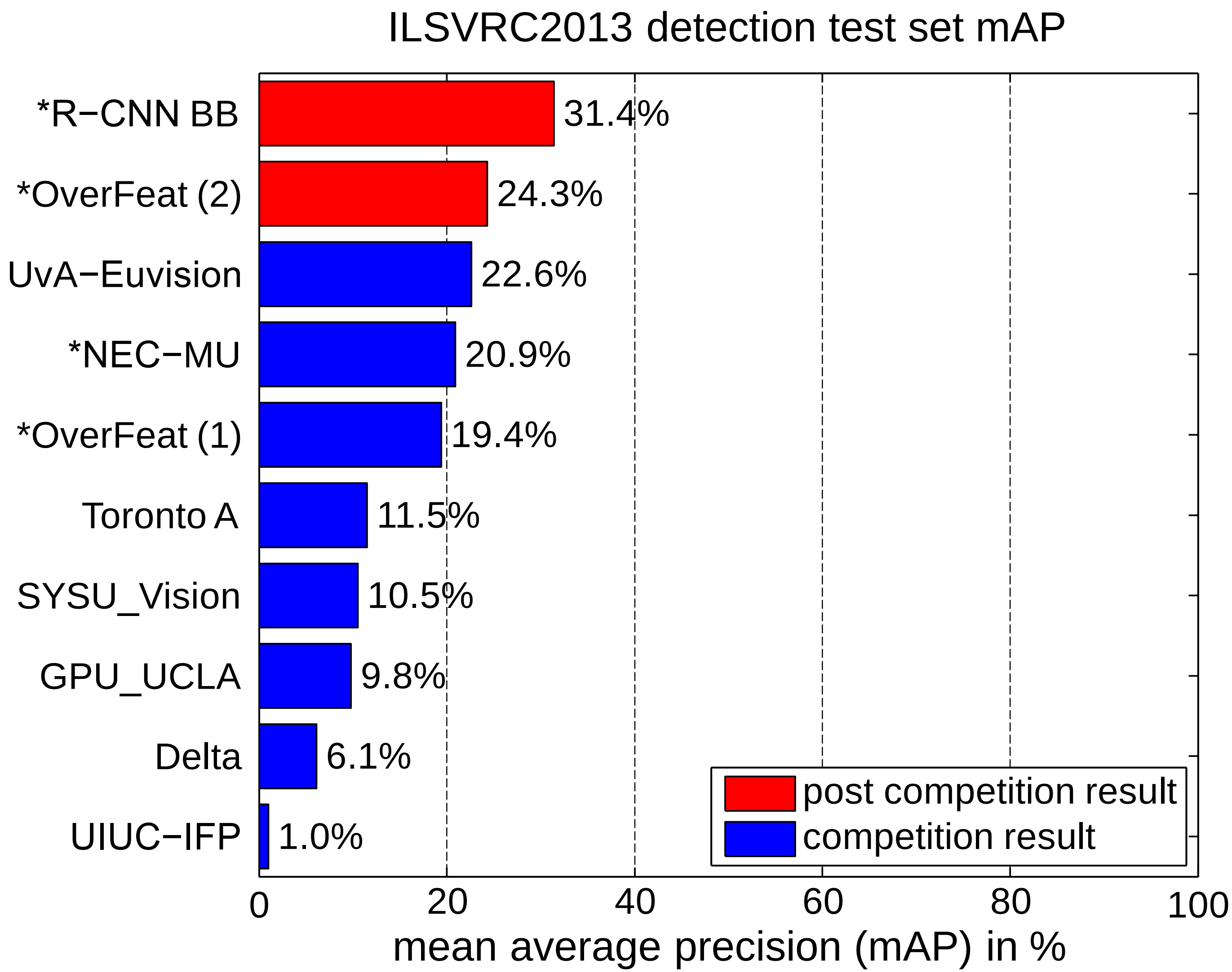


Predicted
object bounding box

Bounding-box regression

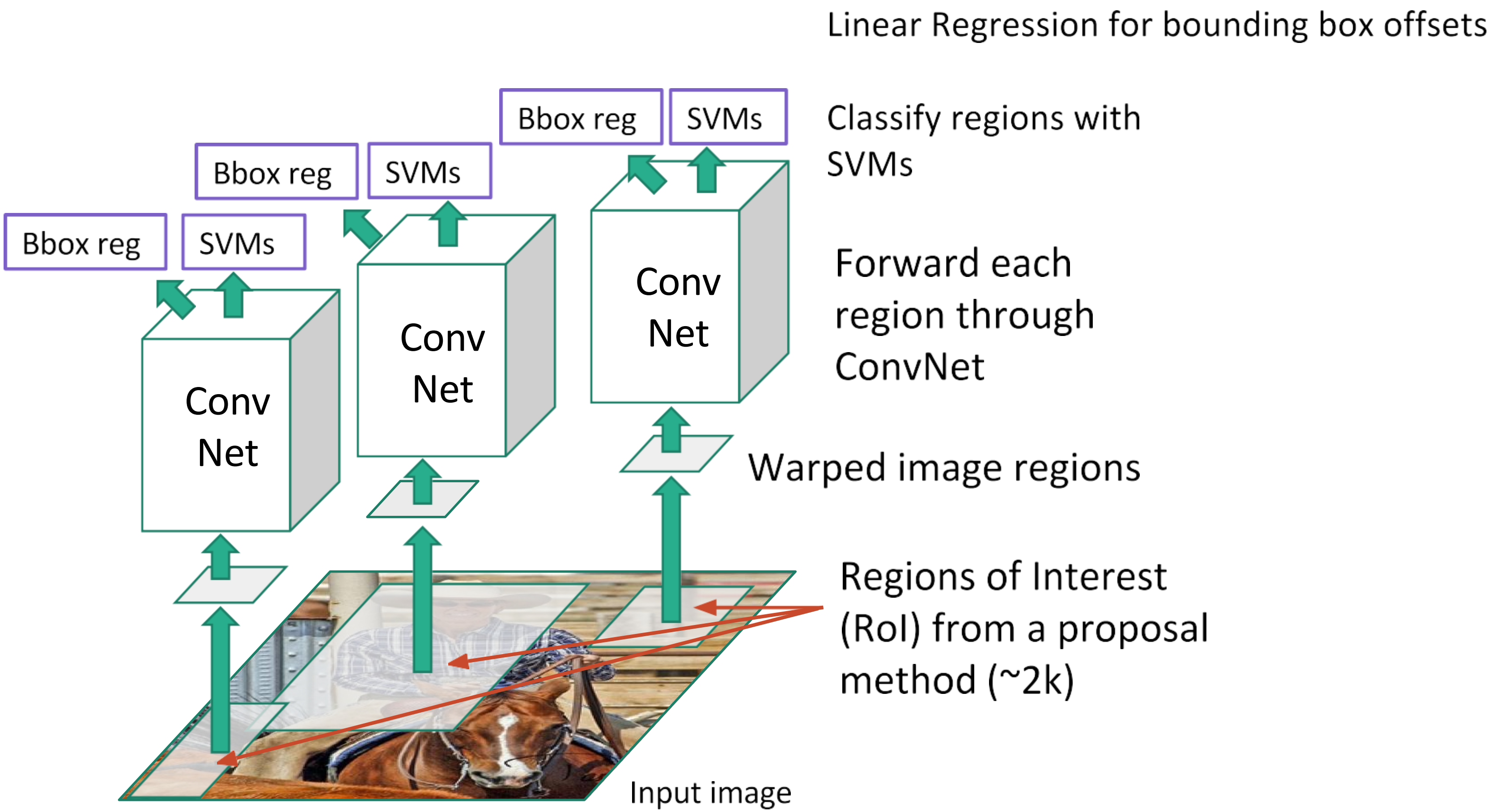
Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

R-CNN on ImageNet detection



Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

R-CNN

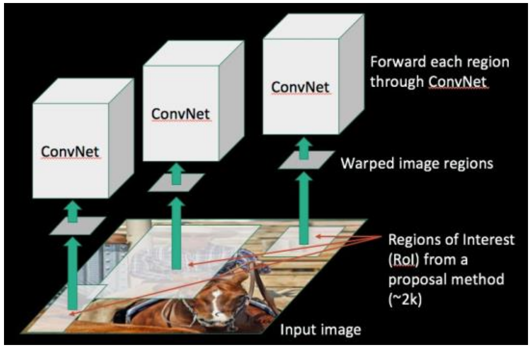


Post hoc component

Girshick et al., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation", CVPR 2014

What’s wrong with slow R-CNN?

- Ad hoc training objectives
  - Fine-tune network with softmax classifier (log loss)
  - Train post-hoc linear SVMs (hinge loss)
  - Train post-hoc bounding-box regressions (least squares)
- Training is slow (84h), takes a lot of disk space
- Inference (detection) is slow
  - 47s / image with VGG16 [Simonyan & Zisserman, ICLR15]



~2000 ConvNet forward passes per image

Girshick, “Fast R-CNN”, ICCV 2015

Fast R-CNN

- Fast test time
- One network, trained in one stage
- Higher mean average precision

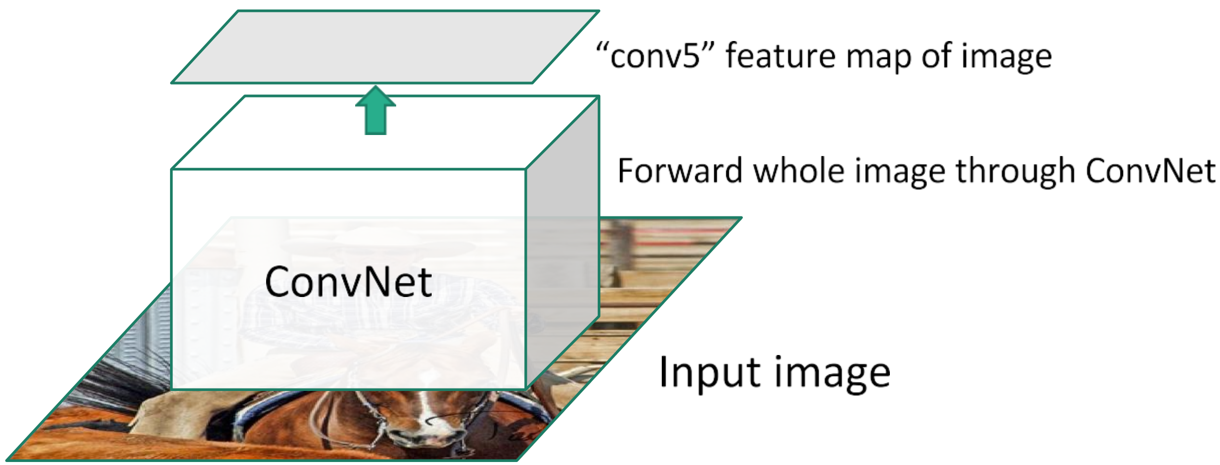
Girshick, "Fast R-CNN", ICCV 2015

Fast R-CNN

Input image

Girshick, "Fast R-CNN", ICCV 2015

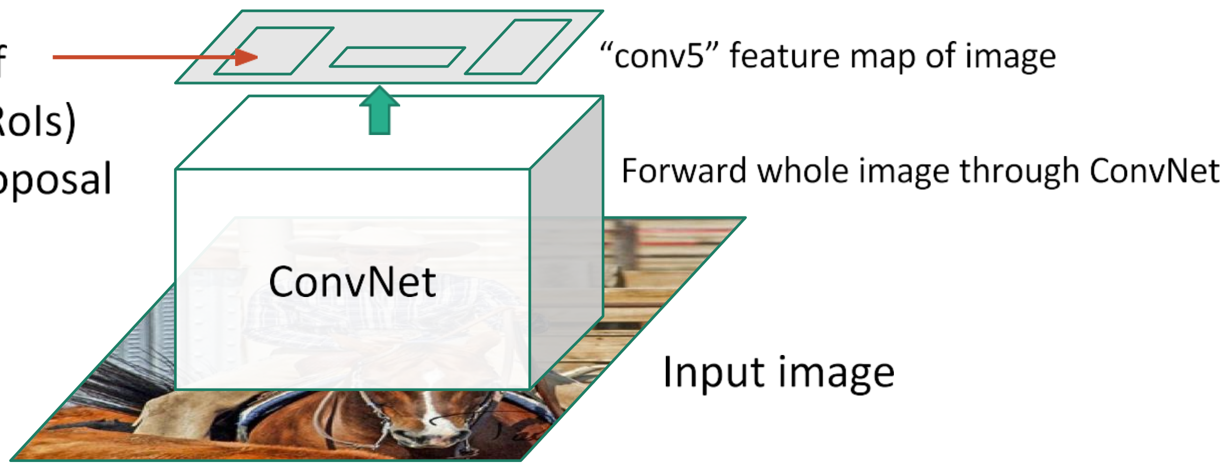
Fast R-CNN



Girshick, "Fast R-CNN", ICCV 2015

Fast R-CNN

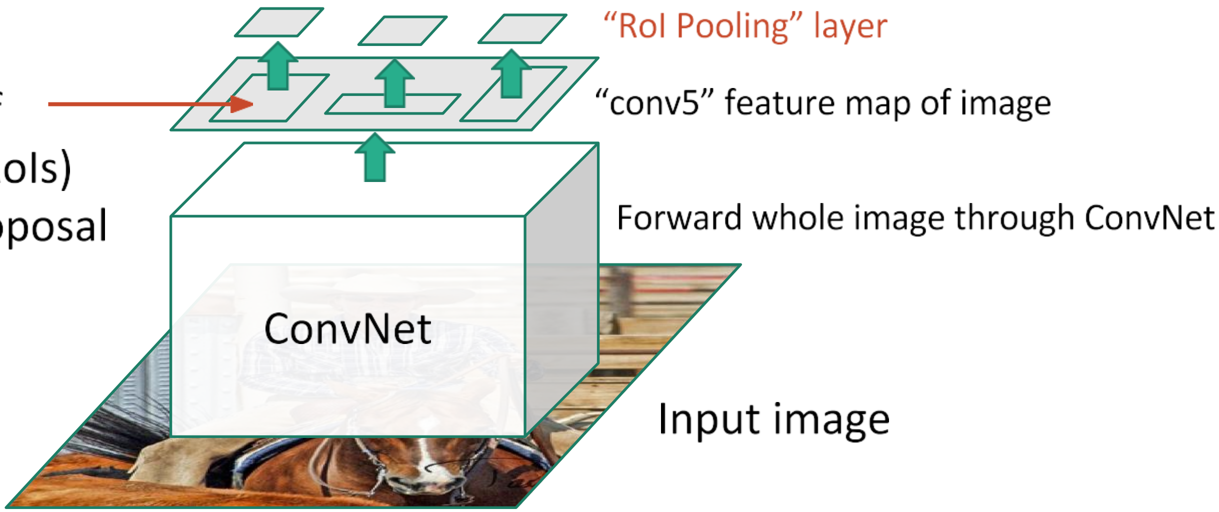
Regions of Interest (RoIs) from a proposal method



Girshick, "Fast R-CNN", ICCV 2015

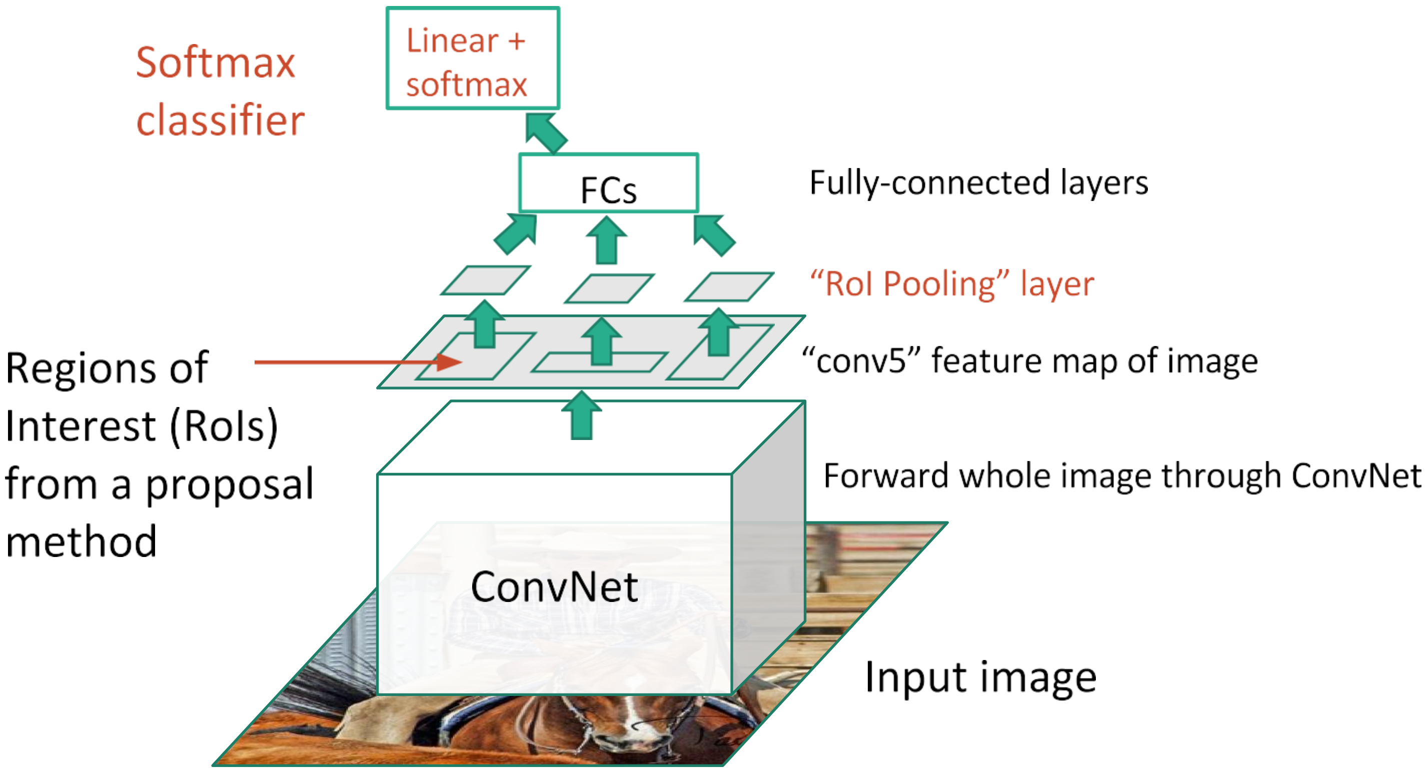
Fast R-CNN

Regions of Interest (RoIs) from a proposal method



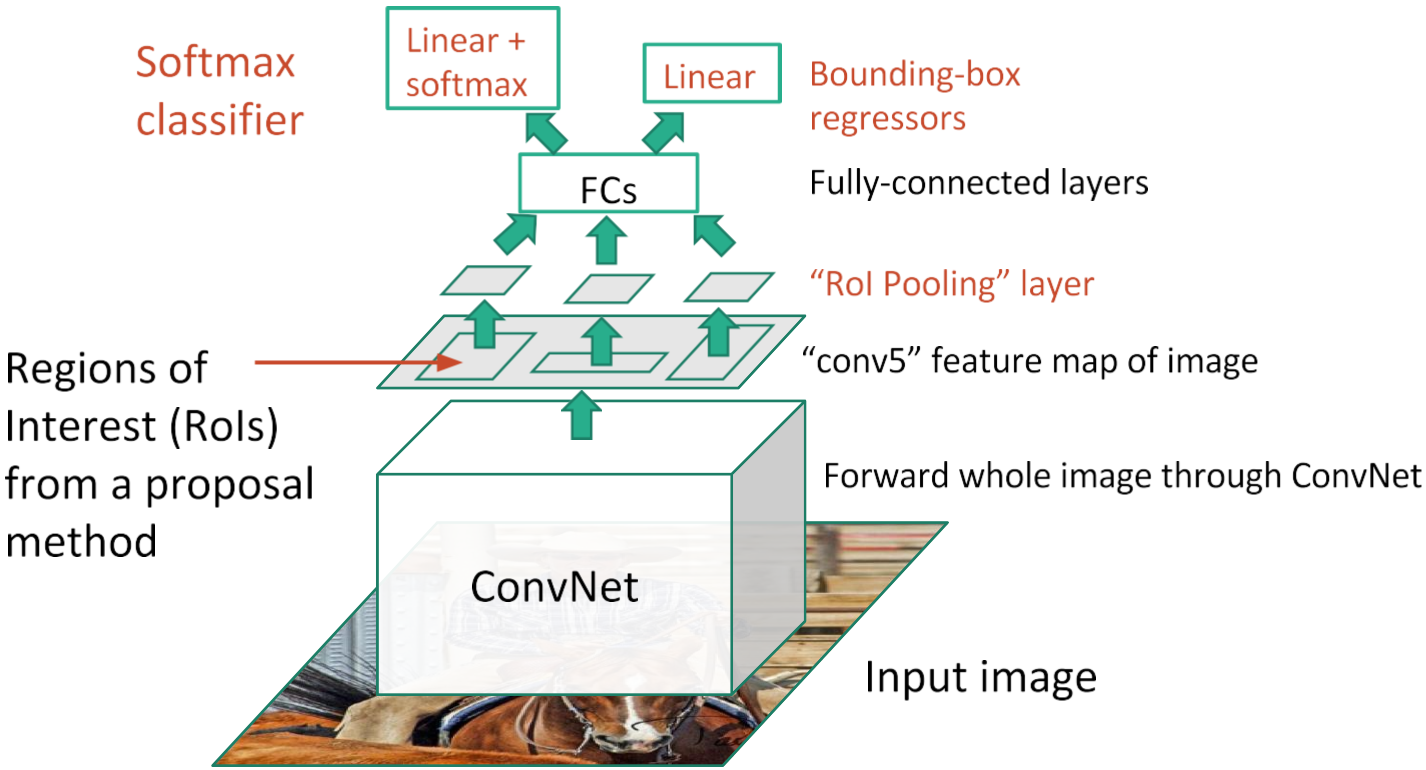
Girshick, "Fast R-CNN", ICCV 2015

Fast R-CNN



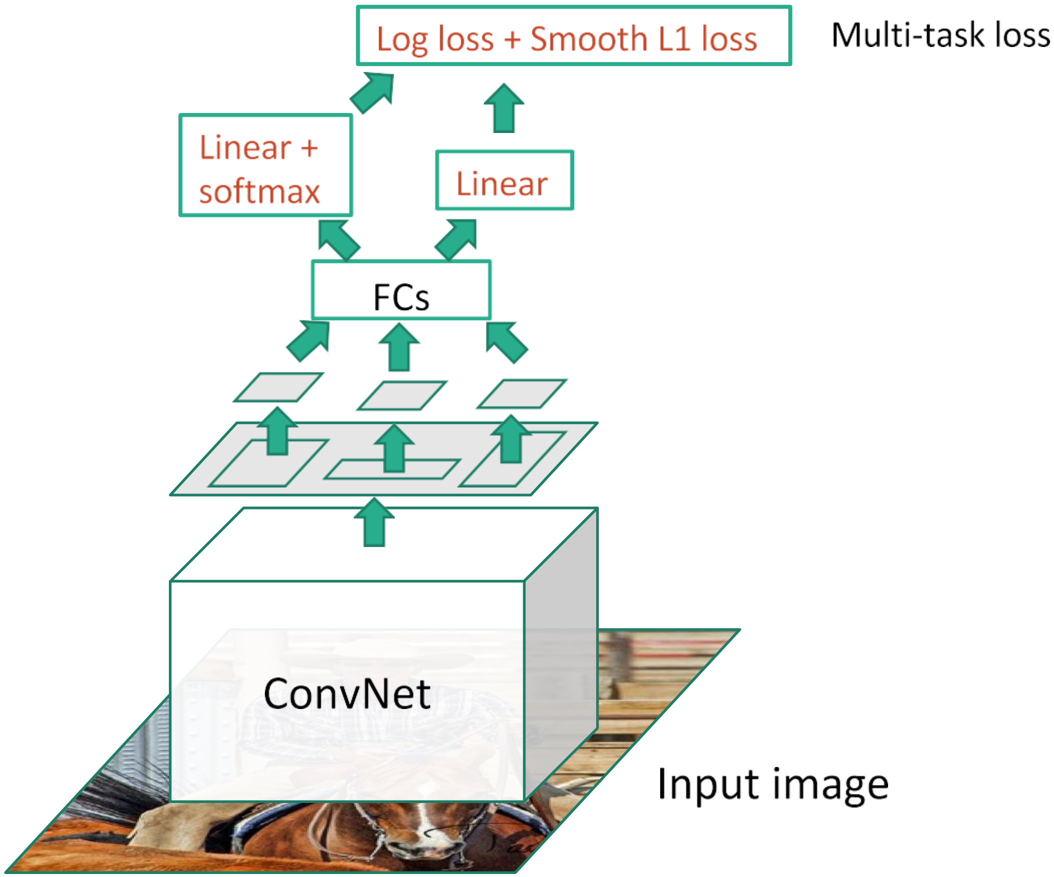
Girshick, "Fast R-CNN", ICCV 2015

Fast R-CNN



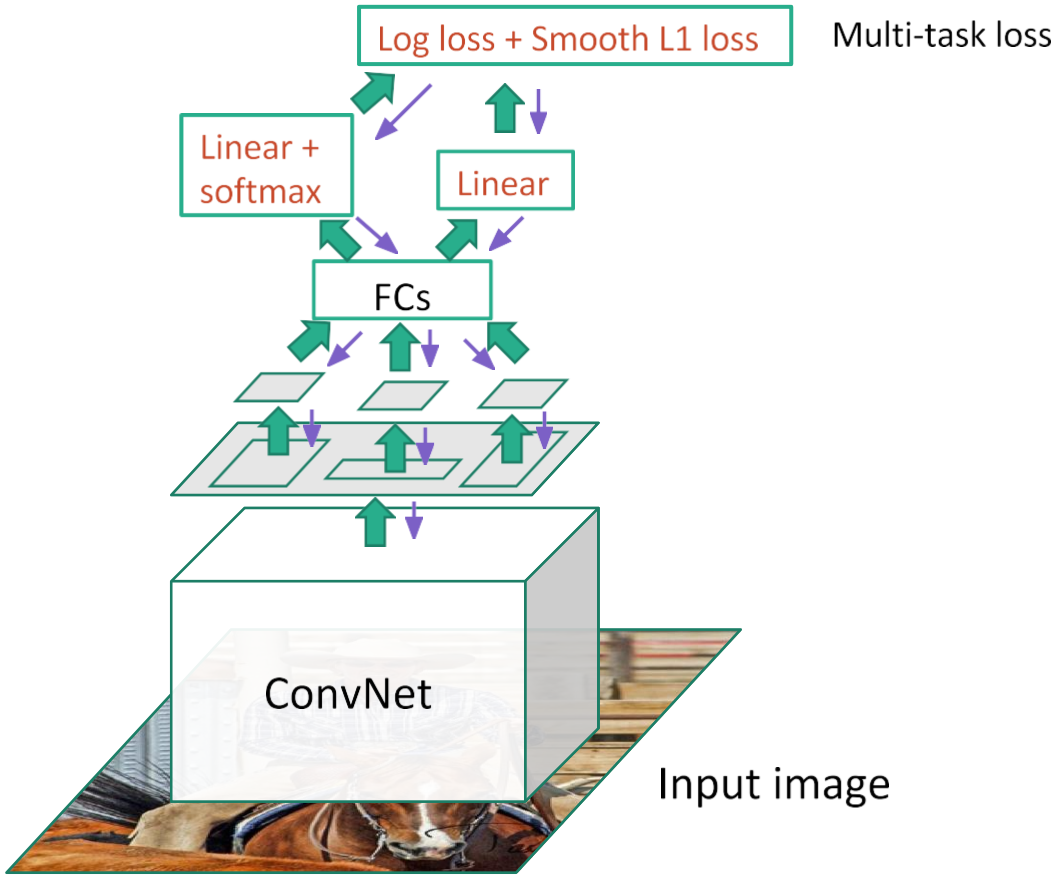
Girshick, "Fast R-CNN", ICCV 2015

Fast R-CNN (Training)



Girshick, "Fast R-CNN", ICCV 2015

Fast R-CNN (Training)



Girshick, "Fast R-CNN", ICCV 2015

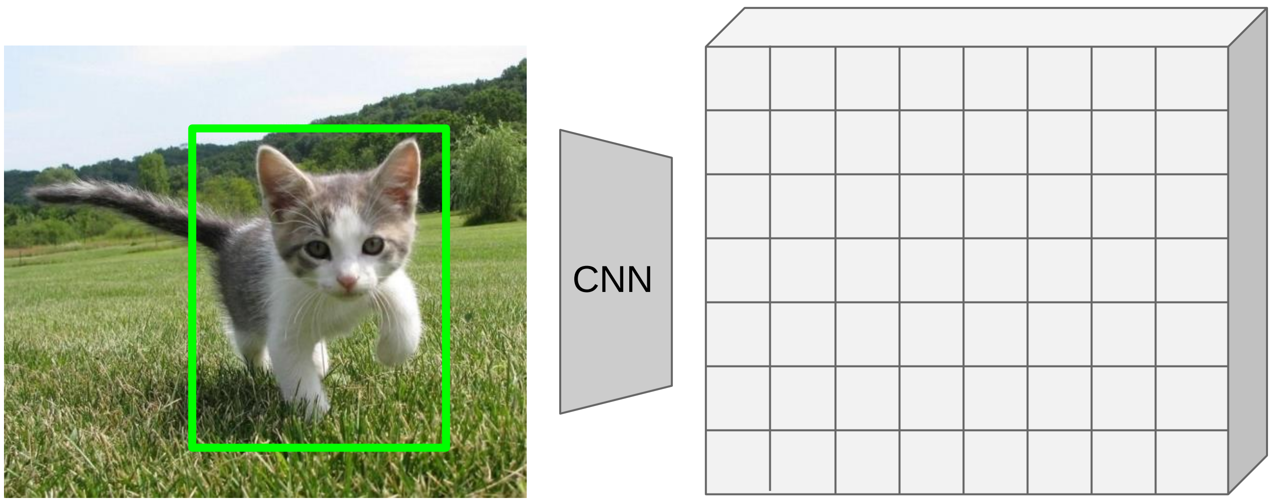
Fast R-CNN vs R-CNN

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

Timings exclude object proposal time, which is equal for all methods.
All methods use VGG16 from Simonyan and Zisserman.

Girshick, “Fast R-CNN”, ICCV 2015

Cropping Features: RoI Pool



Input Image
(e.g. 3 x 640 x 480)

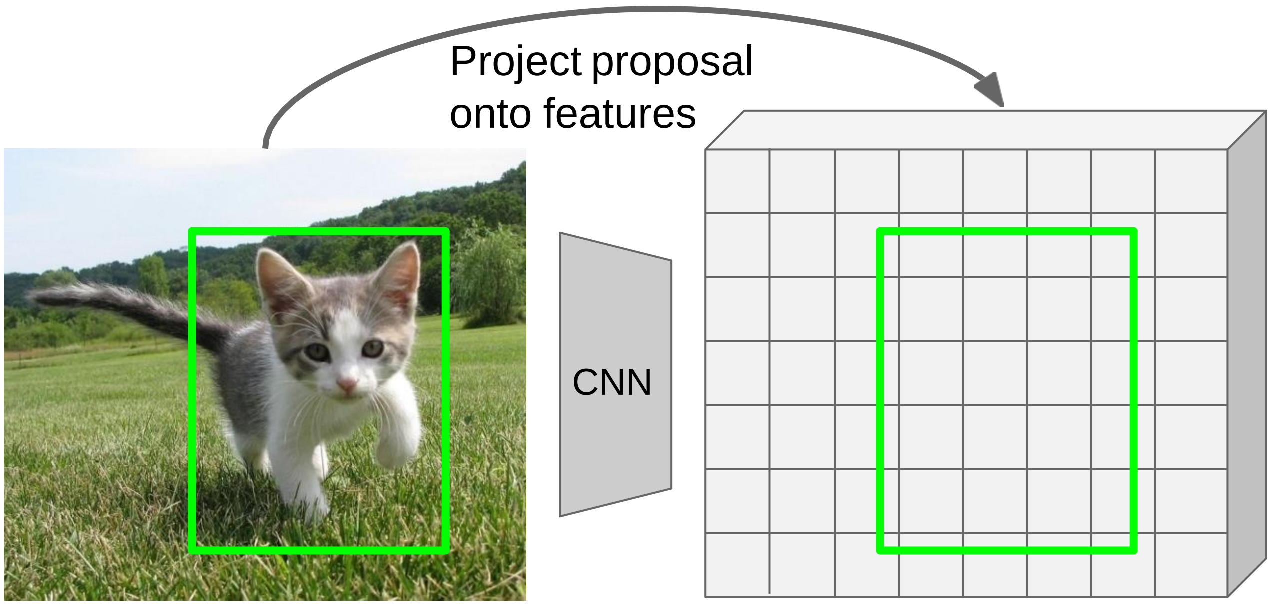
Image features
(e.g. 512 x 20 x 15)

Girshick, “Fast R-CNN”, ICCV 2015.

Girshick, “Fast R-CNN”, ICCV 2015.

Another resource: https://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html#roi-pooling

Cropping Features: RoI Pool



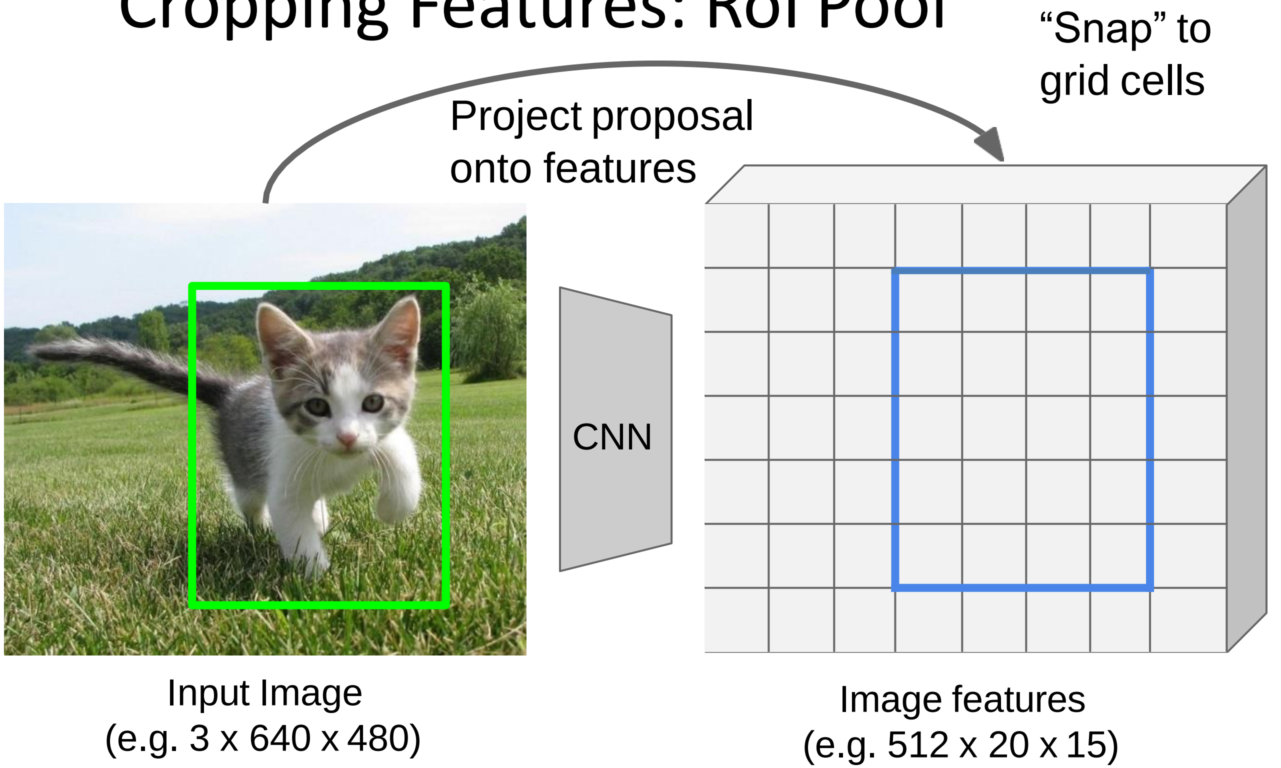
Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

Girshick, “Fast R-CNN”, ICCV 2015.

Girshick, “Fast R-CNN”, ICCV 2015.

Cropping Features: RoI Pool

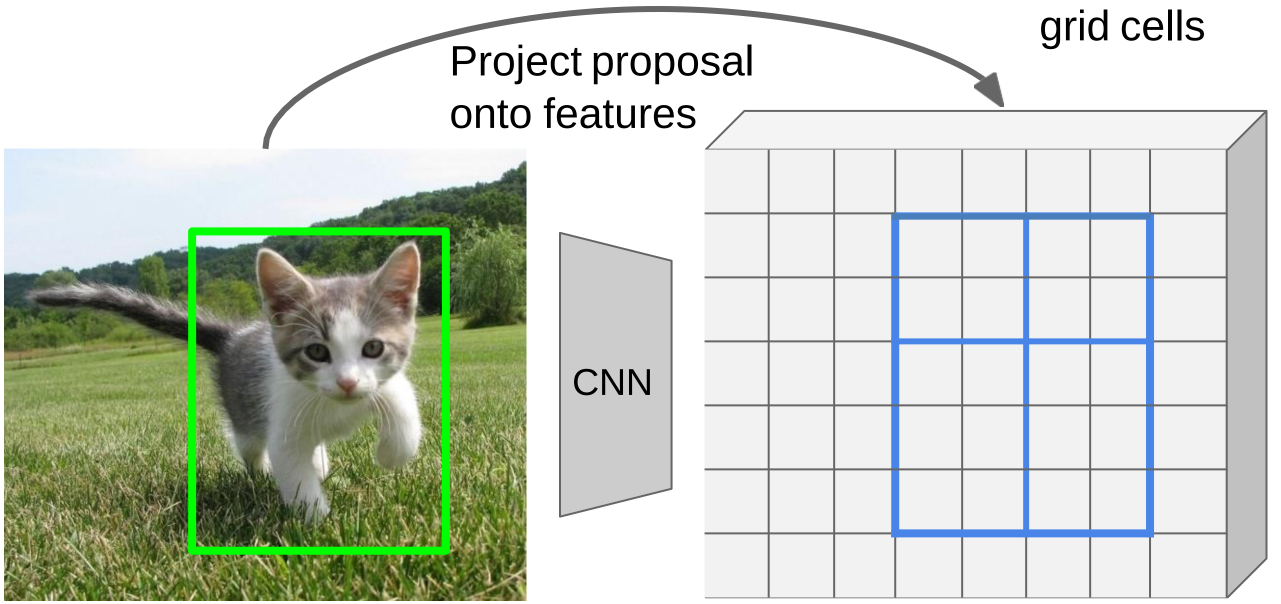


Girshick, "Fast R-CNN", ICCV 2015.

Cropping Features: RoI Pool

“Snap” to grid cells

Divide into 2x2 grid of (roughly) equal subregions



Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

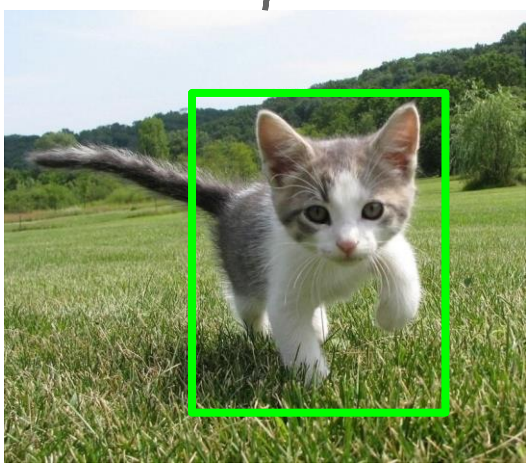
Girshick, “Fast R-CNN”, ICCV 2015.

Cropping Features: RoI Pool

Project proposal
onto features

“Snap” to
grid cells

Divide into 2x2
grid of (roughly)
equal subregions



Input Image
(e.g. 3 x 640 x 480)

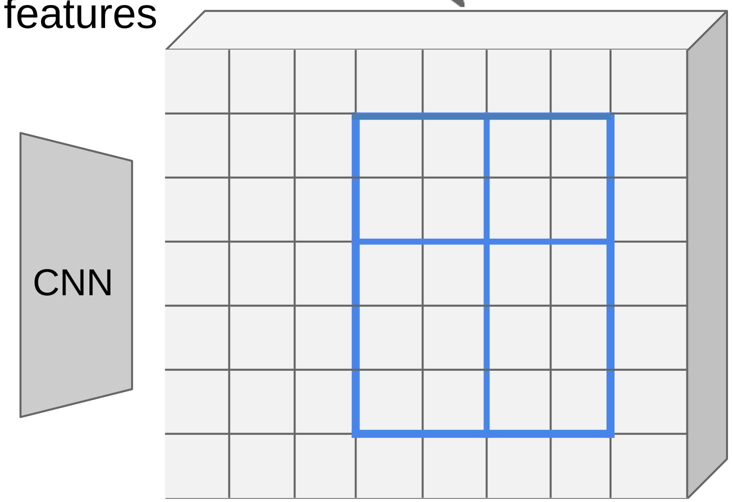
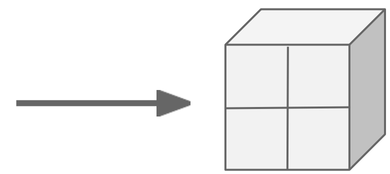


Image features
(e.g. 512 x 20 x 15)

Max-pool within
each subregion



Region features
(here 512 x 2 x 2;
In practice e.g 512 x 7 x 7)

Region features always the
same size even if input
regions have different sizes!

Girshick, “Fast R-CNN”, ICCV 2015.

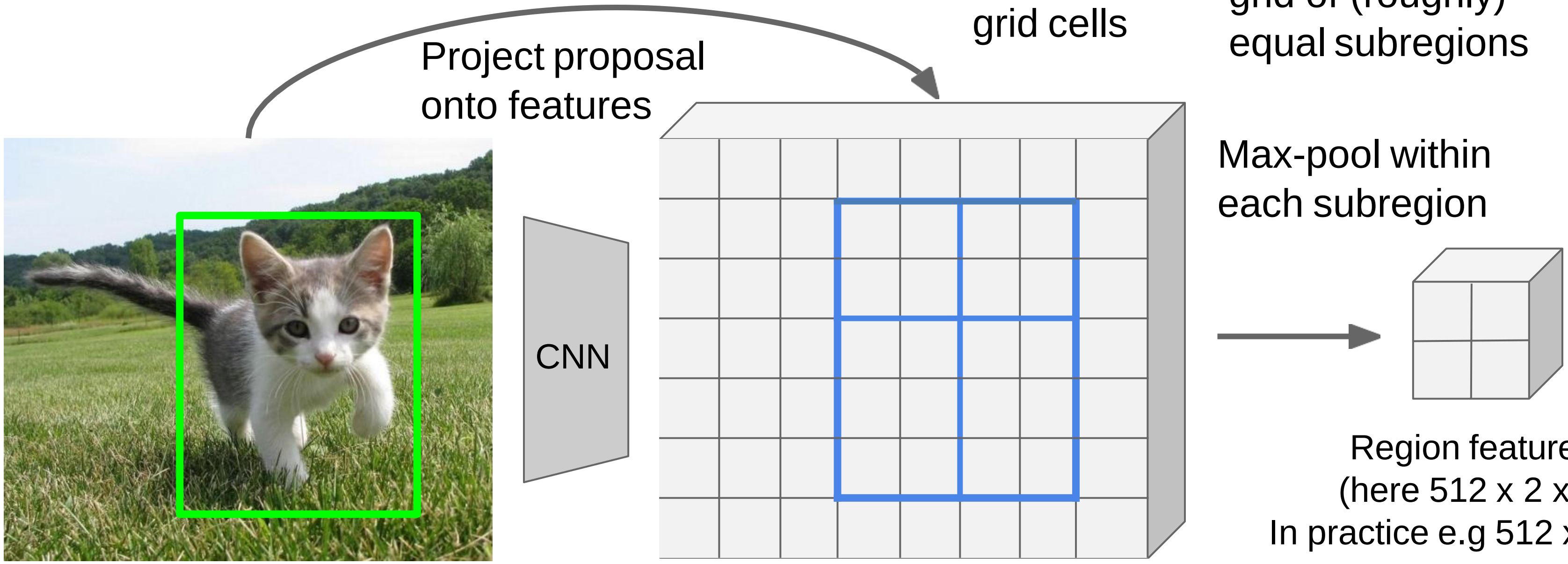
Cropping Features: RoI Pool

Project proposal onto features

"Snap" to grid cells

Divide into 2x2 grid of (roughly) equal subregions

Max-pool within each subregion



Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

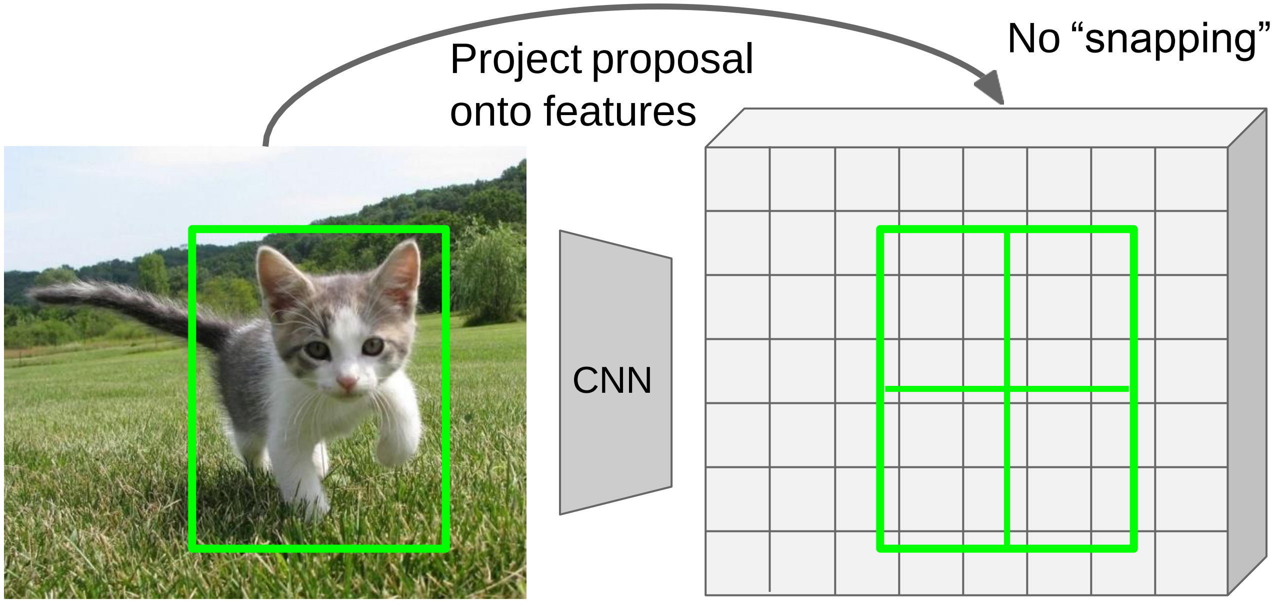
Region features
(here 512 x 2 x 2;
In practice e.g 512 x 7 x 7)

Region features always the same size even if input regions have different sizes!

Problem: Region features slightly misaligned

Girshick, "Fast R-CNN", ICCV 2015.

Cropping Features: RoI Align



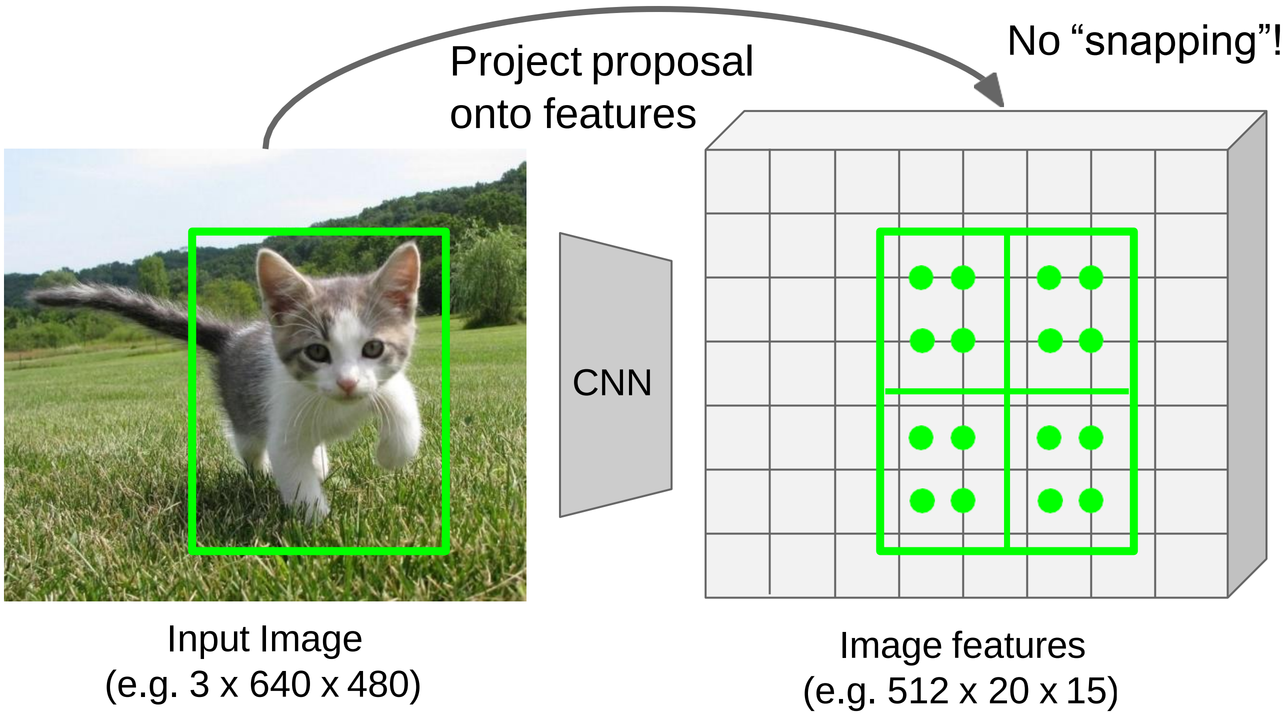
Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

He et al, “Mask R-CNN”, ICCV 2017

Cropping Features: RoI Align

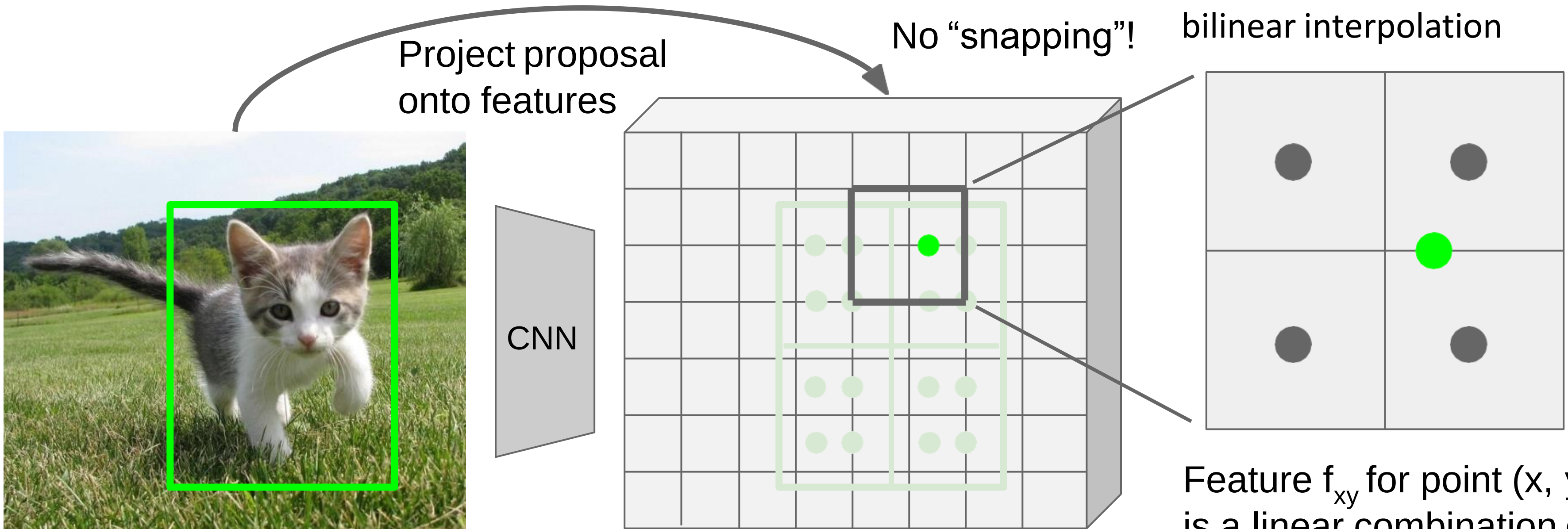
Sample at regular points in each subregion using bilinear interpolation



He et al, “Mask R-CNN”, ICCV 2017

Cropping Features: RoI Align

Sample at regular points in each subregion using bilinear interpolation



Input Image
(e.g. 3 x 640 x 480)

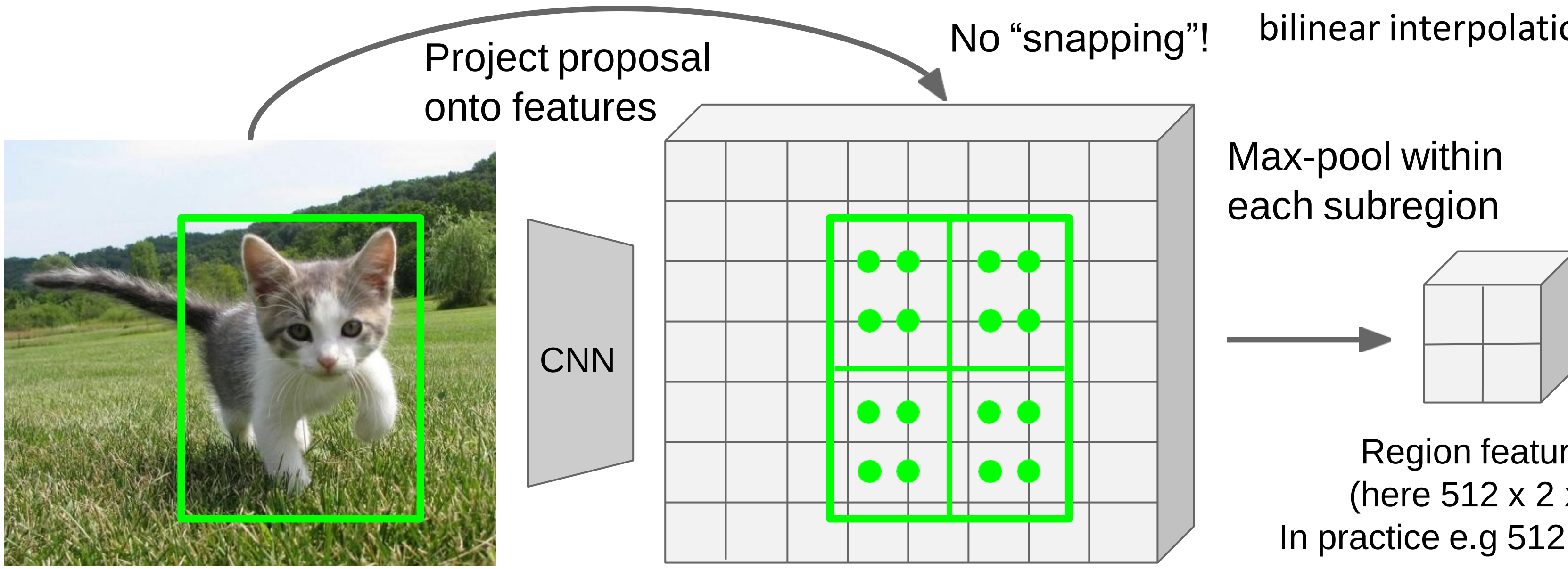
Image features
(e.g. 512 x 20 x 15)

Feature f_{xy} for point (x, y) is a linear combination of features at its four neighboring grid cells

He et al, “Mask R-CNN”, ICCV 2017

Cropping Features: RoI Align

Sample at regular points in each subregion using bilinear interpolation



Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

Region features
(here 512 x 2 x 2;
In practice e.g 512 x 7 x 7)

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

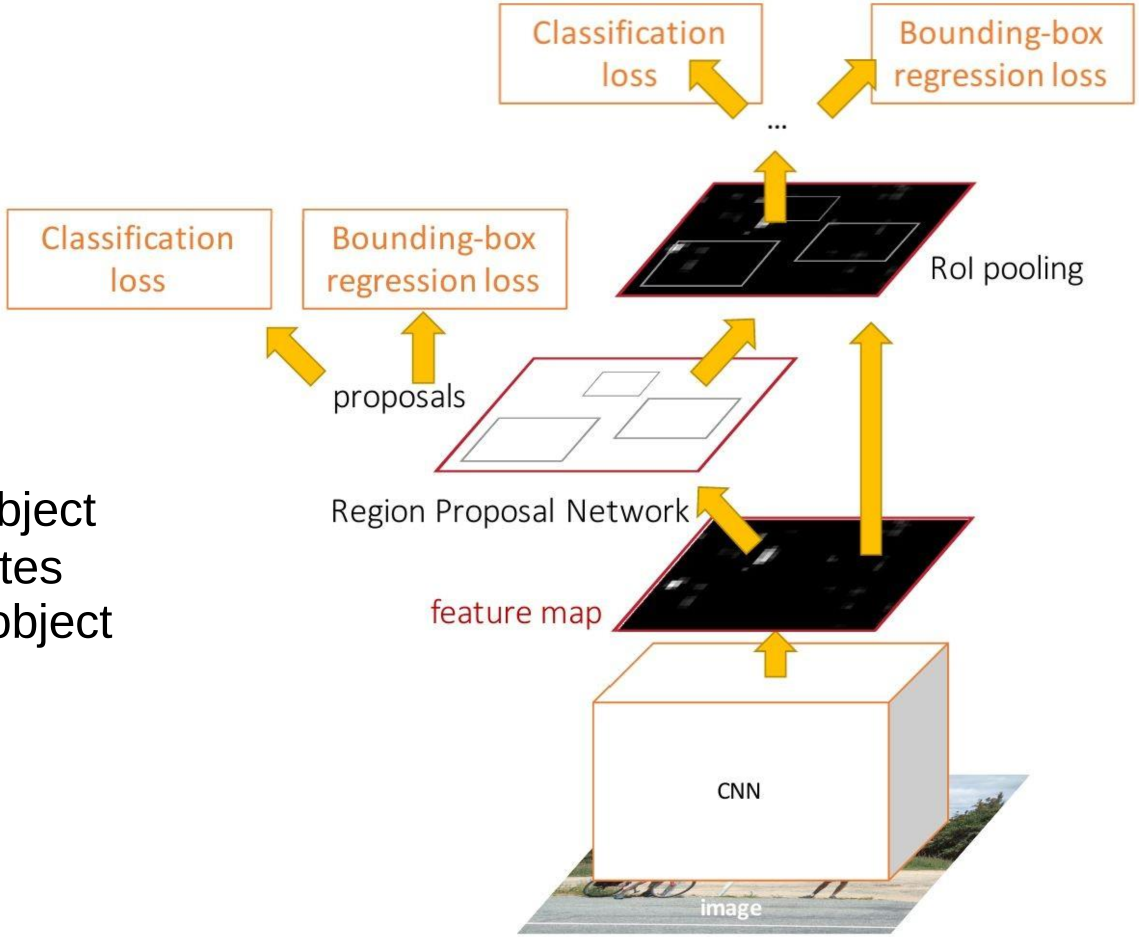
Faster R-CNN

Make CNN do proposals!

Insert **Region Proposal Network (RPN)** to predict proposals from features

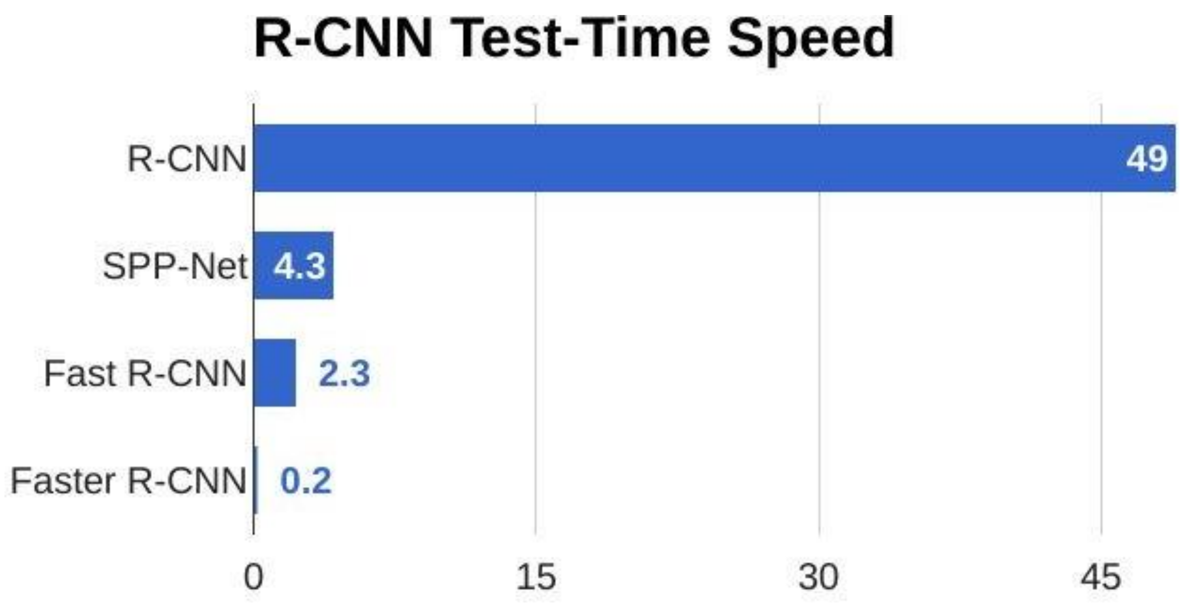
Jointly train with 4 losses:

1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates

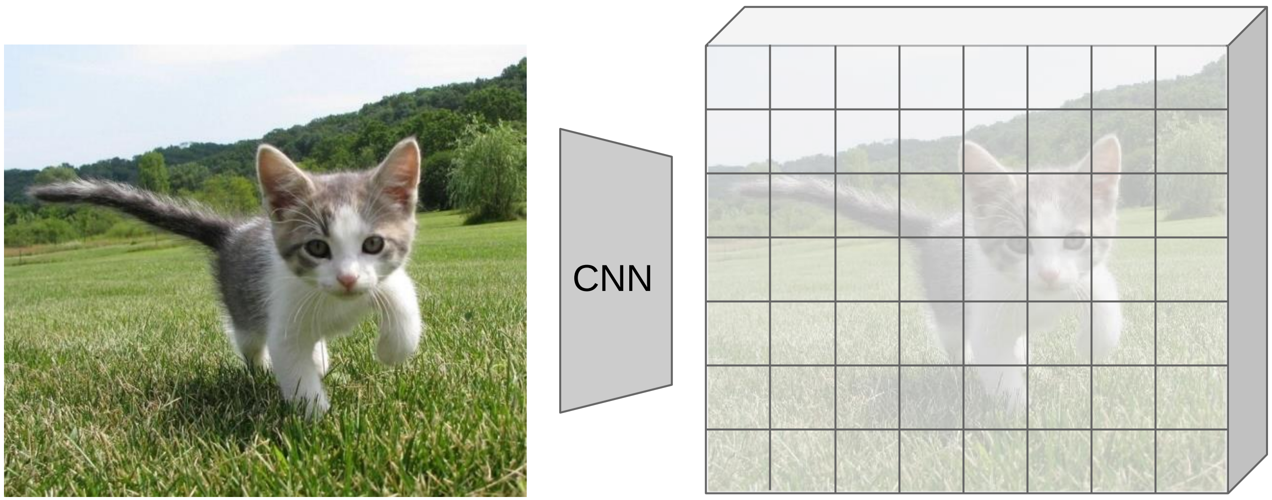


Ren et al, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", NIPS 2015

Faster R-CNN


R-CNN Test-Time Speed
R-CNN
SPP-Net
Fast R-CNN
Faster R-CNN
49
4.3
2.3
0.2
0
15
30
45

Region Proposal Network

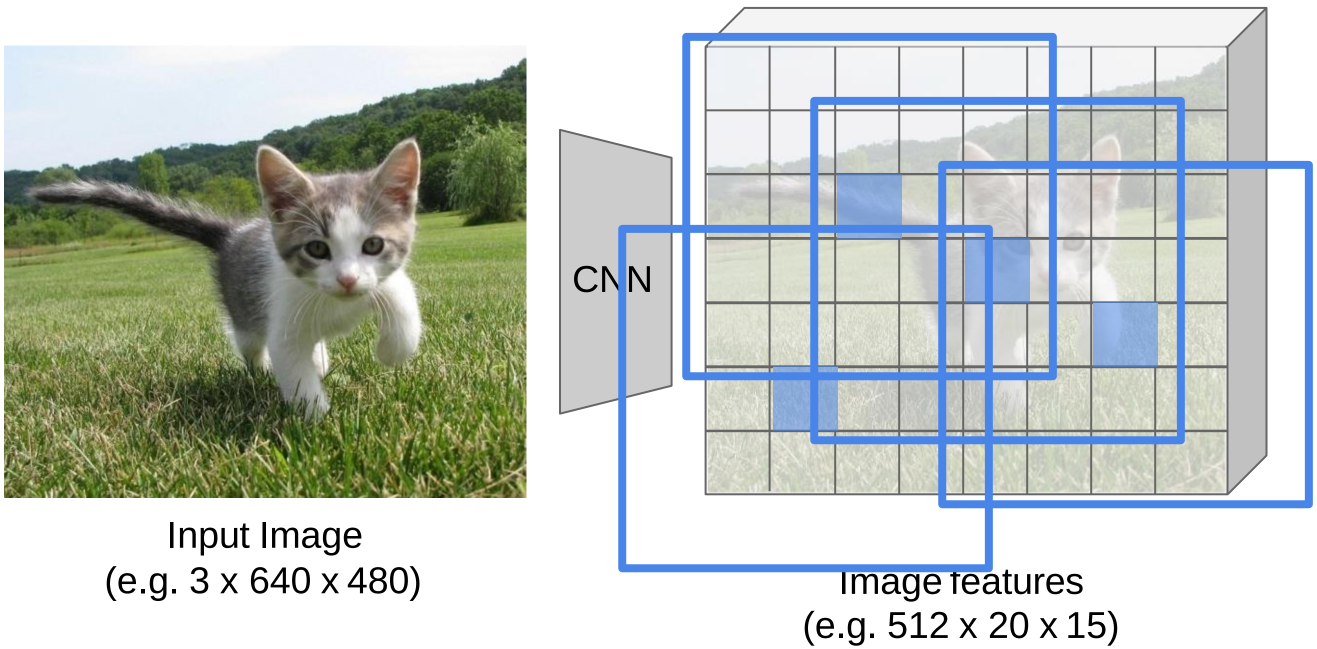


Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

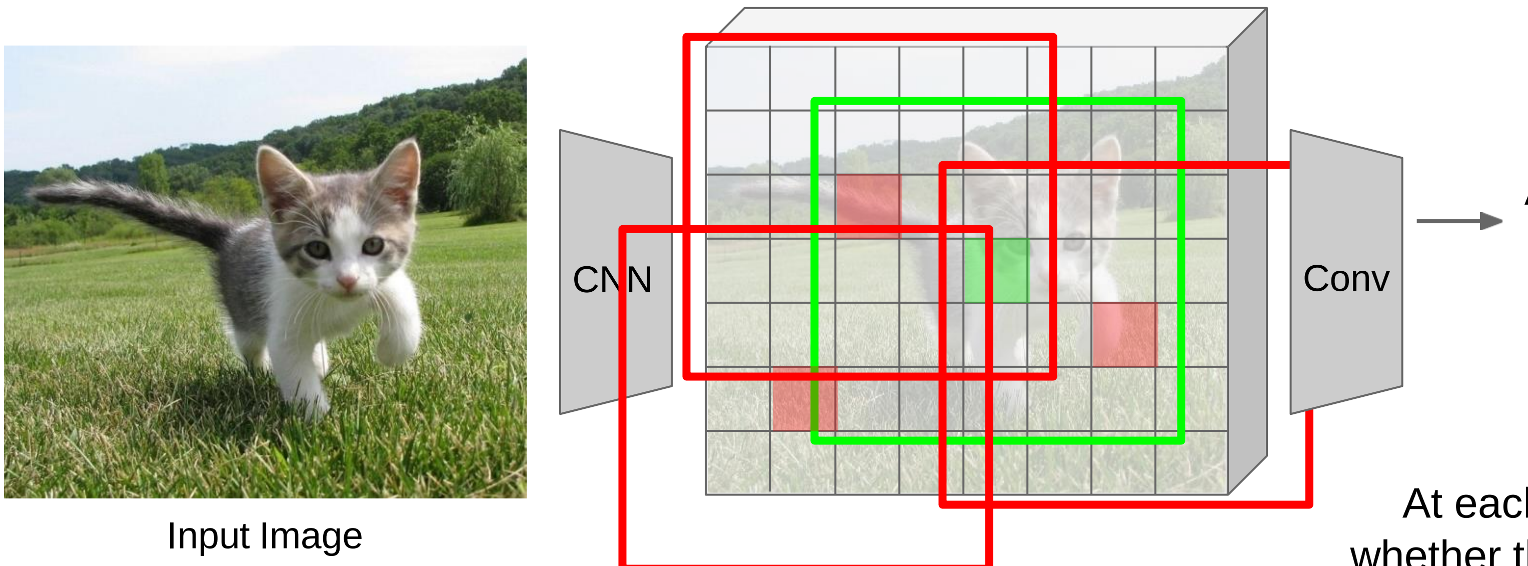
Region Proposal Network

Imagine an **anchor box** of fixed size at each point in the feature map



Region Proposal Network

Imagine an **anchor box** of fixed size at each point in the feature map



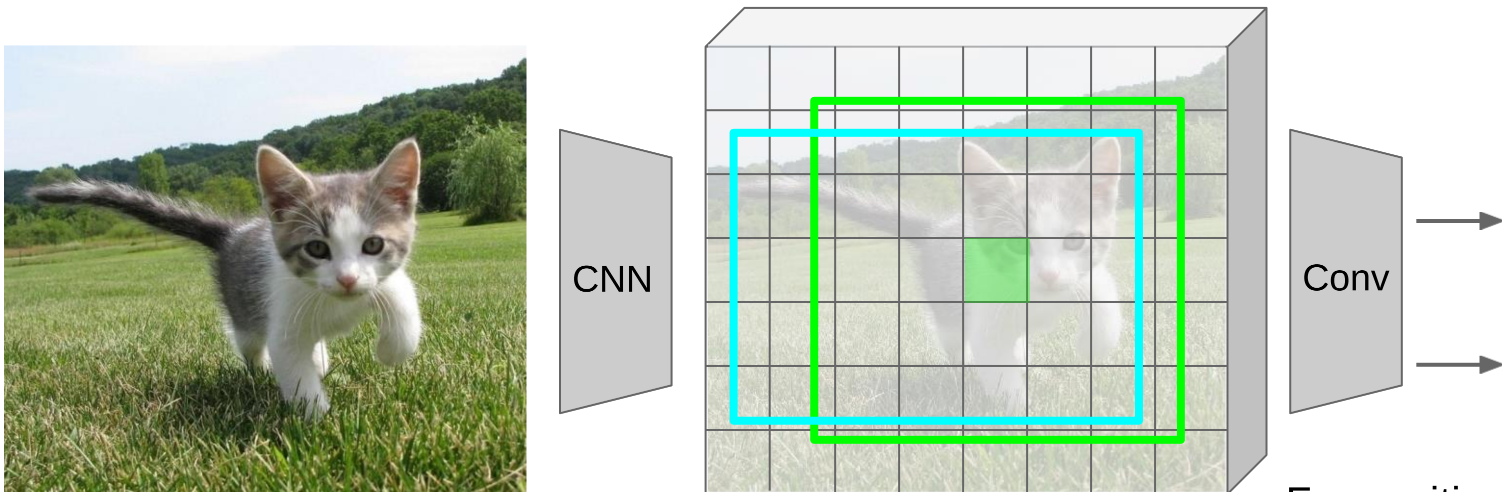
Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

At each point, predict whether the corresponding anchor contains an object (per-pixel logistic regression)

Region Proposal Network

Imagine an **anchor box** of fixed size at each point in the feature map



Anchor is an object?
1 x 20 x 15

Box transforms
4 x 20 x 15

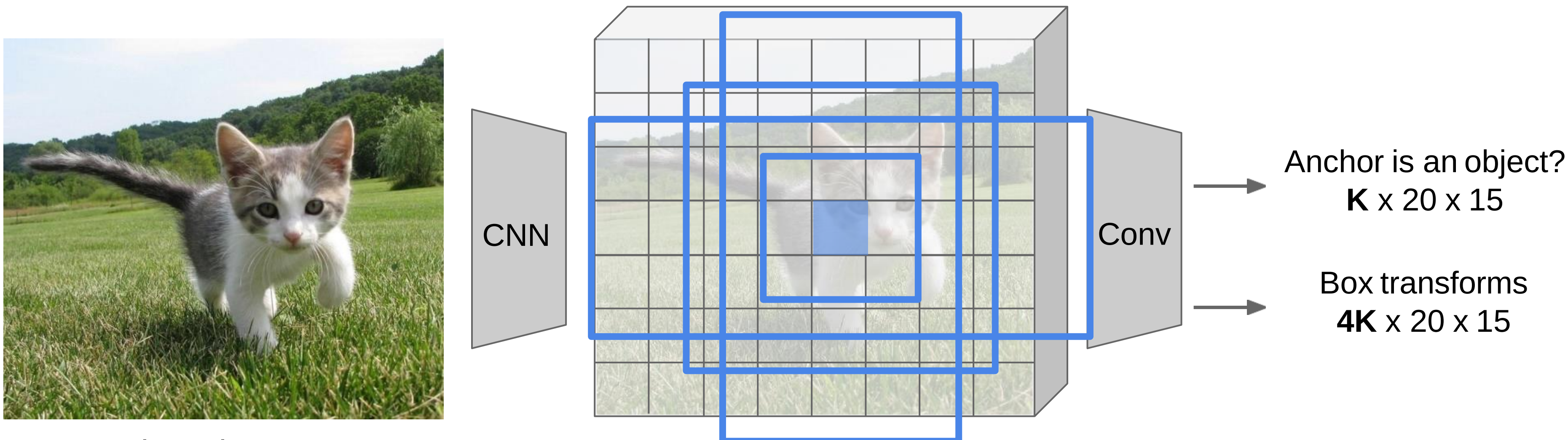
Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

For positive boxes, also predict a transformation from the anchor to the ground-truth box (regress 4 numbers per pixel)

Region Proposal Network

In practice use K different anchor boxes of different size / scale at each point

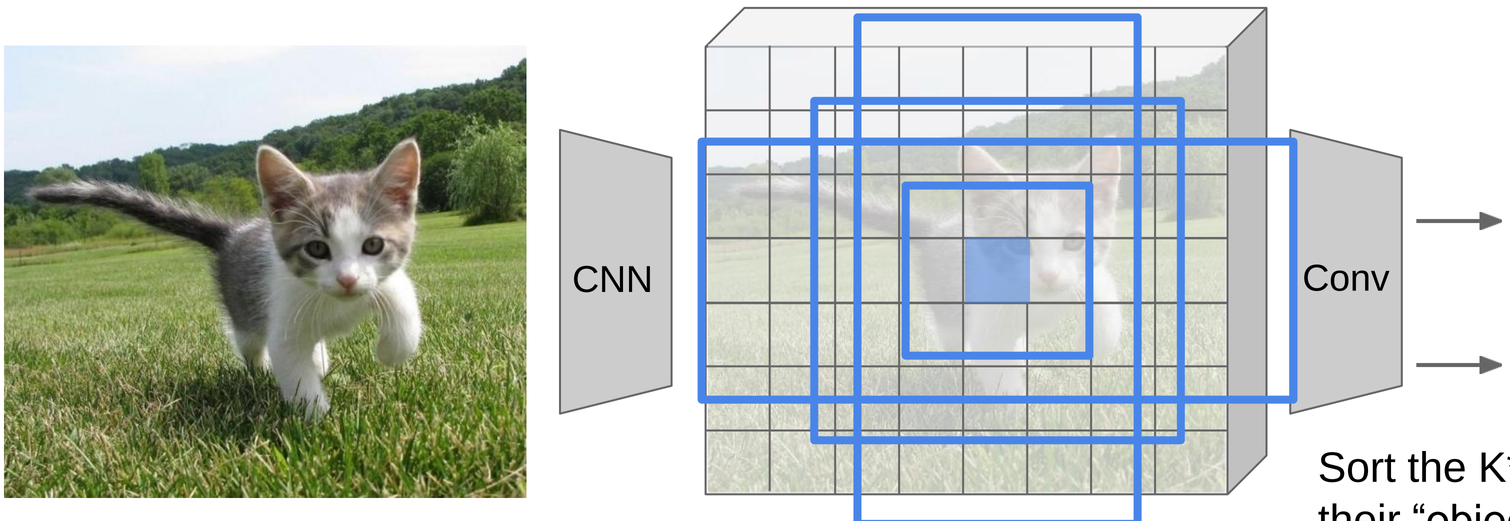


Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

Region Proposal Network

In practice use K different anchor boxes of different size / scale at each point



Input Image
(e.g. 3 x 640 x 480)

Image features
(e.g. 512 x 20 x 15)

Anchor is an object?
K x 20 x 15

Box transforms
4K x 20 x 15

Sort the K*20*15 boxes by their “object” score, take top ~300 as our proposals

Plan for this lecture

- Fully supervised detection
  - Pre-CNN: Deformable part models
  - Detection with region proposals: R-CNN, Fast/er R-CNN
  - Detection without region proposals: YOLO
  - Semantic and instance segmentation: FCN, Mask R-CNN
- Weak or out-of-domain supervision
  - Weakly supervised object detection
  - Domain adaptation

Accurate object detection is slow!

	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img

⅓ Mile, 1760 feet

Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016

Accurate object detection is slow!

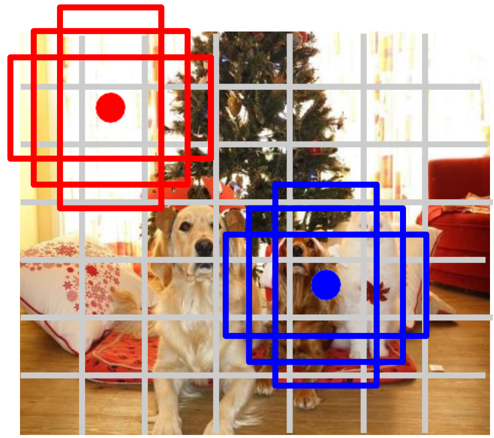
	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
Fast R-CNN	70.0	.5 FPS	2 s/img
Faster R-CNN	73.2	7 FPS	140 ms/img
YOLO	69.0	45 FPS	22 ms/img

2 feet

Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016

Detection without Proposals: YOLO / SSD

Input image
3 x H x W



Divide image into grid
7 x 7

Image a set of **base boxes** centered at each grid cell
Here B = 3

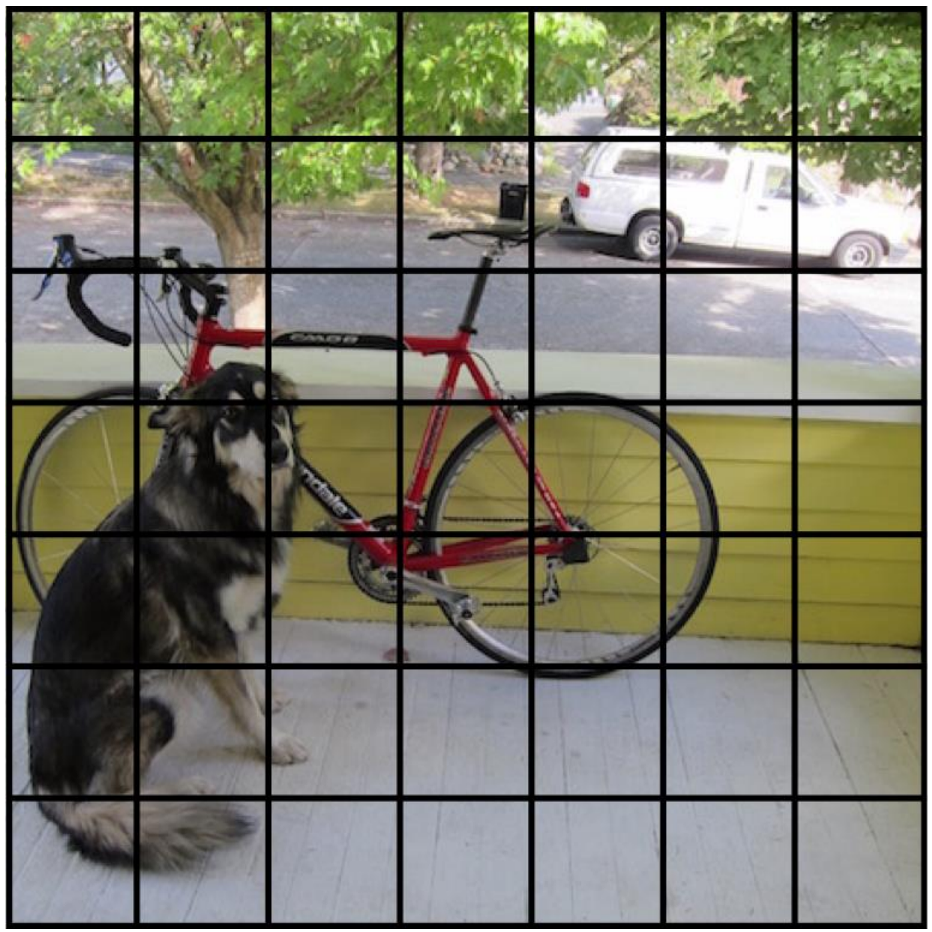
Within each grid cell:
- Regress from each of the B base boxes to a final box with 5 numbers:
  (dx, dy, dh, dw, confidence)
- Predict scores for each of C classes (including background as a class)

Output:
7 x 7 x (5 * B + C)

Redmon et al, "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016
Liu et al, "SSD: Single-Shot MultiBox Detector", ECCV 2016

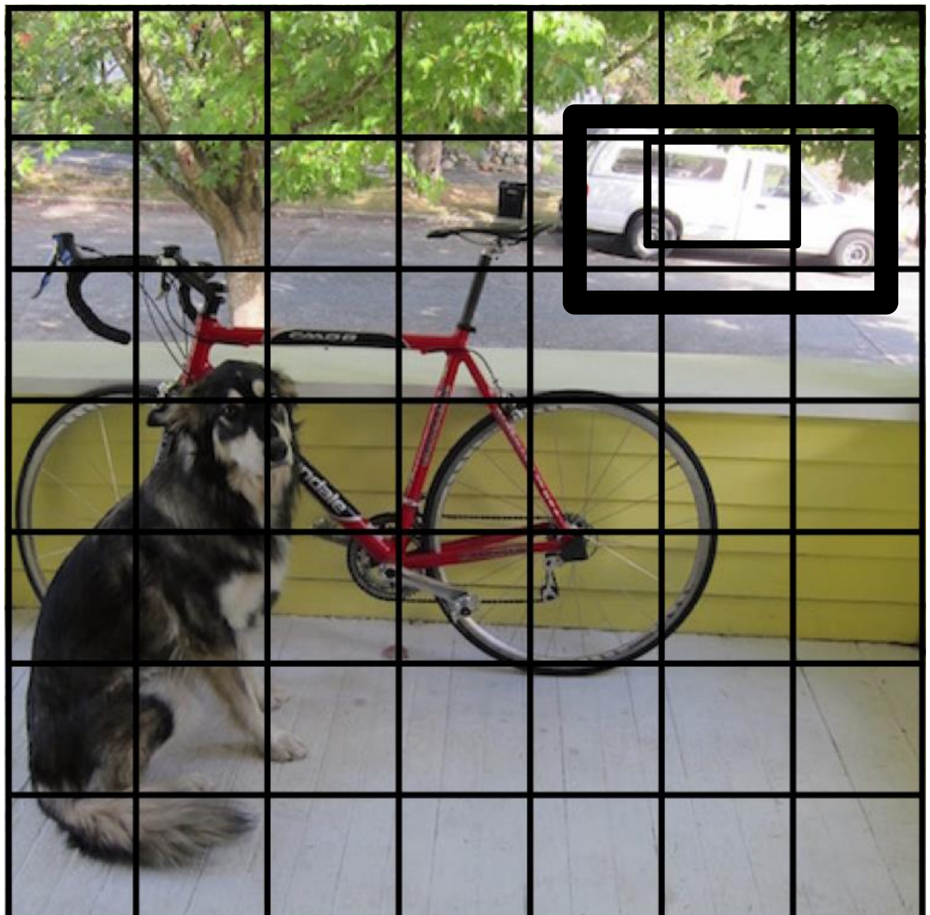
Slide by: Justin Johnson

Split the image into a grid



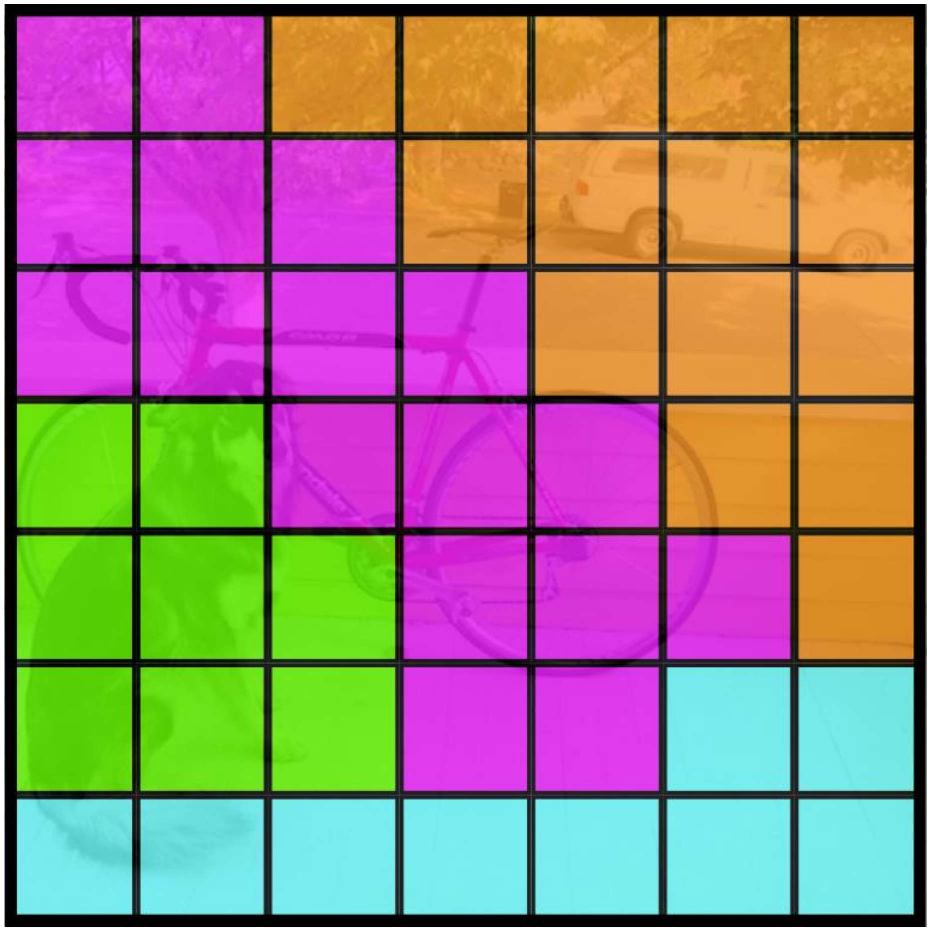
Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016

Each cell predicts boxes and confidences: P(Object)



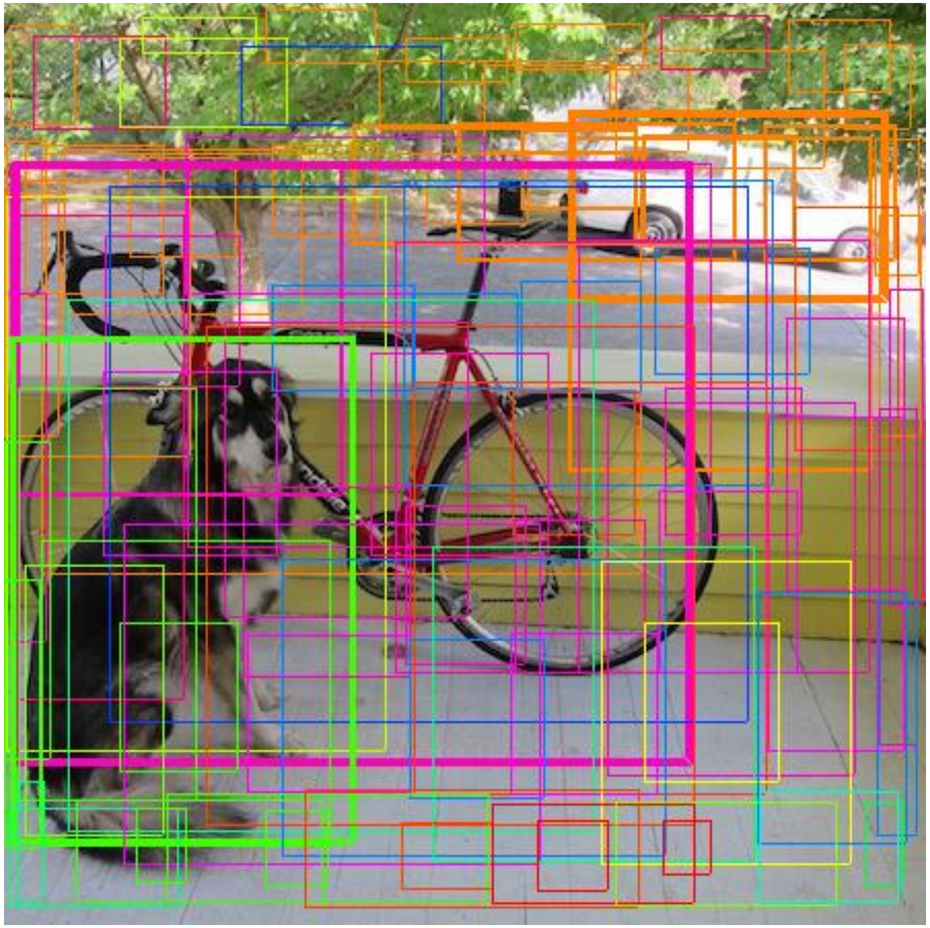
Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016

Each cell also predicts a probability P(Class | Object)



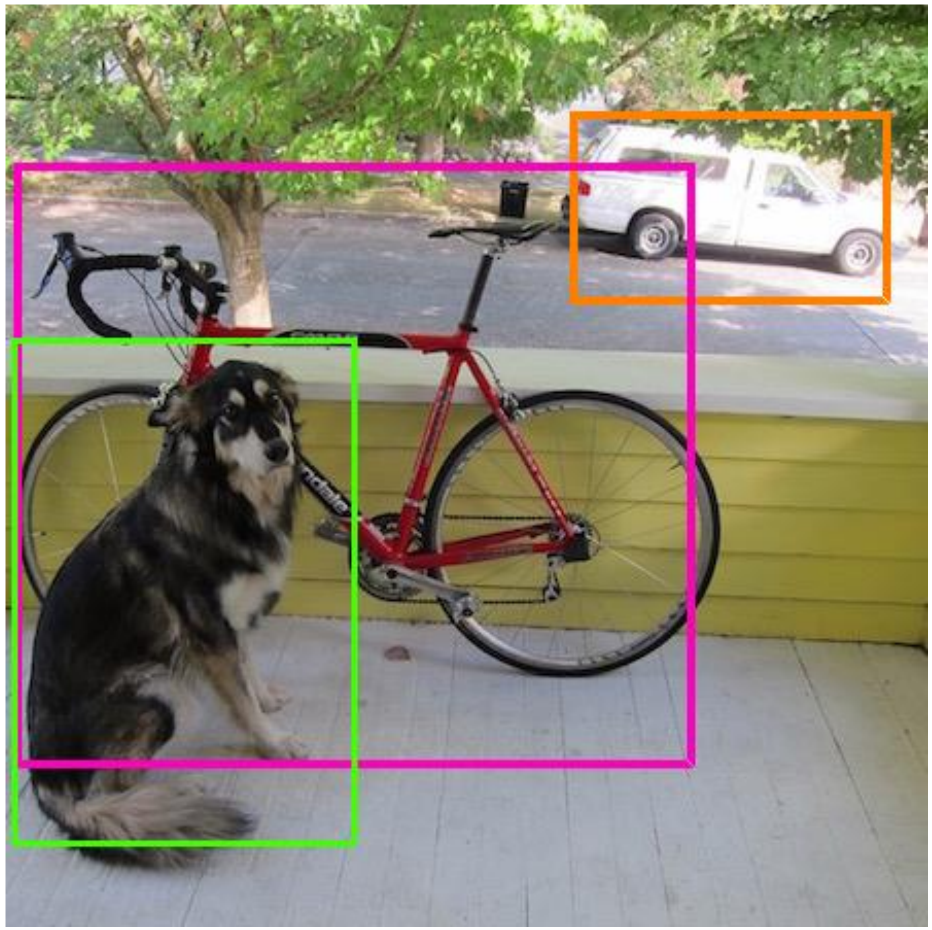
Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016

Combine the box and class predictions



Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016

Finally do NMS and threshold detections

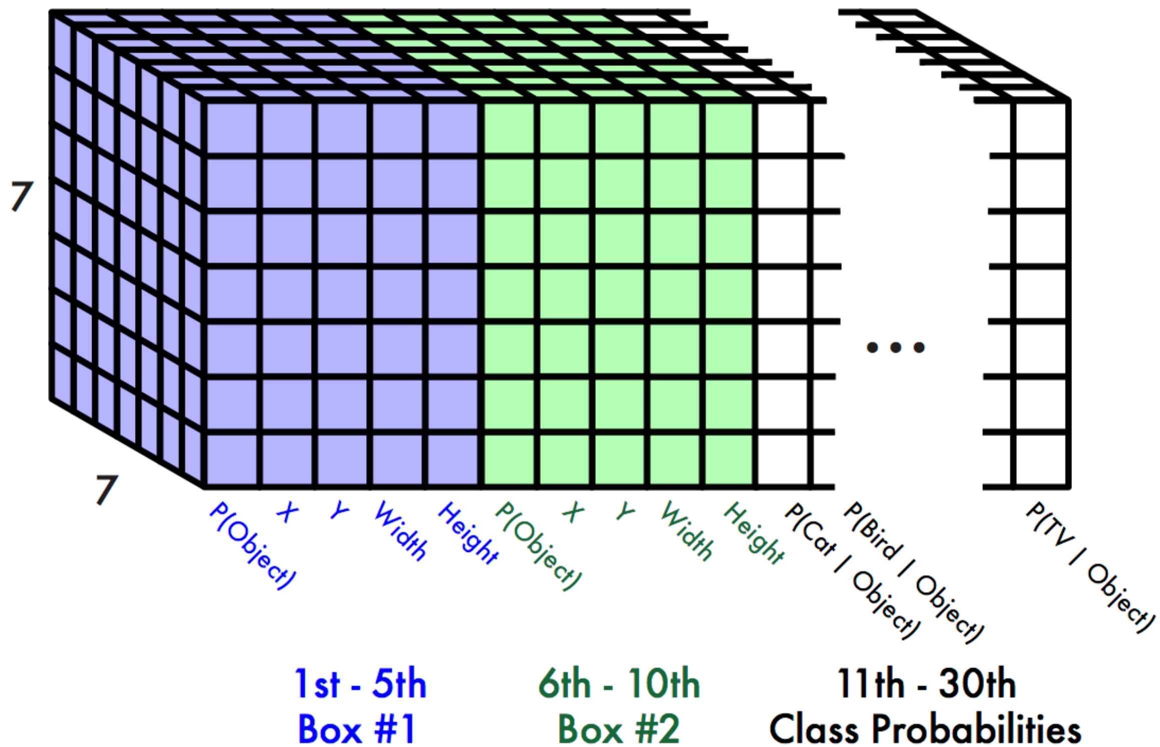


Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016

This parameterization fixes the output size

Each cell predicts:

- For each bounding box:
  - 4 coordinates (x, y, w, h)
  - 1 confidence value
- Some number of class probabilities



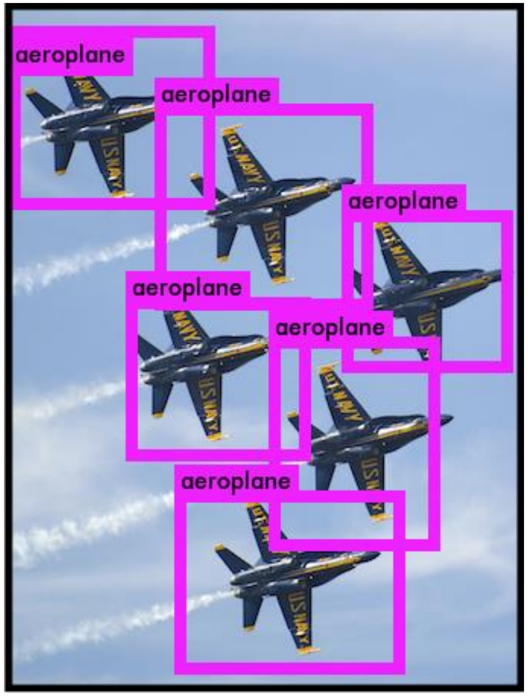
For Pascal VOC:

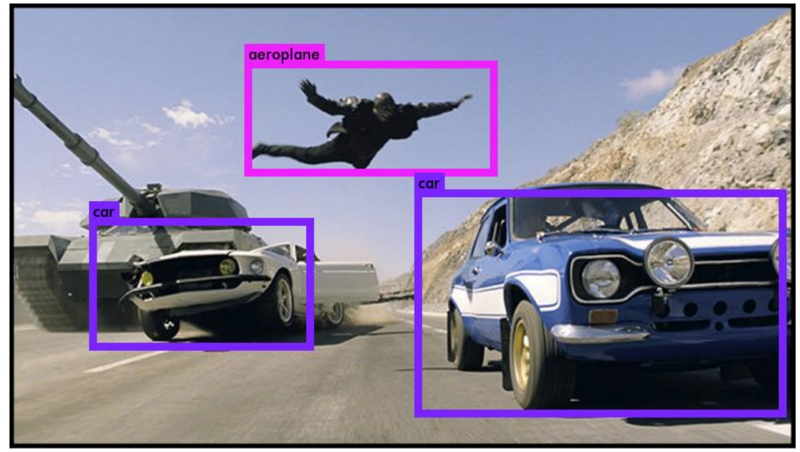
- 7x7 grid
- 2 bounding boxes / cell
- 20 classes

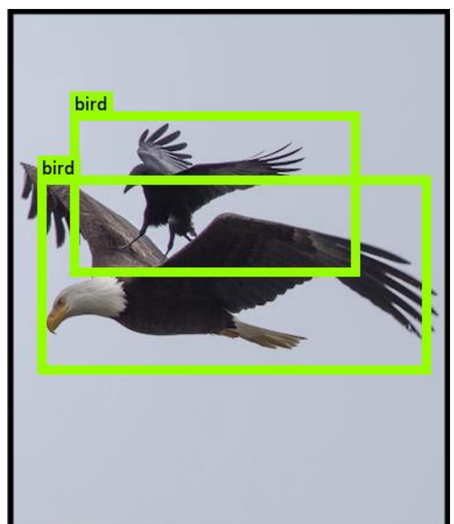
7 x 7 x (2 x 5 + 20) = 7 x 7 x 30 tensor = **1470 outputs**

Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016

YOLO works across many natural images

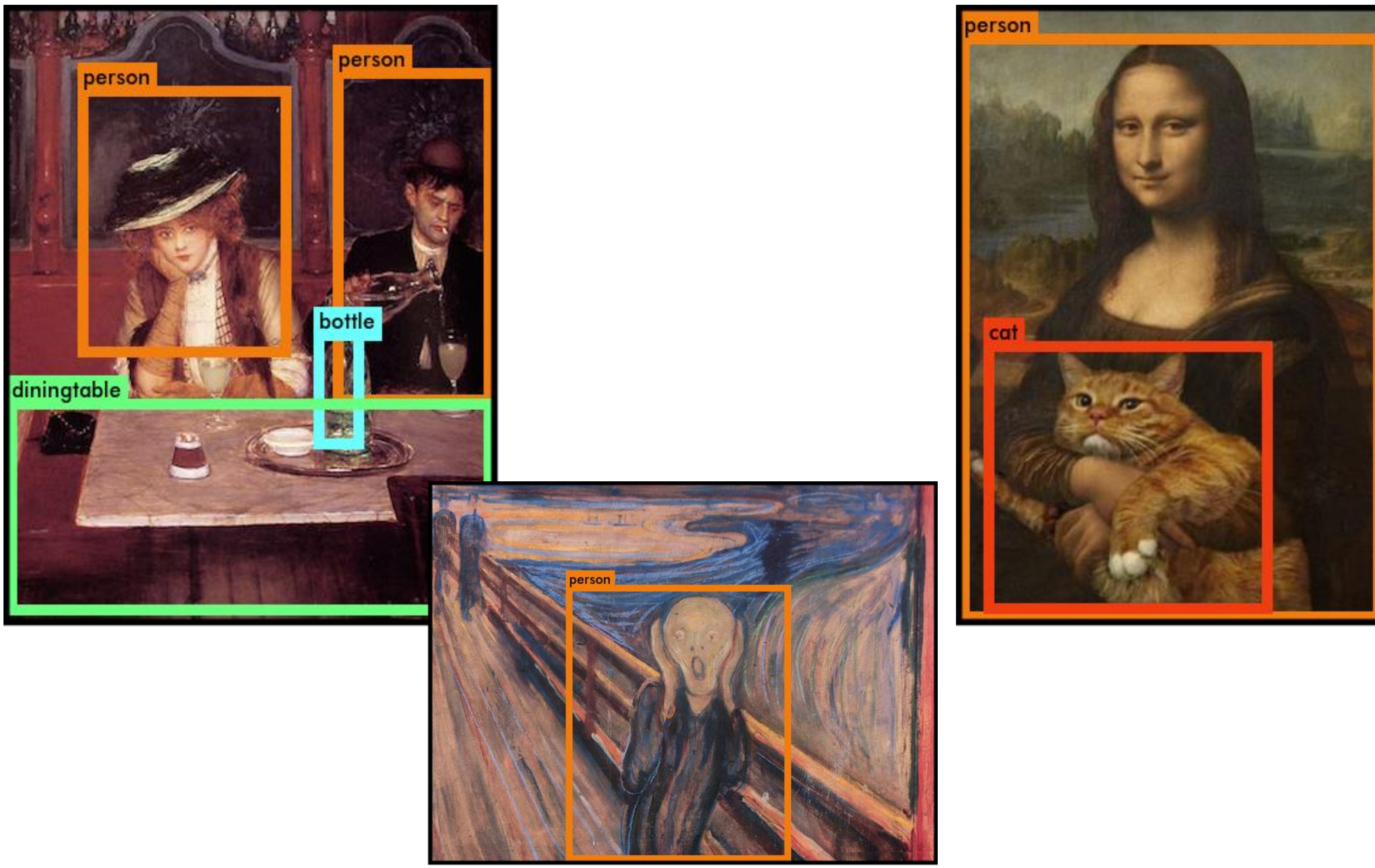






Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016

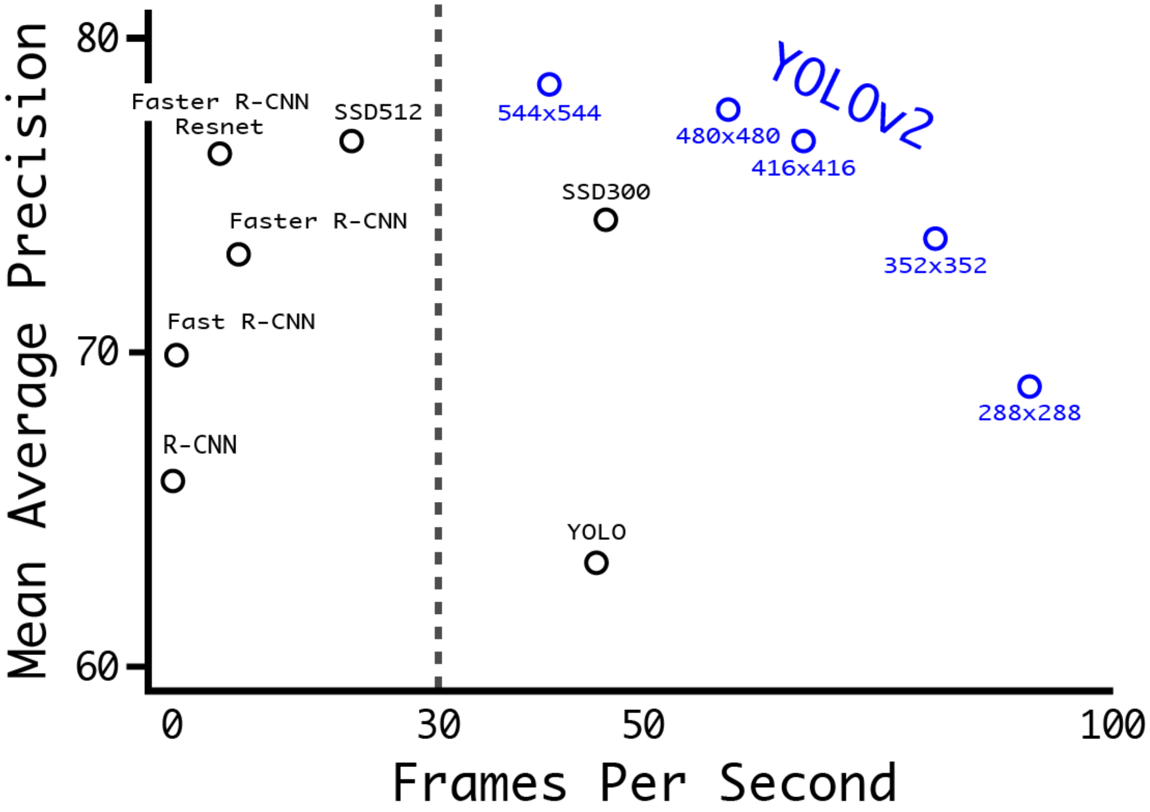
It also generalizes well to new domains



Redmon et al., "You Only Look Once: Unified, Real-Time Object Detection", CVPR 2016

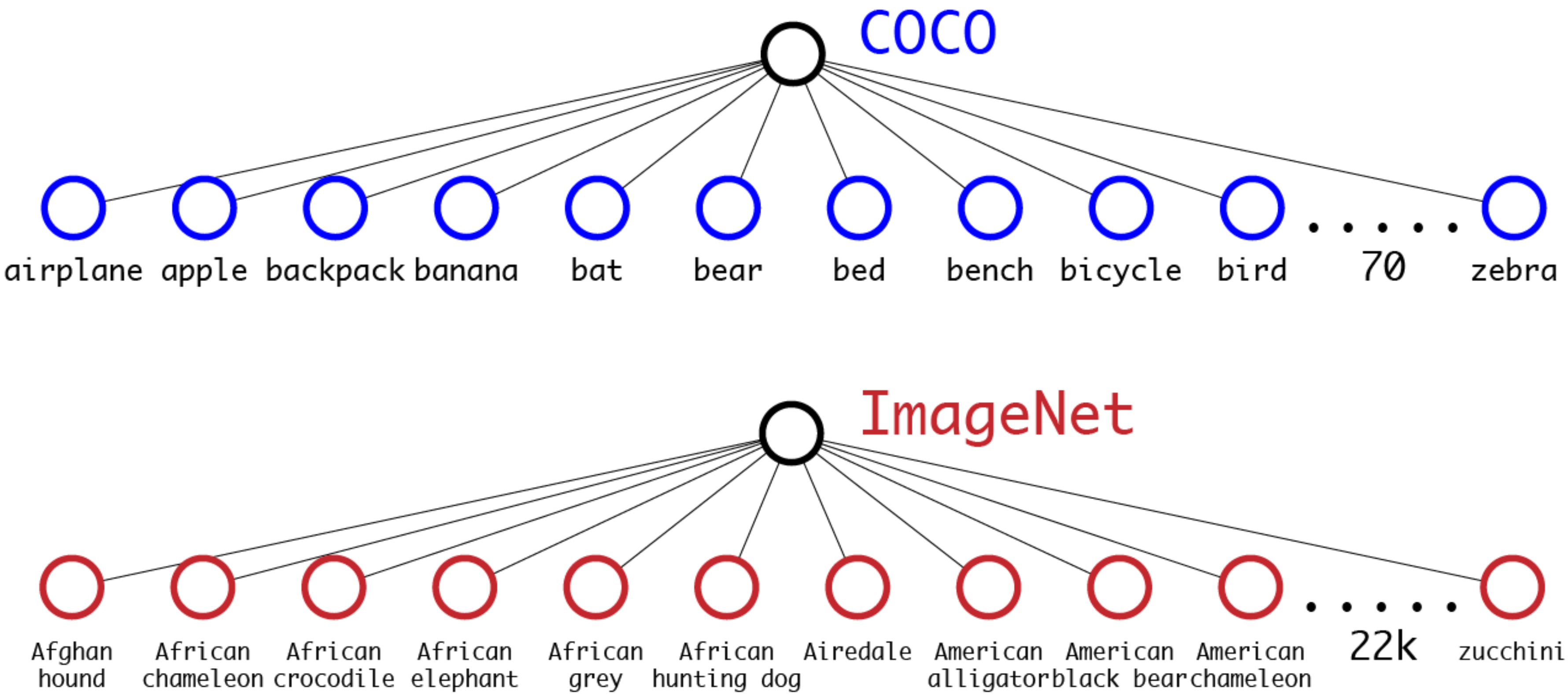
Redmon and Farhadi, "YOLO9000: Better, Faster, Stronger", CVPR 2017

YOLOv2: Fast, Accurate Detection



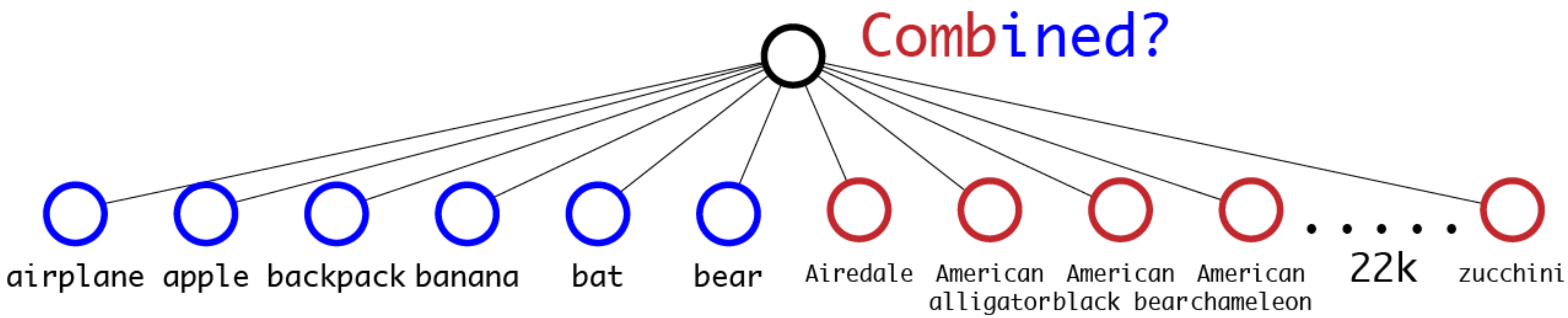
Redmon and Farhadi, "YOLO9000: Better, Faster, Stronger", CVPR 2017

Typically use softmax over all classes



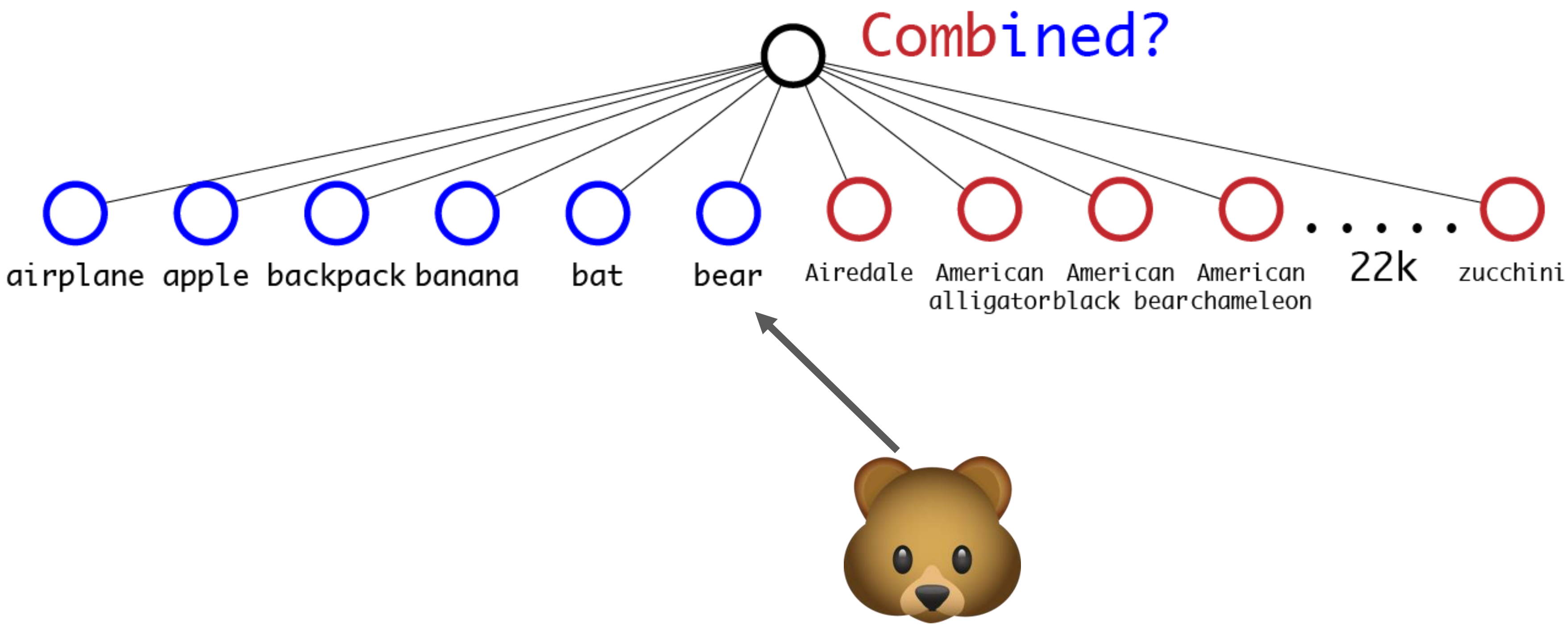
Redmon and Farhadi, "YOLO9000: Better, Faster, Stronger", CVPR 2017

Can’t just mash classes together...



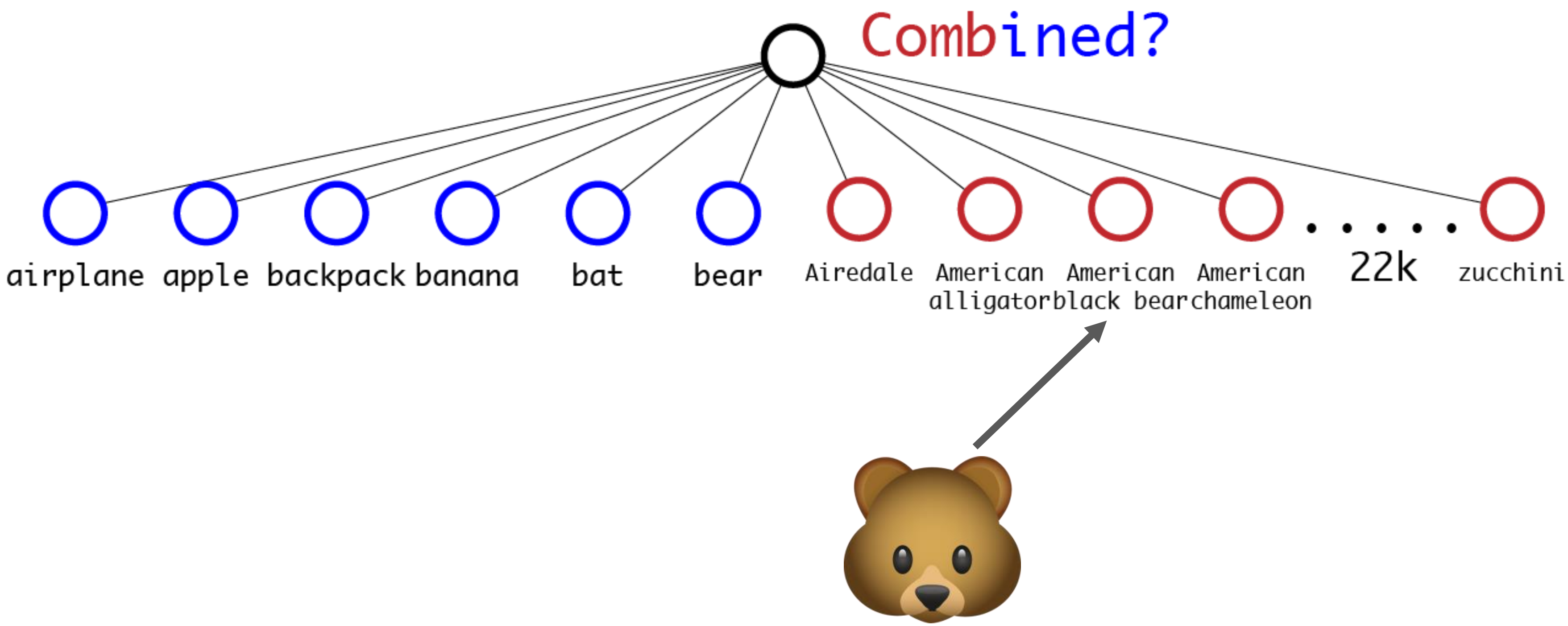
Redmon and Farhadi, “YOLO9000: Better, Faster, Stronger”, CVPR 2017

Can’t just mash classes together...



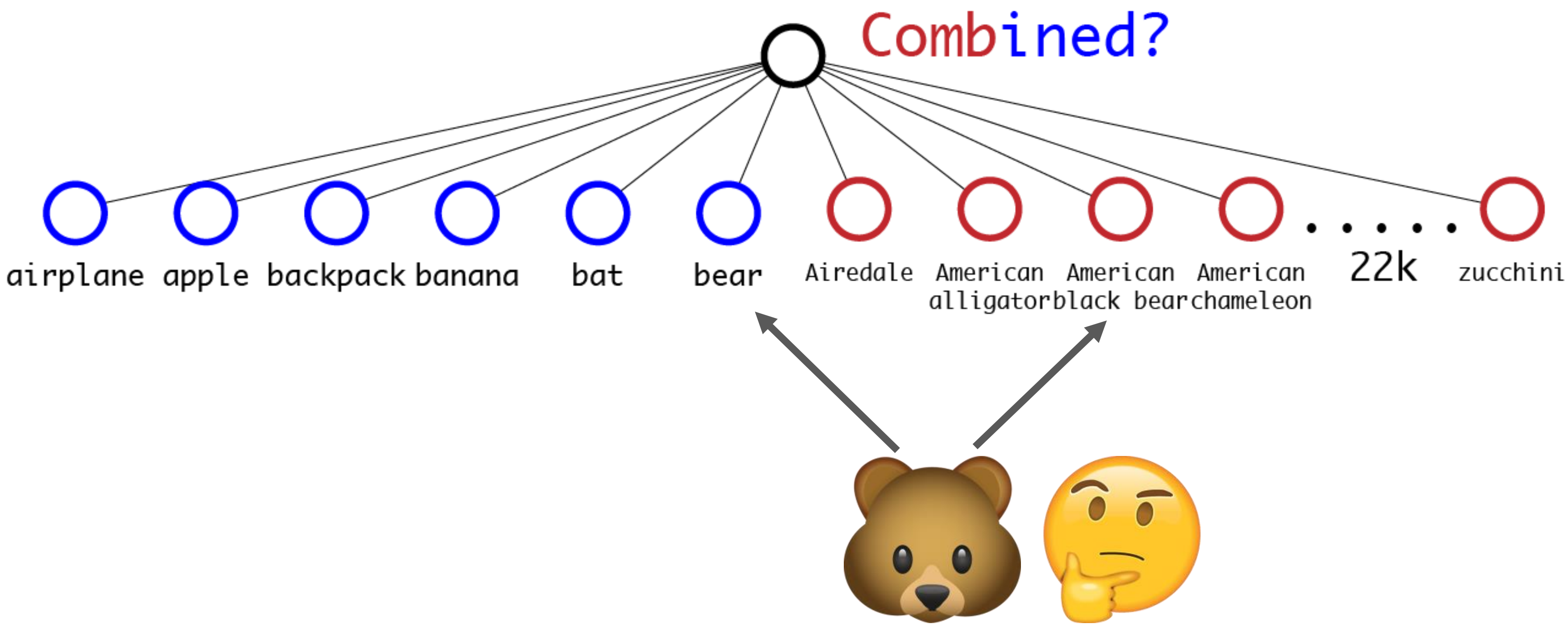
Redmon and Farhadi, “YOLO9000: Better, Faster, Stronger”, CVPR 2017

Can't just mash classes together...

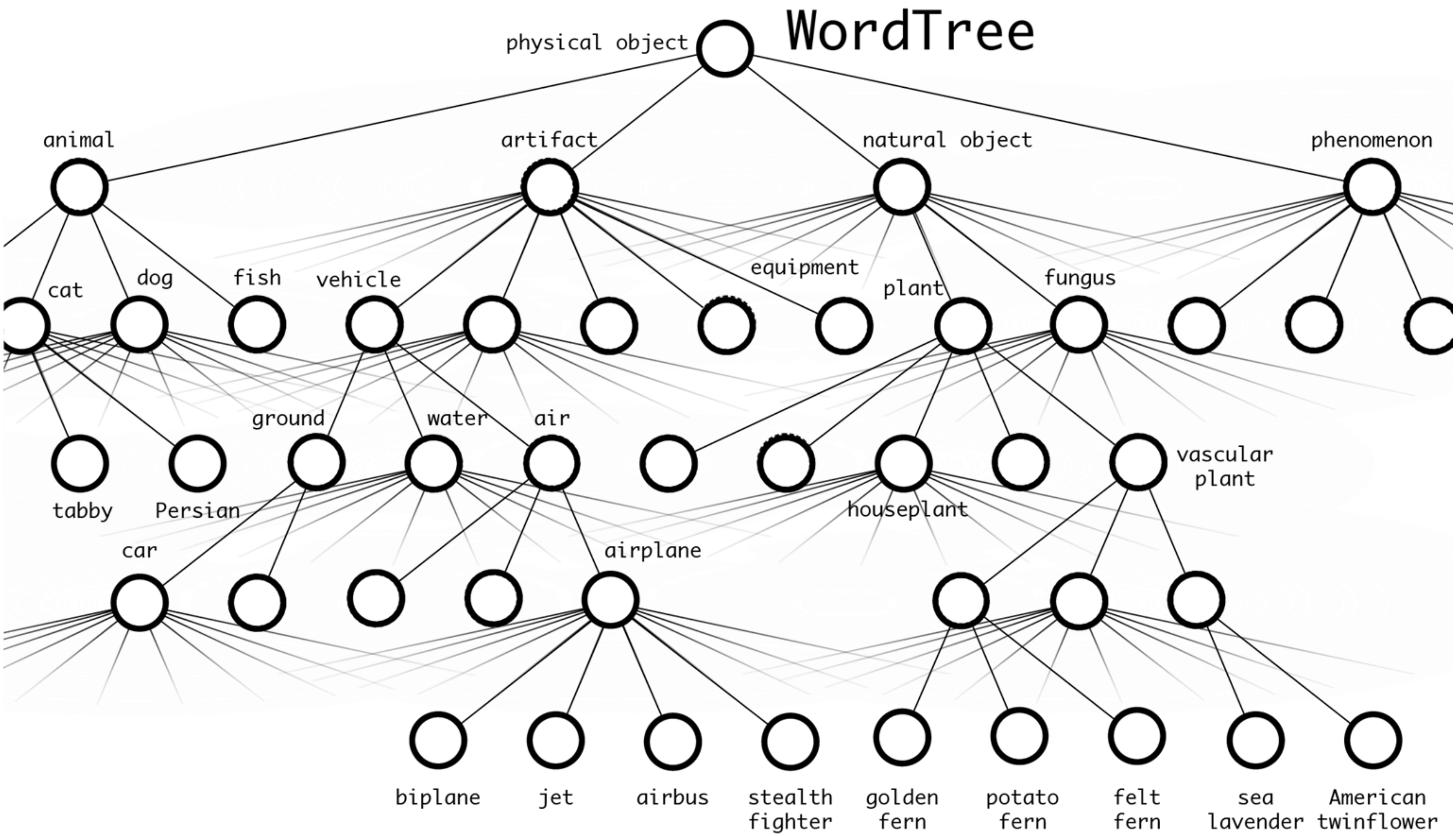


Redmon and Farhadi, "YOLO9000: Better, Faster, Stronger", CVPR 2017

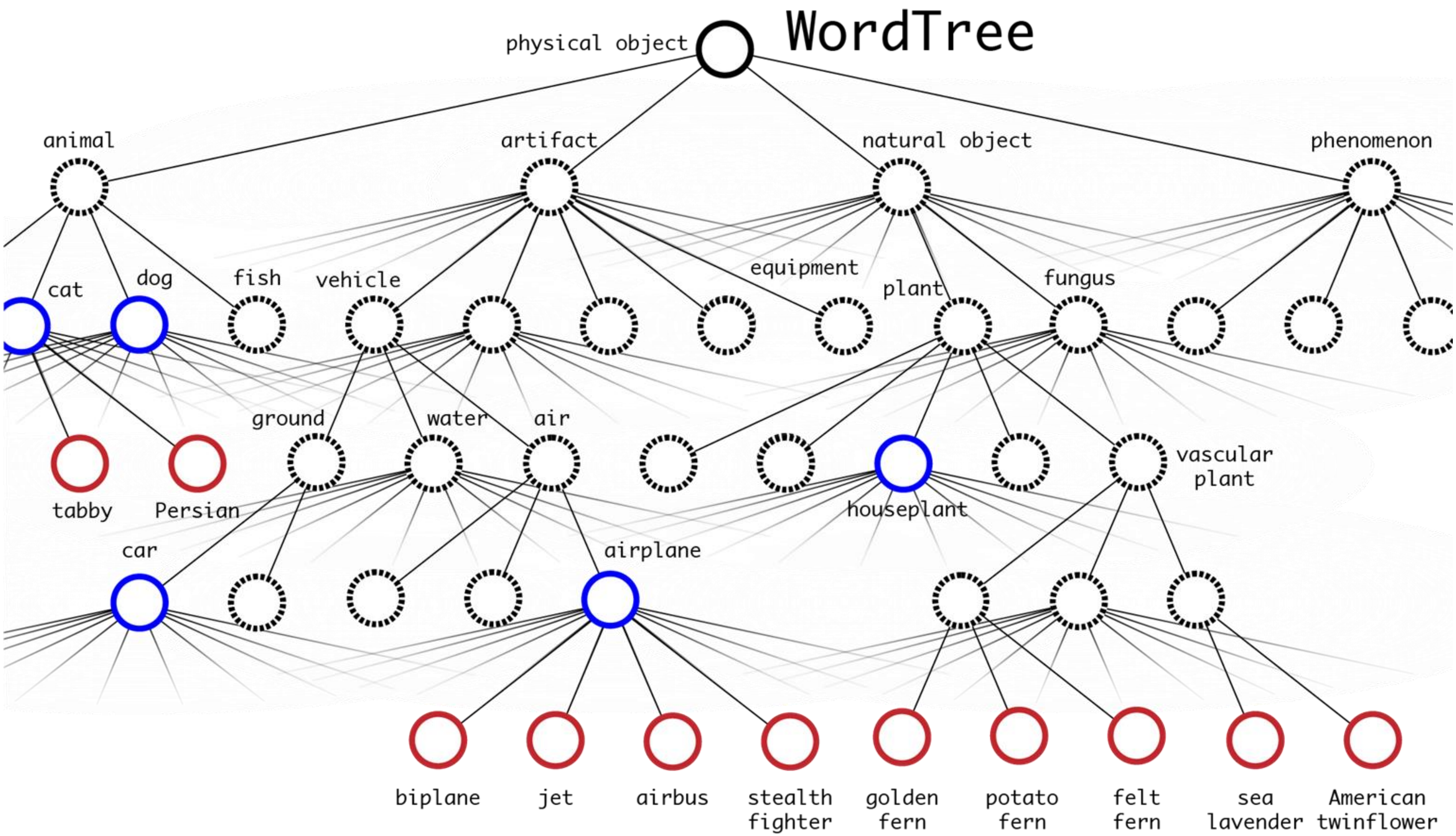
Can’t just mash classes together...



Redmon and Farhadi, “YOLO9000: Better, Faster, Stronger”, CVPR 2017

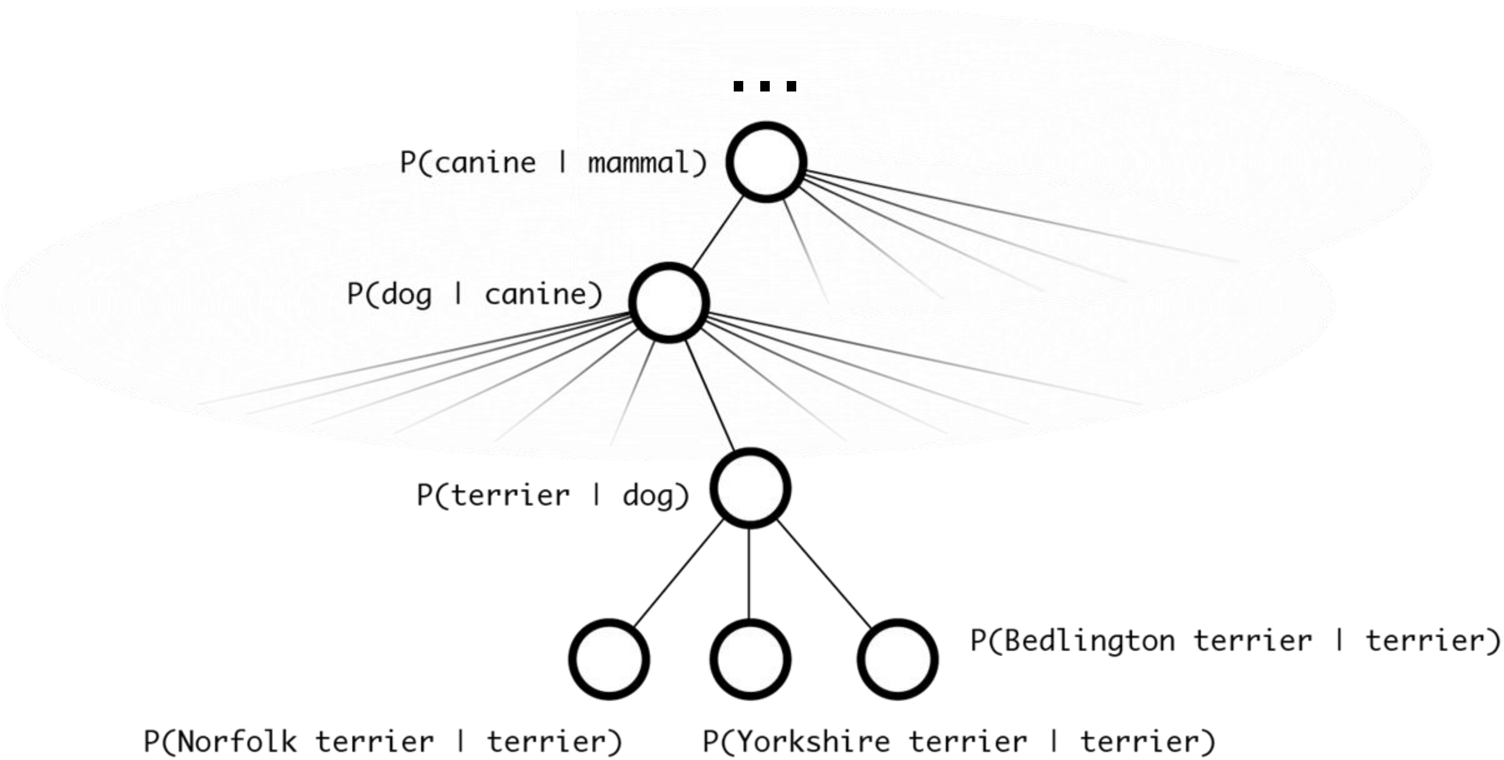


Redmon and Farhadi, “YOLO9000: Better, Faster, Stronger”, CVPR 2017



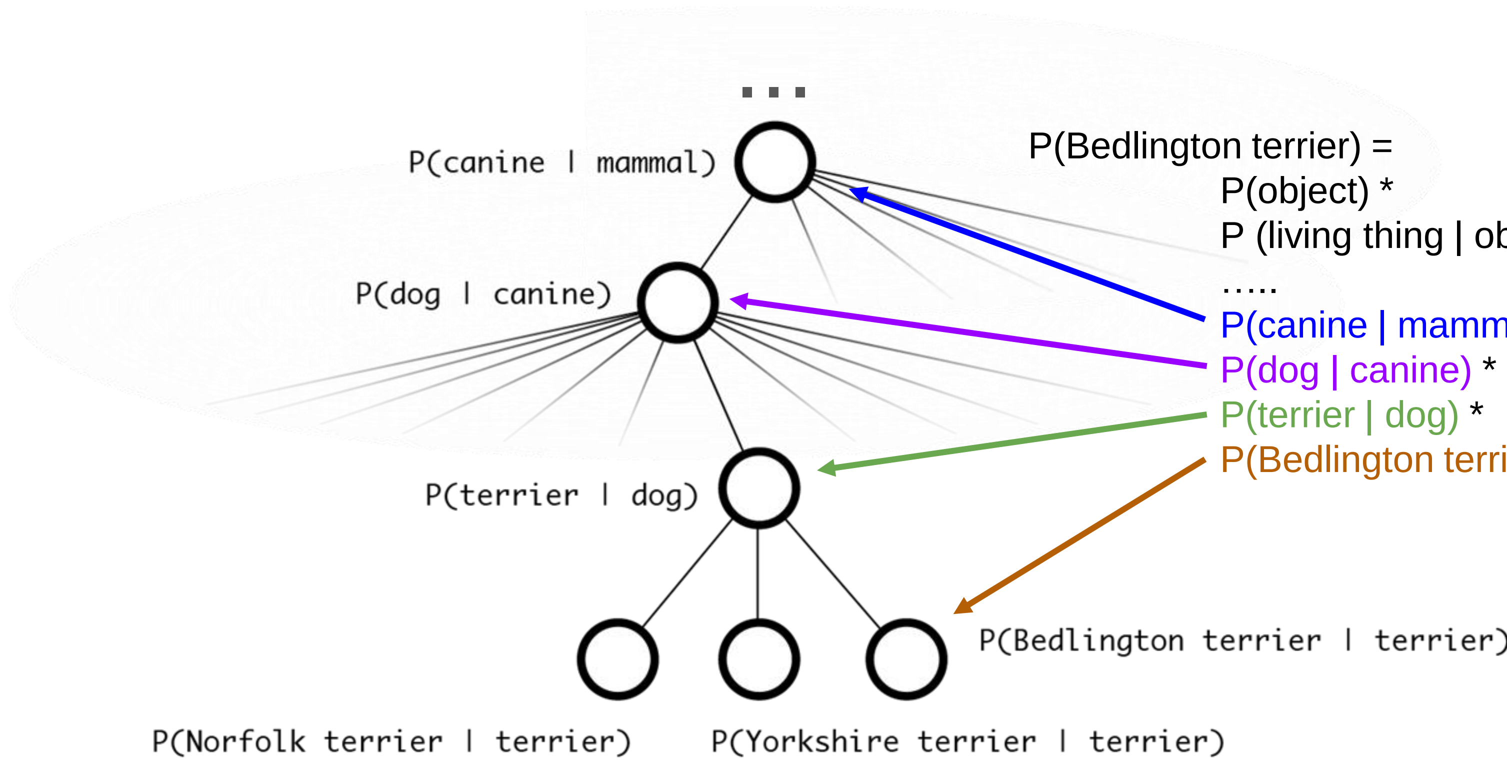
Redmon and Farhadi, "YOLO9000: Better, Faster, Stronger", CVPR 2017

Each node is a conditional probability



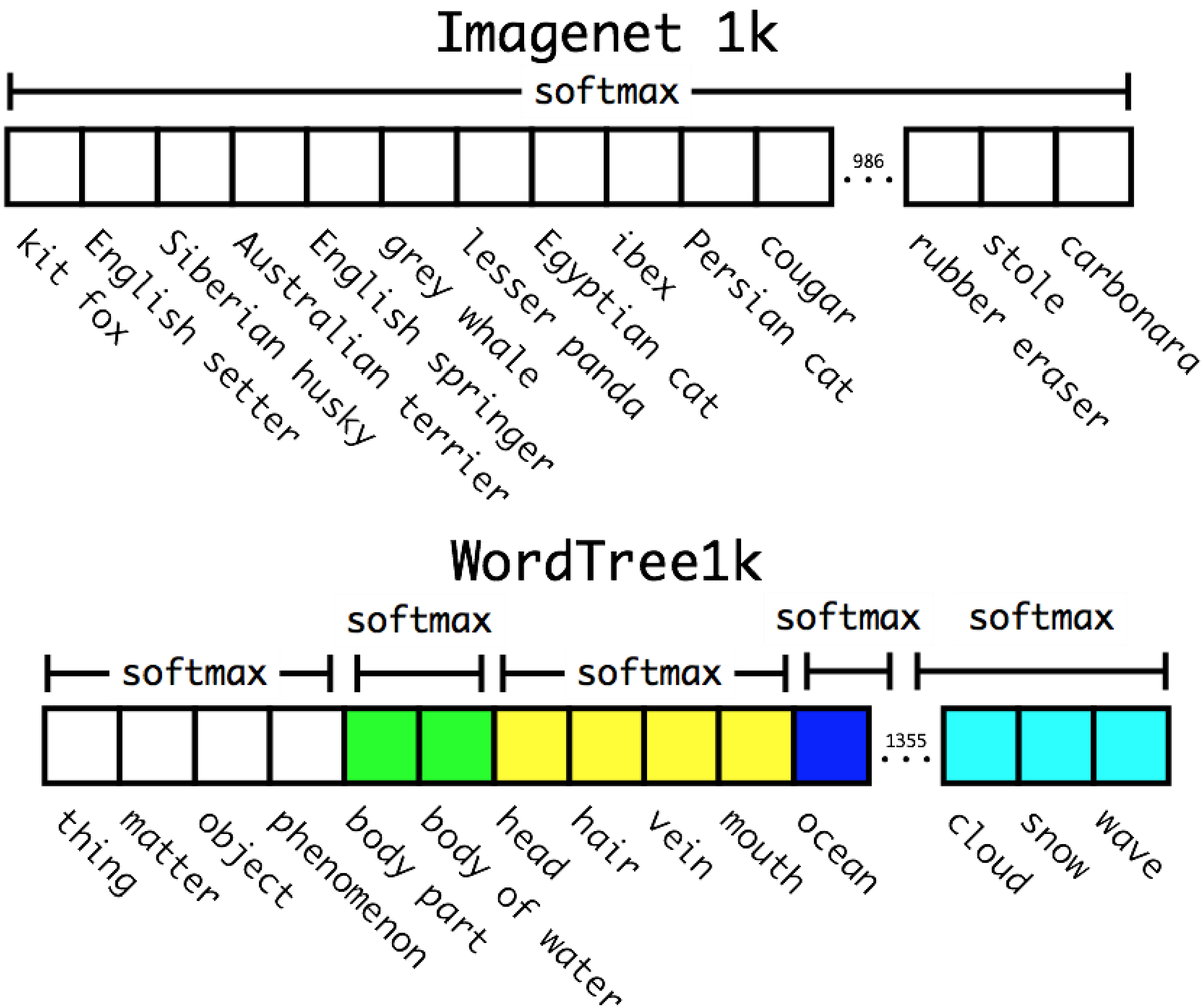
Redmon and Farhadi, "YOLO9000: Better, Faster, Stronger", CVPR 2017

Each node is a conditional probability

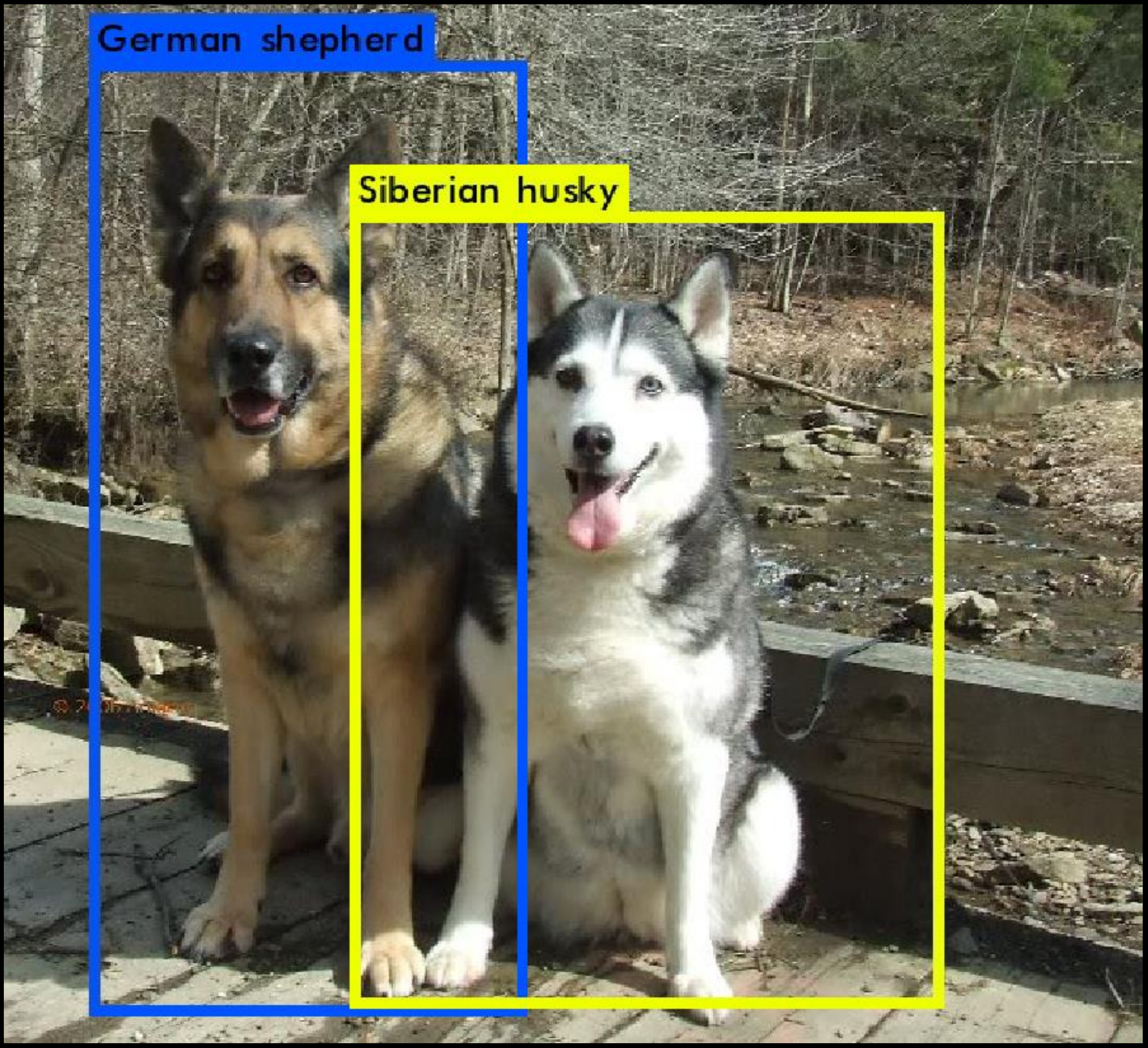


P(Bedlington terrier) =
P(object) *
P (living thing | object) *
…..
P(canine | mammal) *
P(dog | canine) *
P(terrier | dog) *
P(Bedlington terrier | terrier)

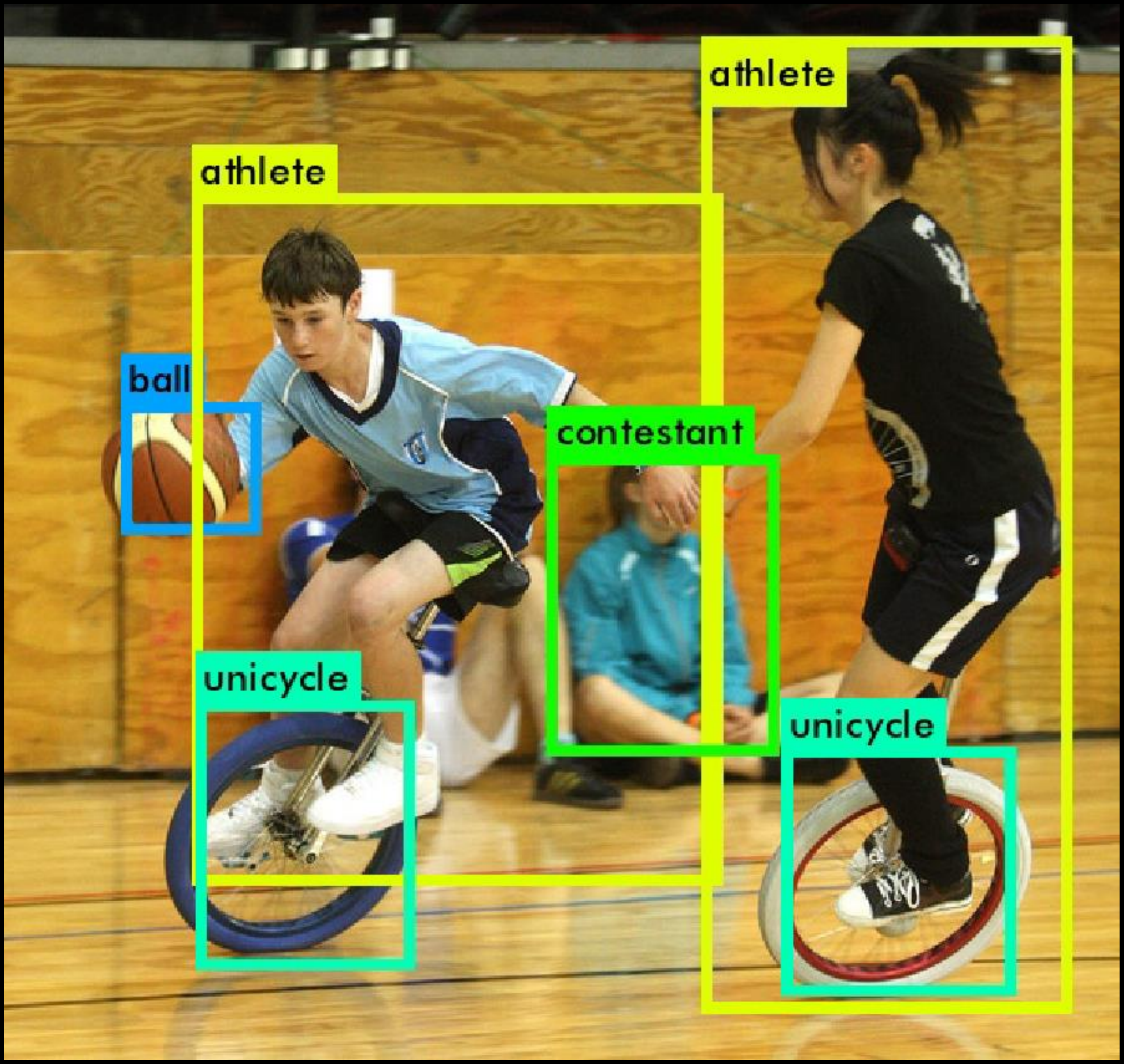
Redmon and Farhadi, "YOLO9000: Better, Faster, Stronger", CVPR 2017



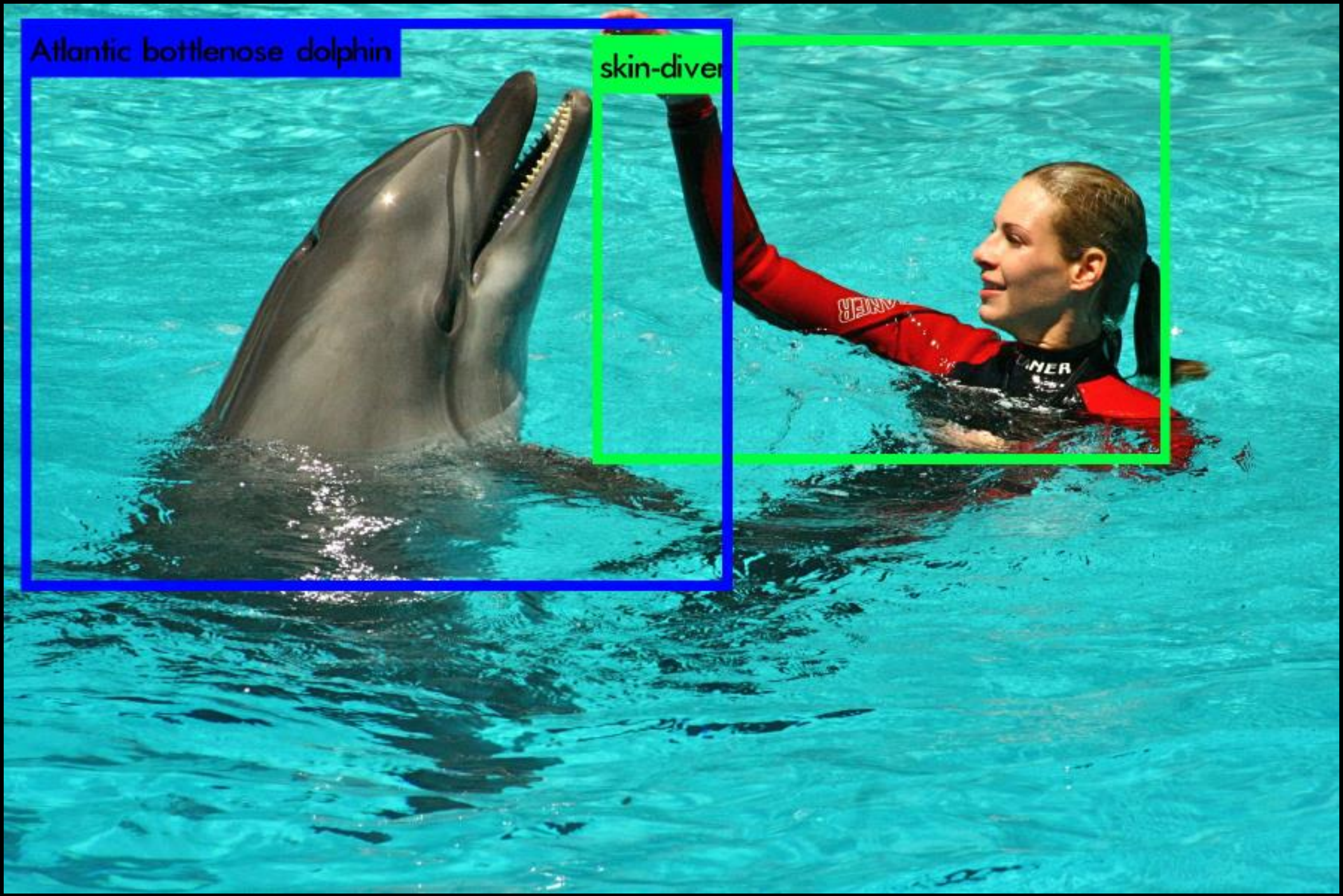
Redmon and Farhadi, “YOLO9000: Better, Faster, Stronger”, CVPR 2017



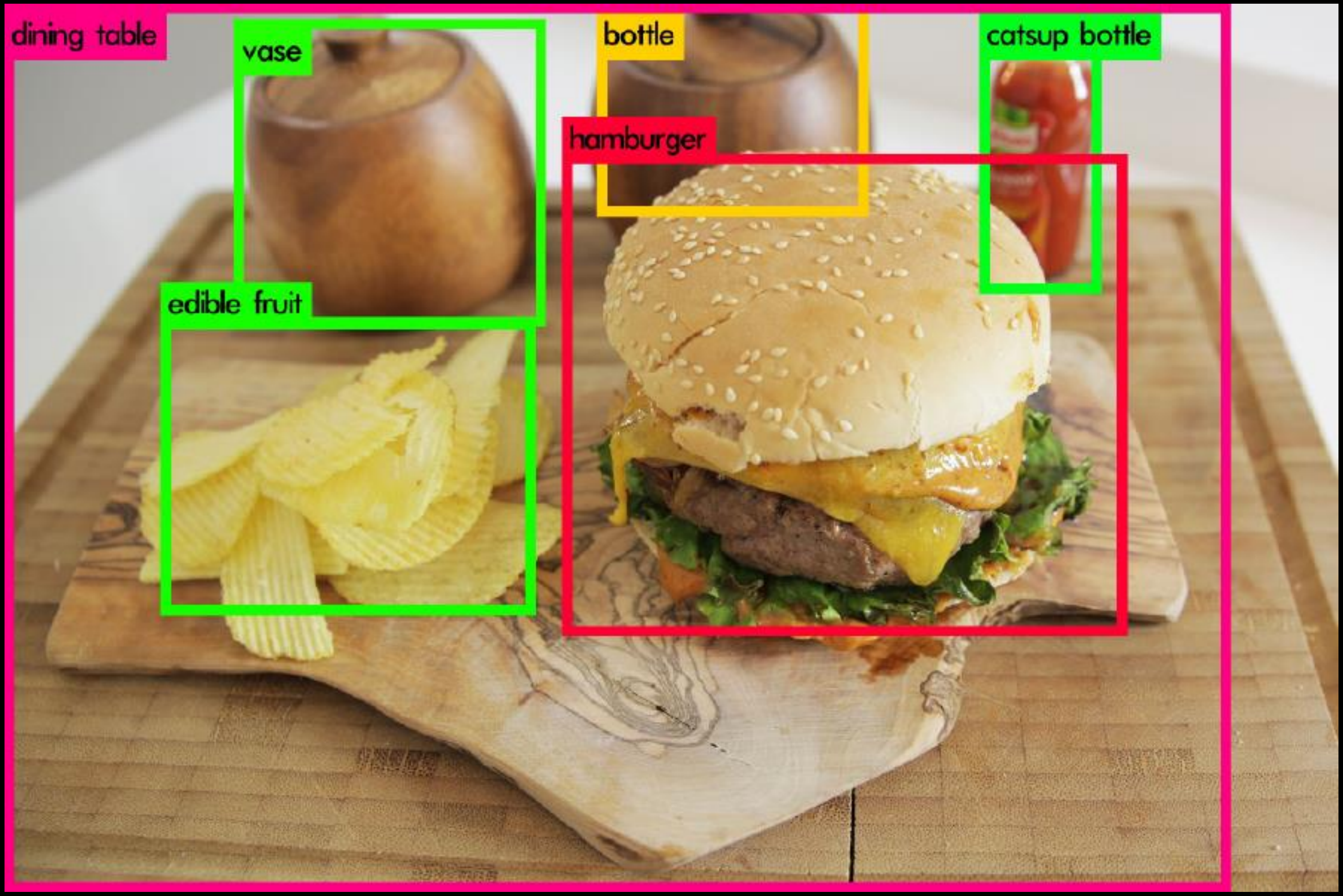
Redmon and Farhadi, "YOLO9000: Better, Faster, Stronger", CVPR 2017



Redmon and Farhadi, "YOLO9000: Better, Faster, Stronger", CVPR 2017



Redmon and Farhadi, "YOLO9000: Better, Faster, Stronger", CVPR 2017



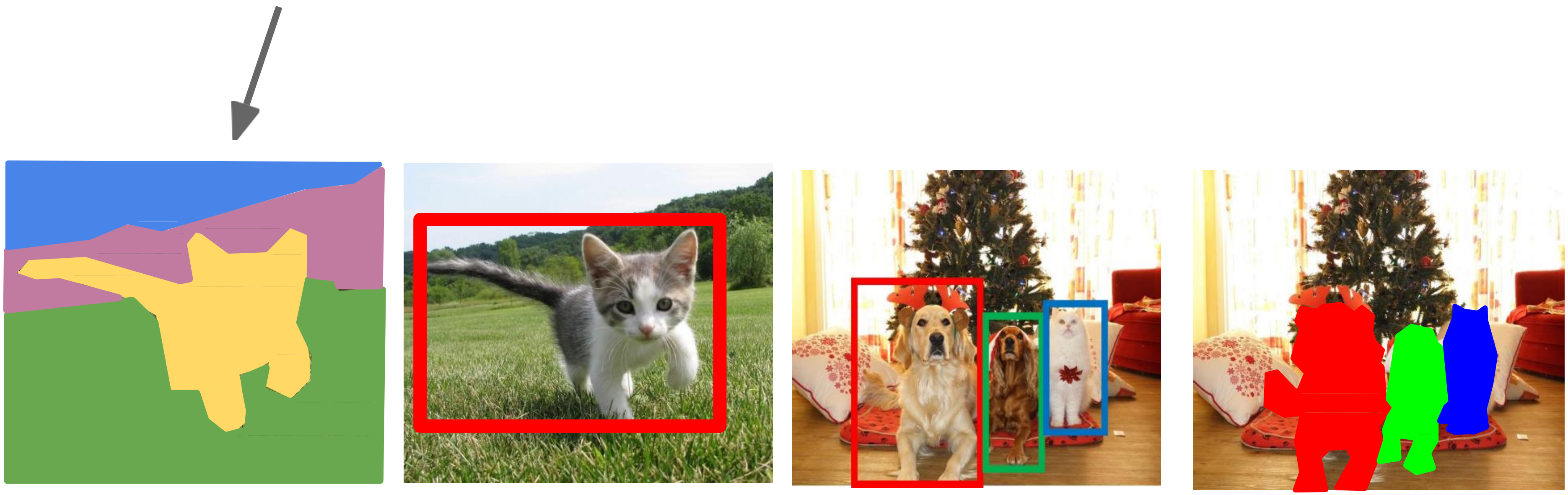
Redmon and Farhadi, “YOLO9000: Better, Faster, Stronger”, CVPR 2017

Redmon and Farhadi, "YOLO9000: Better, Faster, Stronger", CVPR 2017

Plan for this lecture

- Fully supervised detection
  - Pre-CNN: Deformable part models
  - Detection with region proposals: R-CNN, Fast/er R-CNN
  - Detection without region proposals: YOLO
  - Semantic and instance segmentation: FCN, Mask R-CNN
- Weak or out-of-domain supervision
  - Weakly supervised object detection
  - Domain adaptation

Semantic Segmentation



GRASS, CAT, TREE, SKY

CAT

DOG, DOG, CAT

DOG, DOG, CAT

No objects, just pixels

Single Object

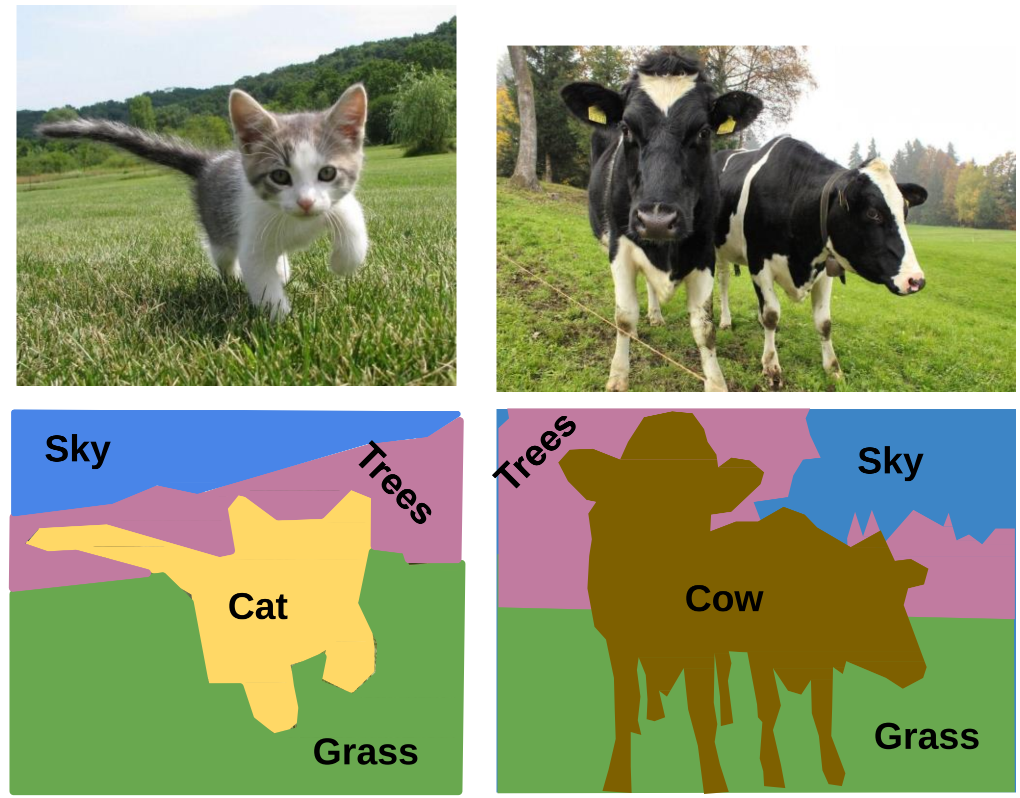
Multiple Object

Slide by: Justin Johnson

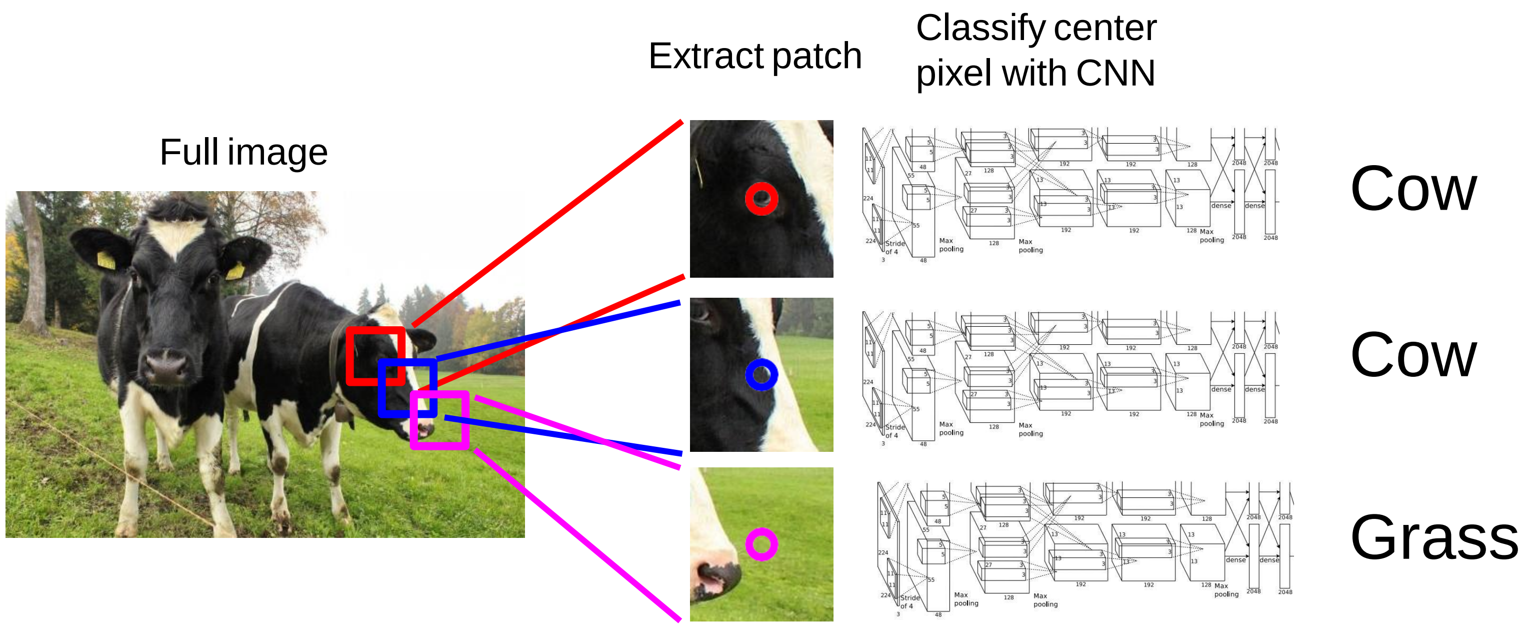
Semantic Segmentation

Label each pixel in the image with a category label

Don't differentiate instances, only care about pixels



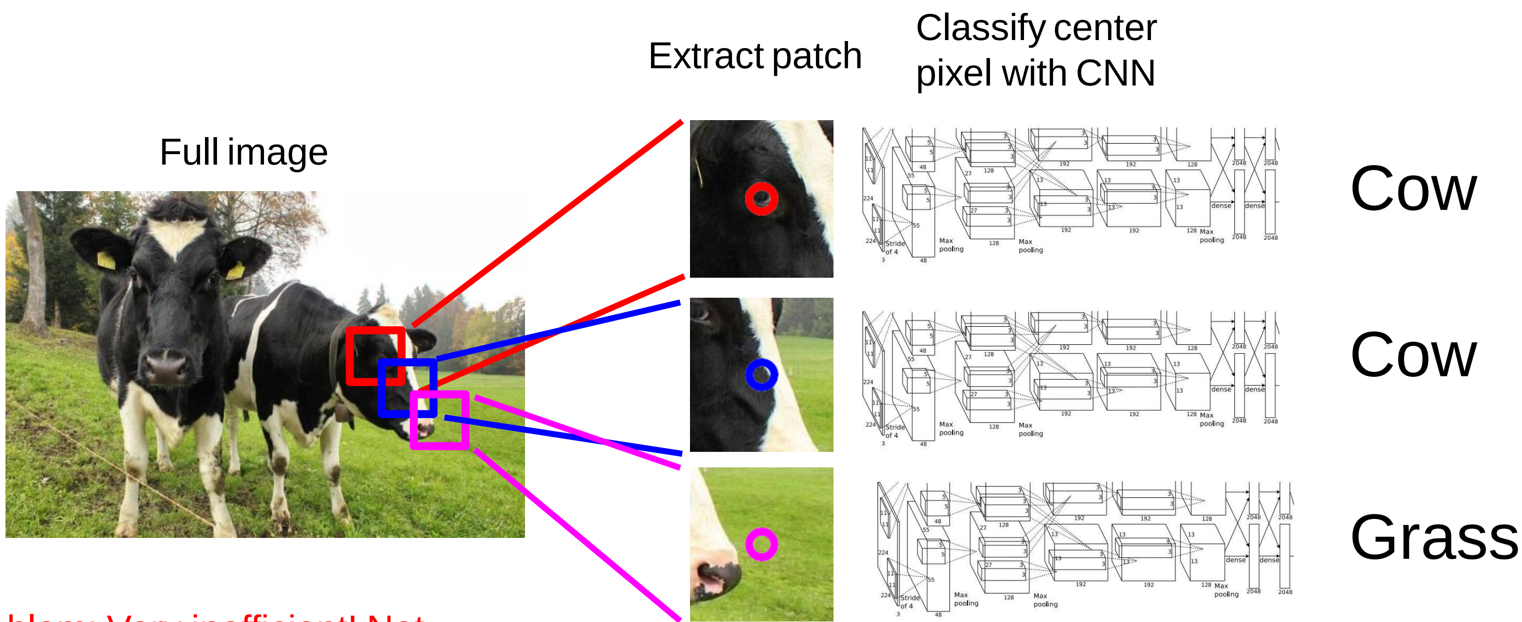
Semantic Segmentation Idea: Sliding Window



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

Slide by: Justin Johnson

Semantic Segmentation Idea: Sliding Window



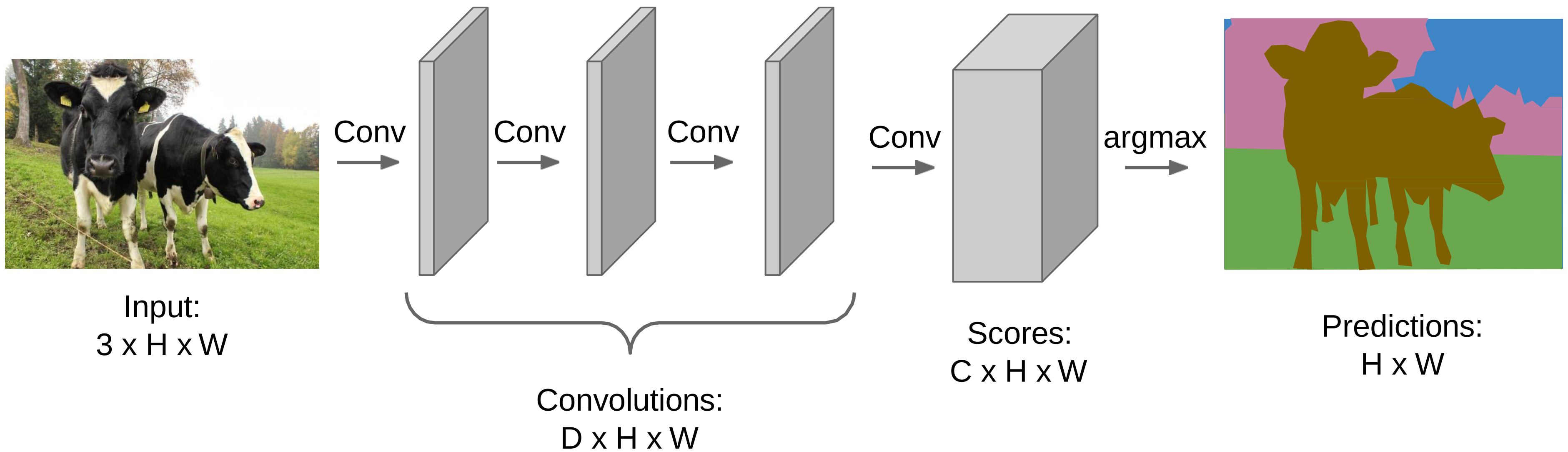
Problem: Very inefficient! Not reusing shared features between overlapping patches

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

Slide by: Justin Johnson

Semantic Segmentation Idea: Fully Convolutional

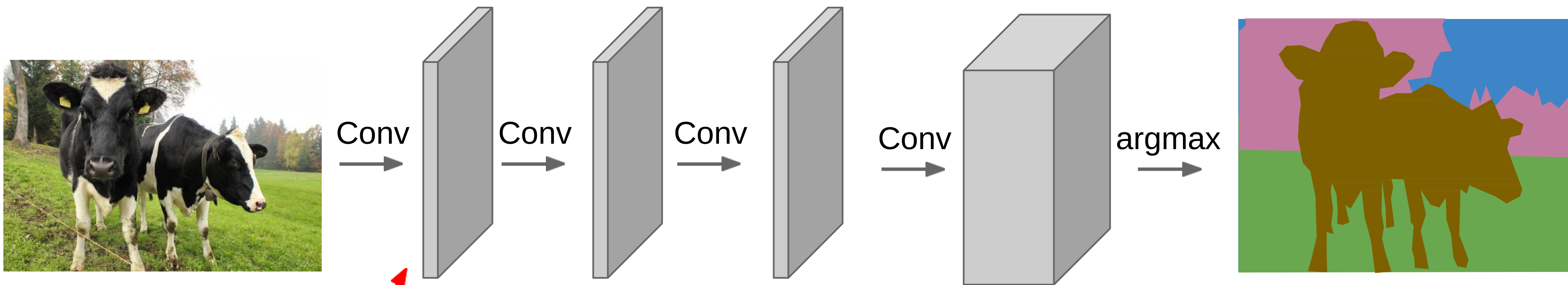
Design a network as a bunch of convolutional layers
to make predictions for pixels all at once!



Slide by: Justin Johnson

Semantic Segmentation Idea: Fully Convolutional

Design a network as a bunch of convolutional layers to make predictions for pixels all at once!



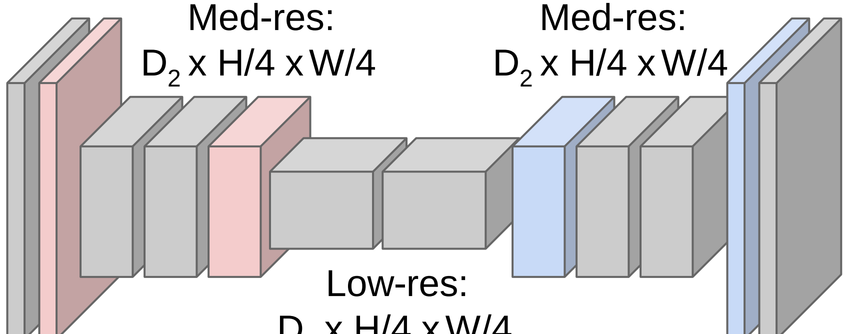
Problem: convolutions at original image resolution will be very expensive ...

Slide by: Justin Johnson

Semantic Segmentation Idea: Fully Convolutional

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Input:
3 x H x W



High-res:
D_1 x H/2 x W/2

Med-res:
D_2 x H/4 x W/4

Low-res:
D_3 x H/4 x W/4

Med-res:
D_2 x H/4 x W/4

High-res:
D_1 x H/2 x W/2

Predictions:
H x W

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Slide by: Justin Johnson

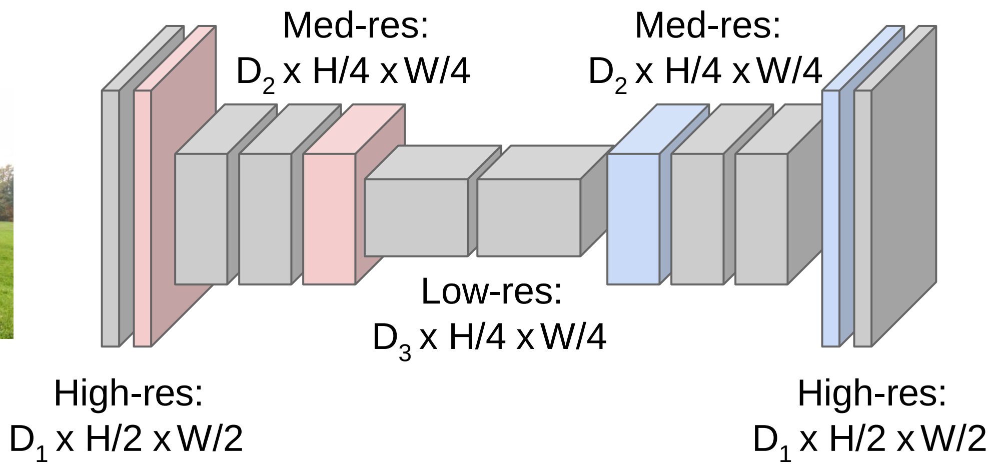
Semantic Segmentation Idea: Fully Convolutional

Downsampling: Pooling, strided convolution

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Upsampling: ???

Input: 3 x H x W



Predictions: H x W

Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Slide by: Justin Johnson

In-Network upsampling: “Unpooling”

Nearest Neighbor

1	2
3	4

Input: 2 x 2

1	1	2	2
1	1	2	2
3	3	4	4
3	3	4	4

Output: 4 x 4

“Bed of Nails”

1	2
3	4

Input: 2 x 2

1	0	2	0
0	0	0	0
3	0	4	0
0	0	0	0

Output: 4 x 4

Slide by: Justin Johnson

In-Network upsampling: “Max Unpooling”

Max Pooling
Remember which element was max!

1	2	6	3
3	5	2	1
1	2	2	1
7	3	4	8

Input: 4 x 4

5	6
7	8

Output: 2 x 2

…

Rest of the network

Max Unpooling
Use positions from pooling layer

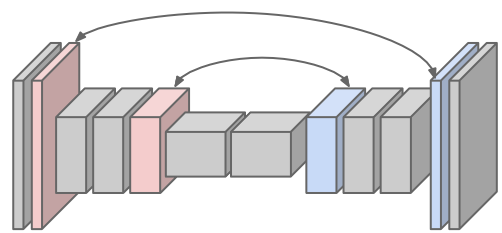
1	2
3	4

Input: 2 x 2

0	0	2	0
0	1	0	0
0	0	0	0
3	0	0	4

Output: 4 x 4

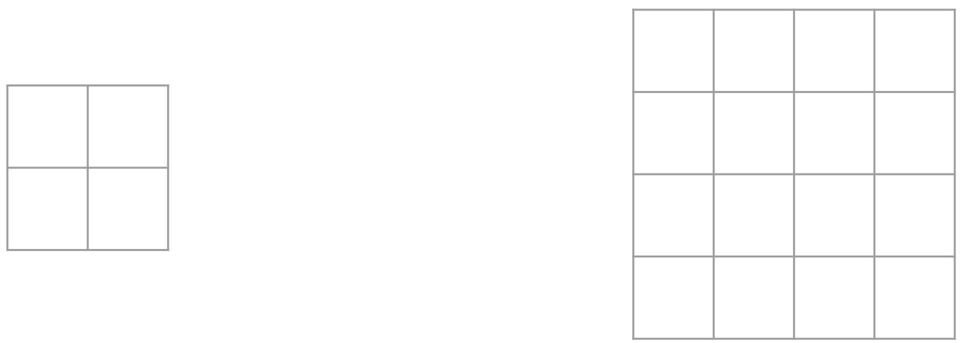
Corresponding pairs of downsampling and upsampling layers



Slide by: Justin Johnson

Learnable Upsampling: Transpose Convolution

3 x 3 **transpose** convolution, stride 2 pad 1



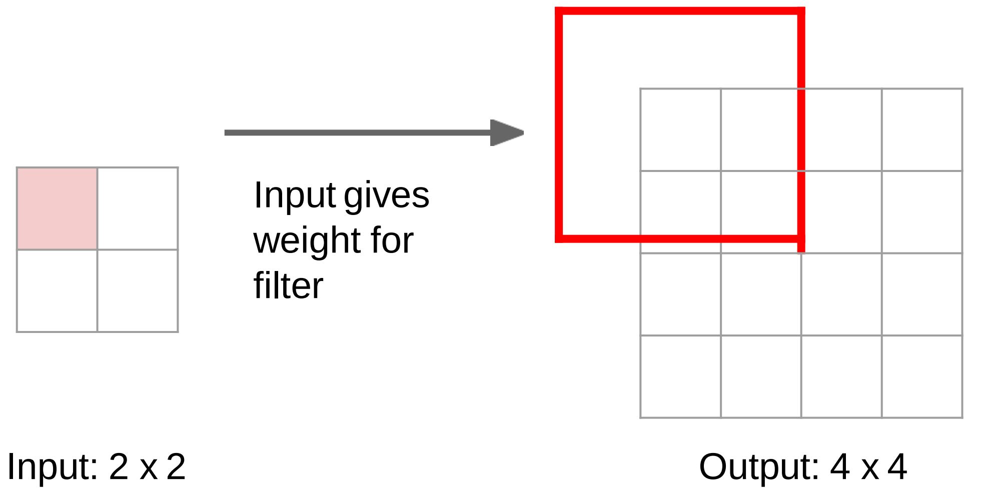
Input: 2 x 2

Output: 4 x 4

Slide by: Justin Johnson

Learnable Upsampling: Transpose Convolution

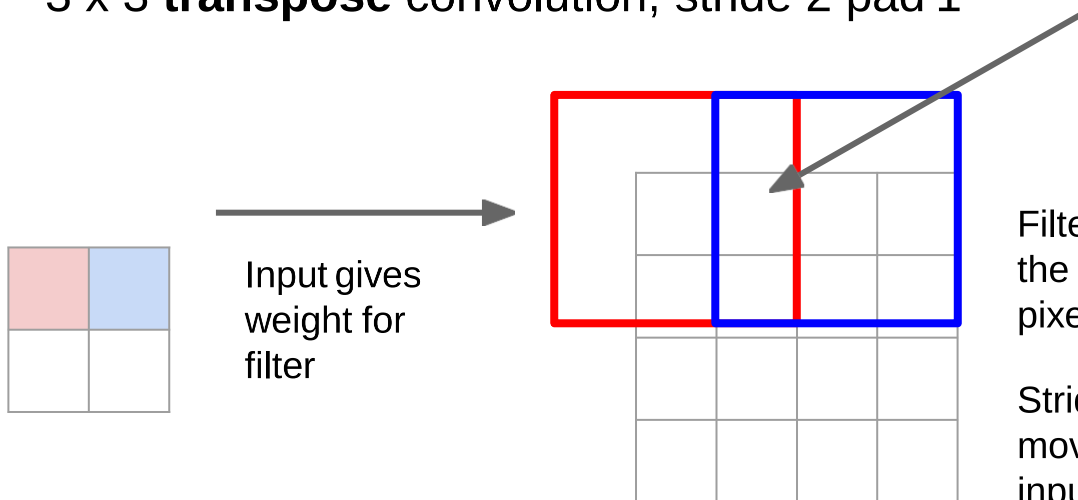
3 x 3 **transpose** convolution, stride 2 pad 1



Slide by: Justin Johnson

Learnable Upsampling: Transpose Convolution

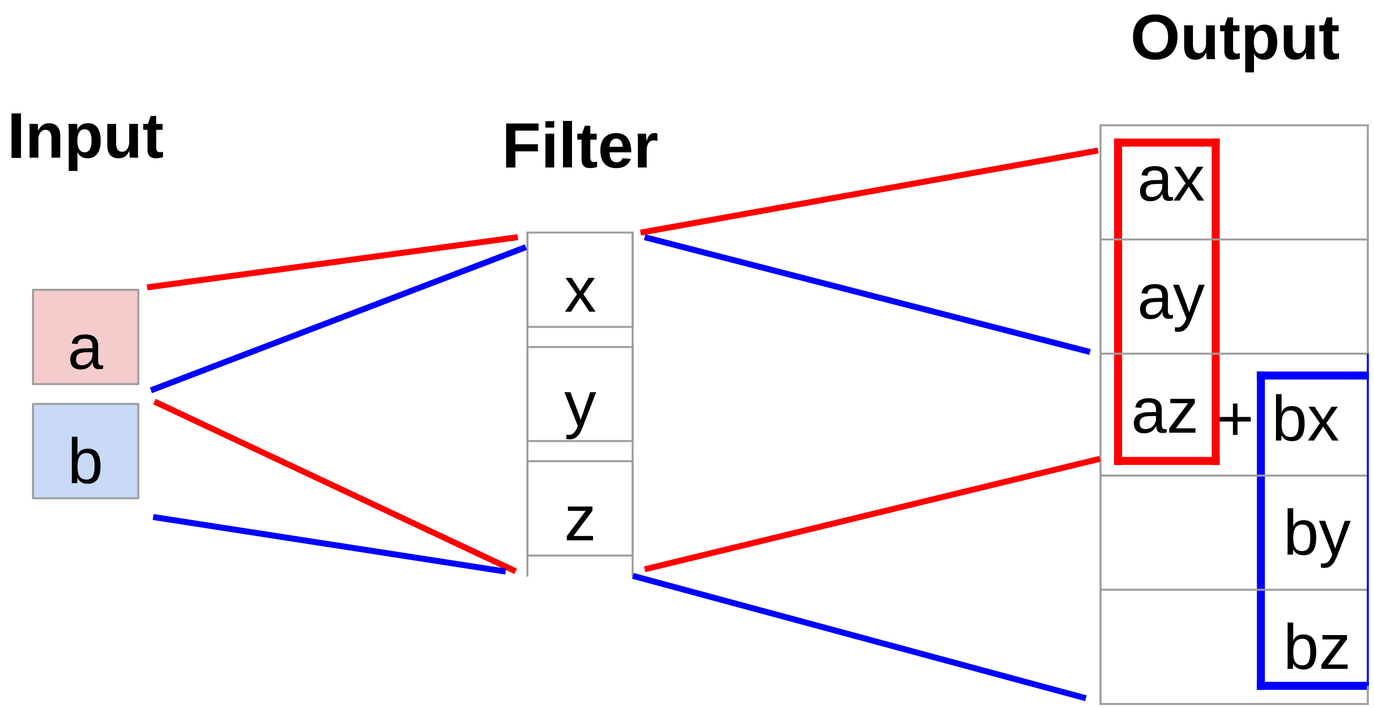
3 x 3 **transpose** convolution, stride 2 pad 1



Filter moves 2 pixels in the <u>output</u> for every one pixel in the <u>input</u>

Stride gives ratio between movement in output and input

Transpose Convolution: 1D Example



Output contains copies of the filter weighted by the input, summing at where at overlaps in the output

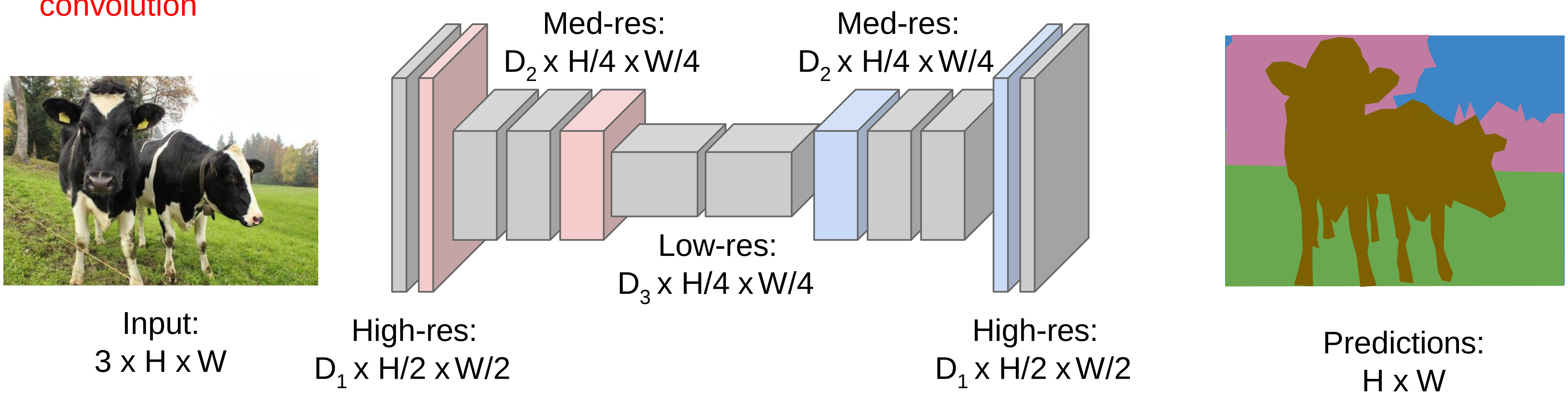
Adapted from Justin Johnson

Semantic Segmentation Idea: Fully Convolutional

Downsampling: Pooling, strided convolution

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

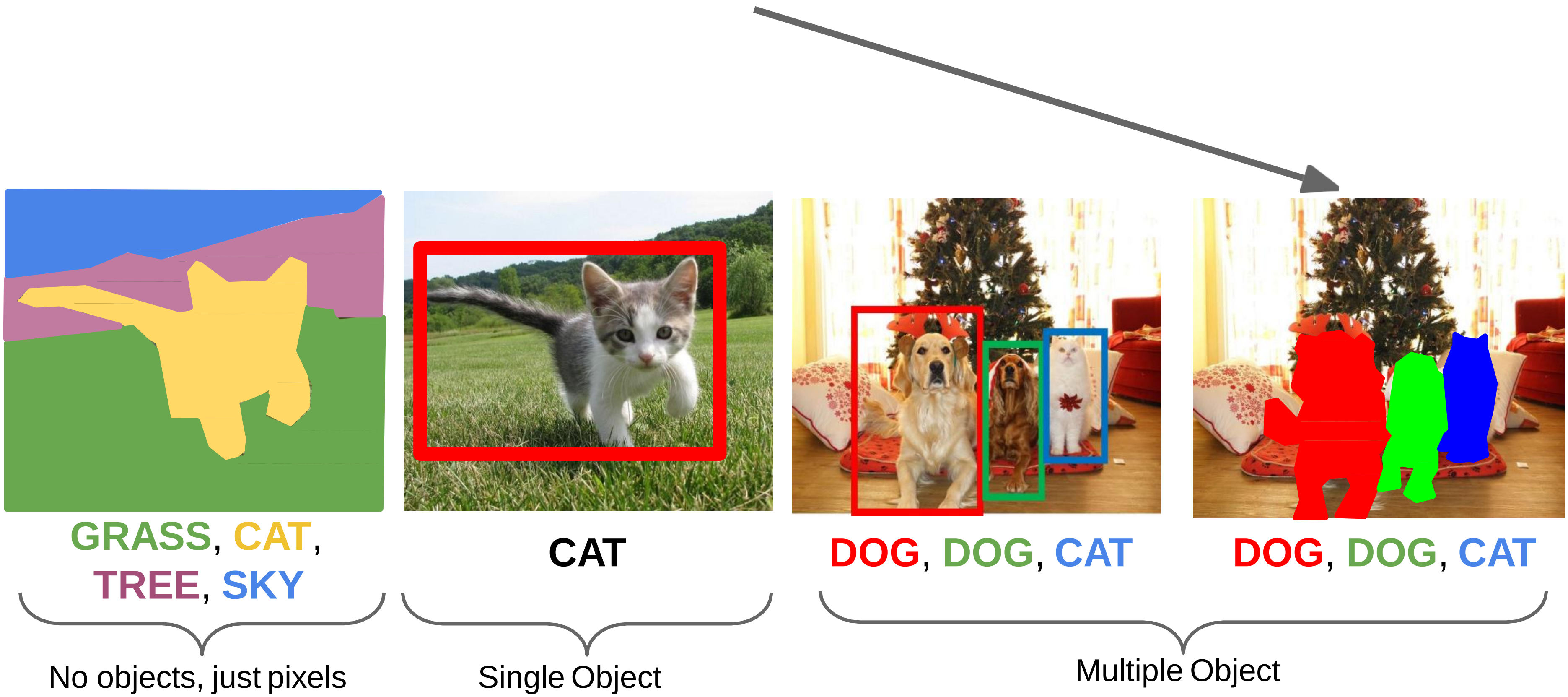
Upsampling: Unpooling or strided transpose convolution



Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

Slide by: Justin Johnson

Instance Segmentation



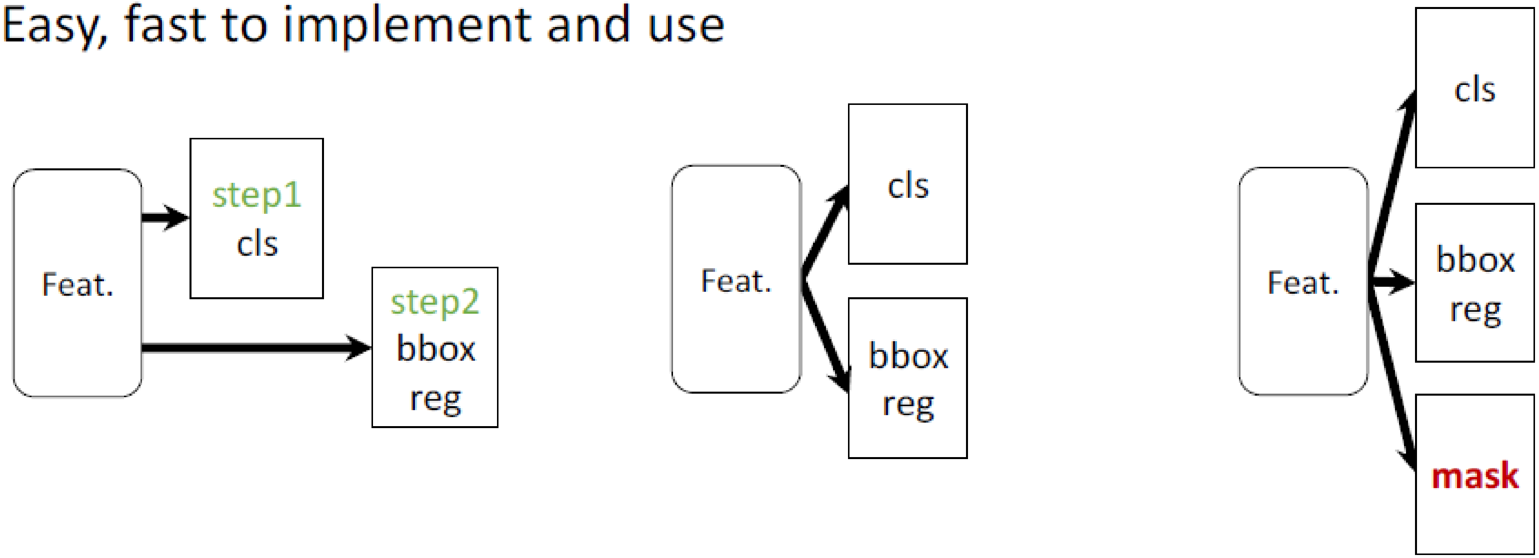
Slide by: Justin Johnson

Mask R-CNN

He et al, “Mask R-CNN”, ICCV 2017

What is Mask R-CNN: Parallel Heads

- Easy, fast to implement and use



(slow) R-CNN

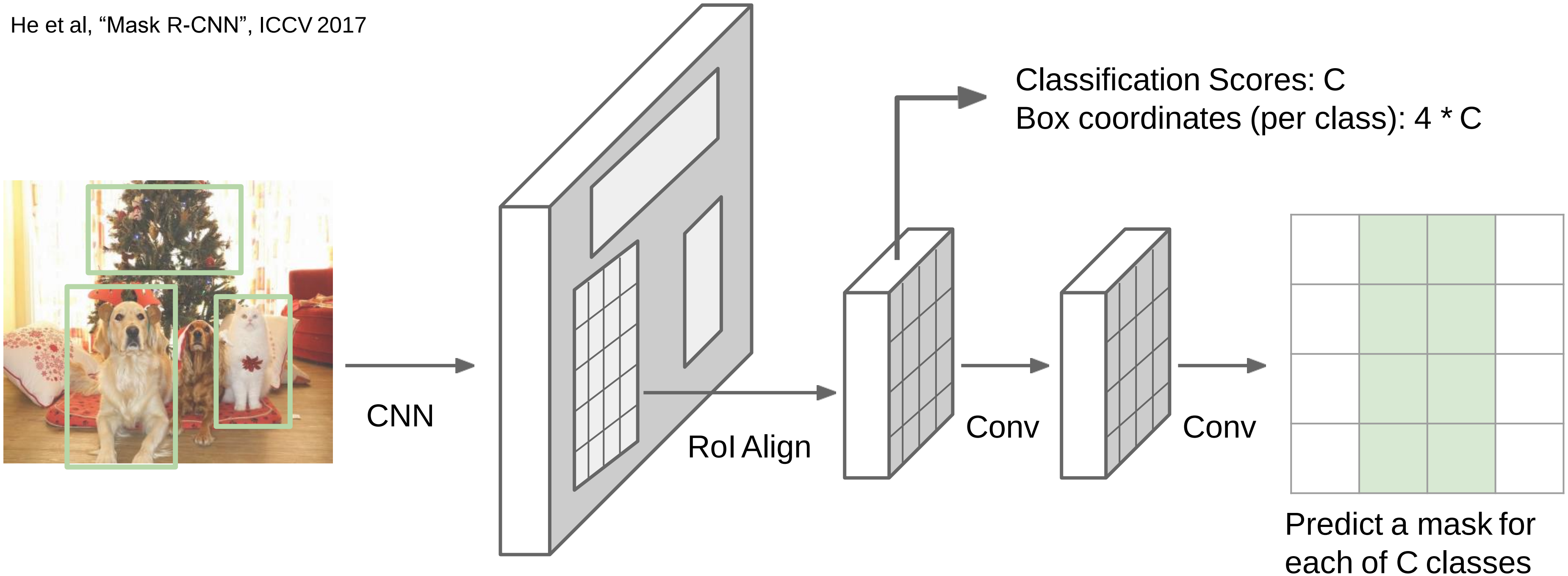
Fast/er R-CNN

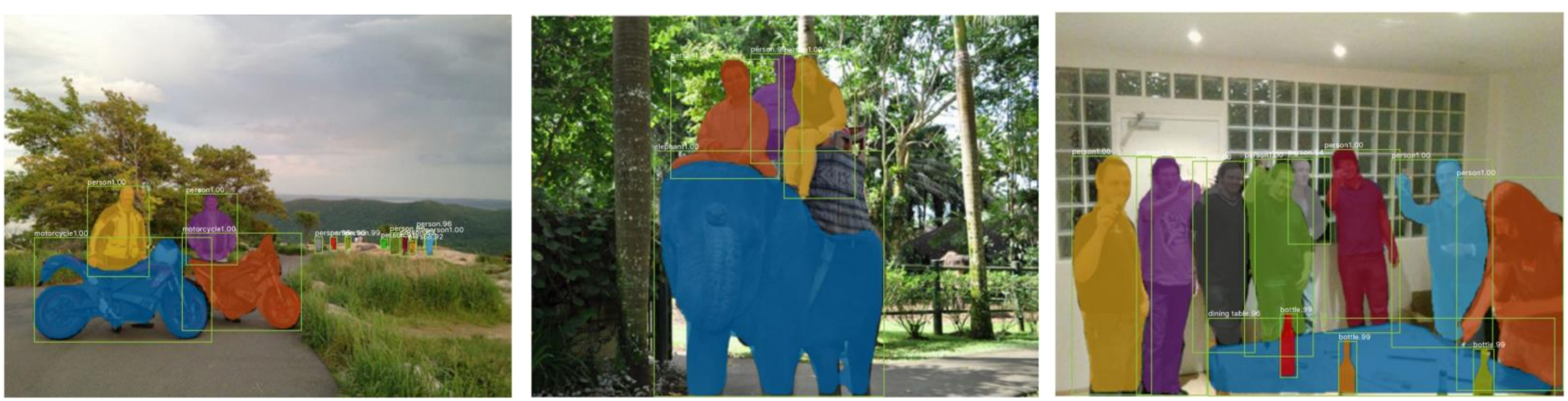
Mask R-CNN

Slide by: Kaiming He

Mask R-CNN

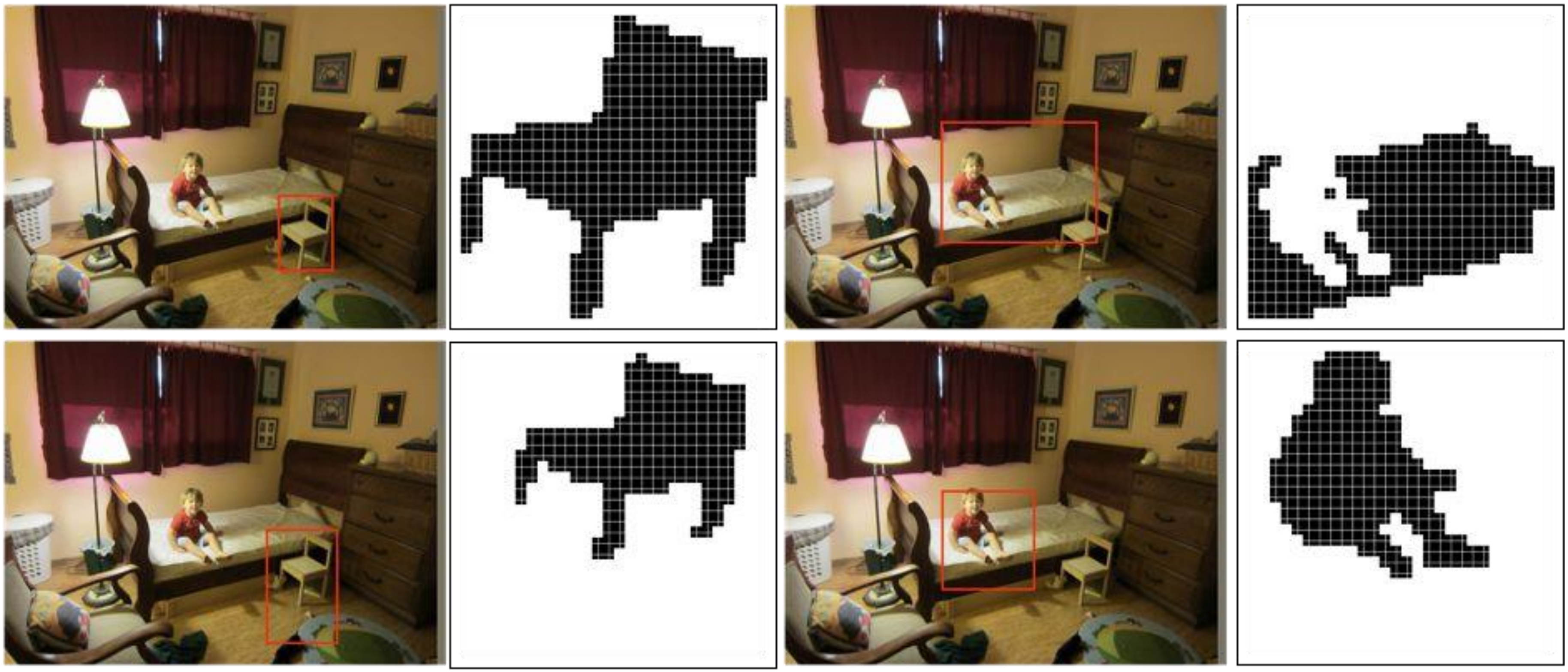
He et al, “Mask R-CNN”, ICCV 2017





Adapted from Justin Johnson

Mask R-CNN: Example Mask Training Targets



Slide by: Justin Johnson

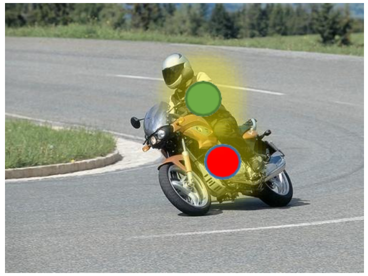
Plan for this lecture

- Fully supervised detection
  - Pre-CNN: Deformable part models
  - Detection with region proposals: R-CNN, Fast/er R-CNN
  - Detection without region proposals: YOLO
  - Semantic and instance segmentation: FCN, Mask R-CNN
- Weak or out-of-domain supervision
  - Weakly supervised object detection
  - Domain adaptation

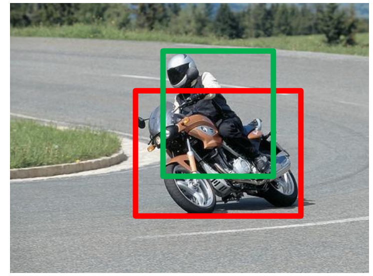
Weakly supervised object detection

Manual supervision for object recognition

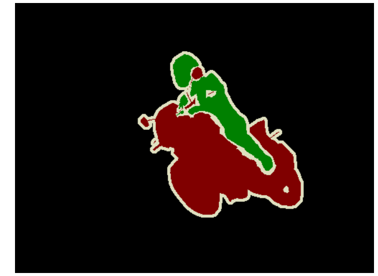
{motorbike,person}



{motorbike (point), person (point)}



{motorbike (b-box), person (b-box)}



{motorbike (pixel labels), person (pixel labels)}

1 sec per class

2.4 sec per instance

10 sec per instance

78 sec per instance

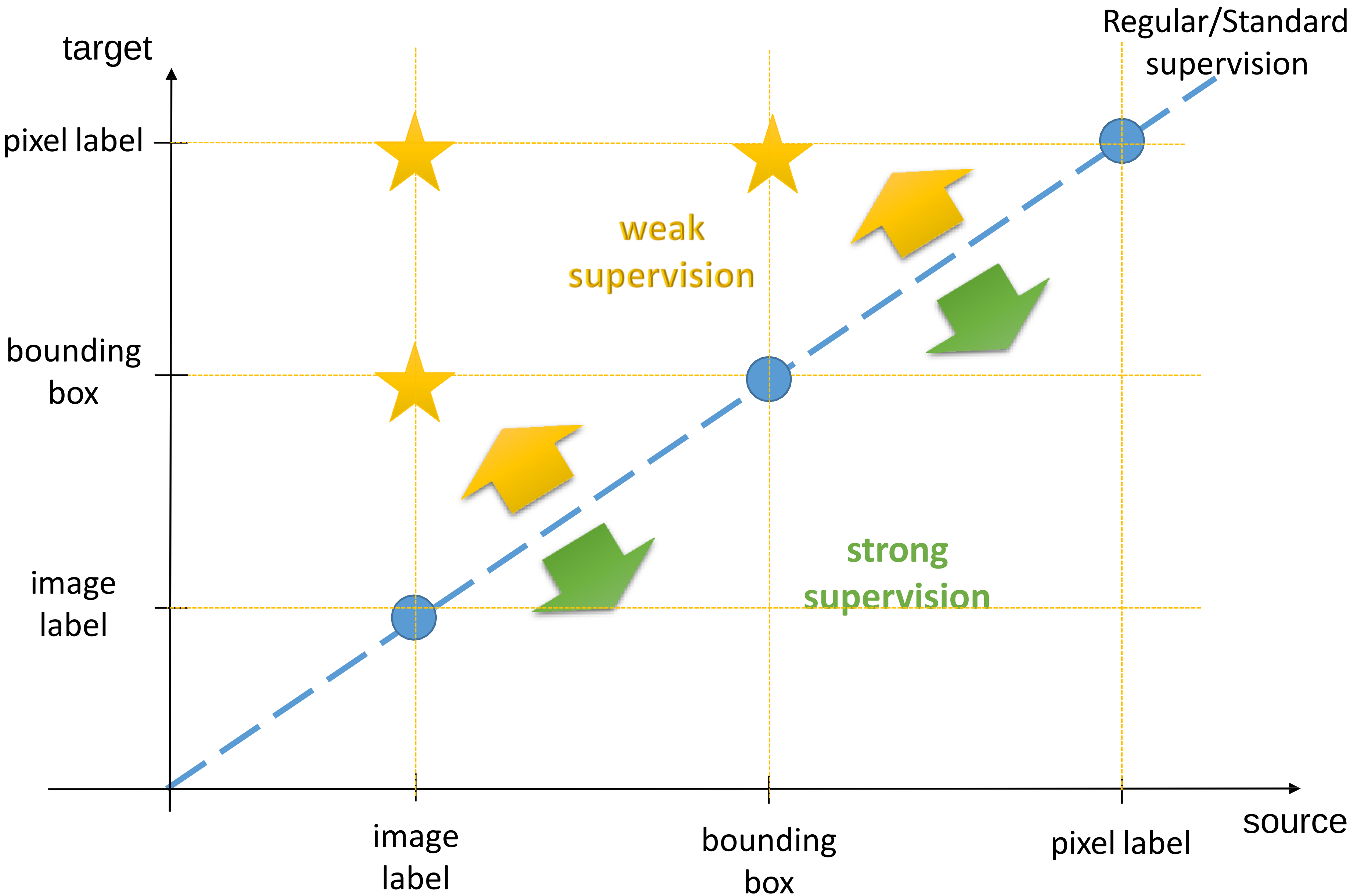
annotation time

Berman et al., What's the Point: Semantic Segmentation with Point Supervision, ECCV 16

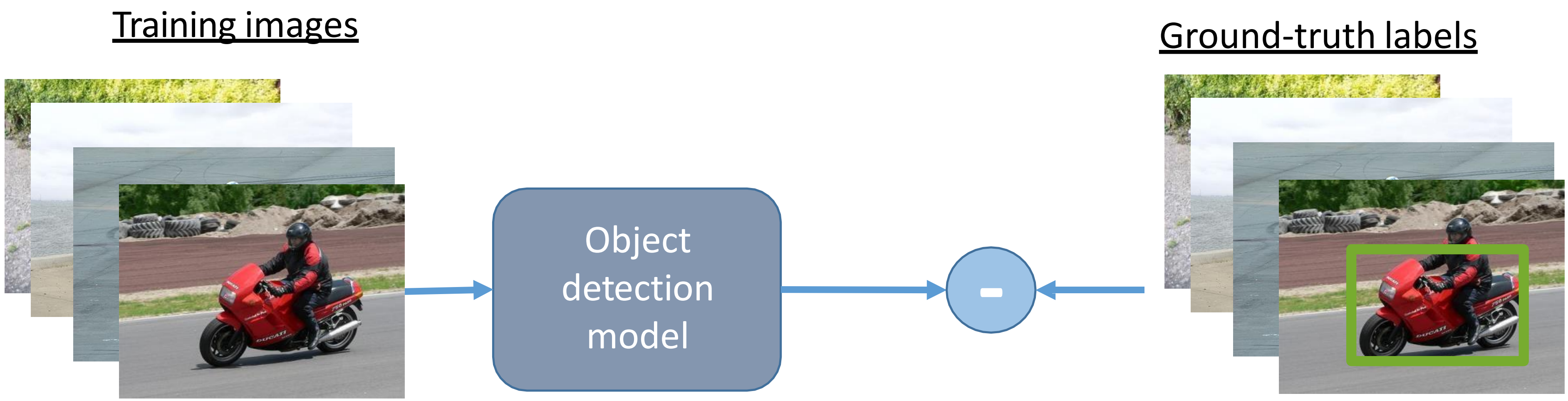
Weak supervision

Lower degree (or cheaper) annotation at train time than the required output at test time

Manual supervision for object recognition



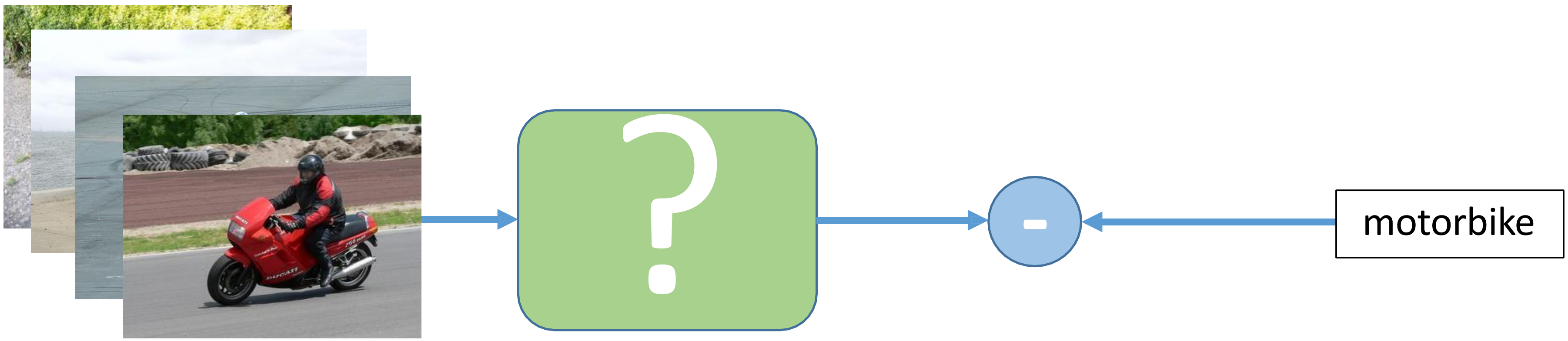
Standard supervised object detection



Weakly supervised object detection (WSOD)

<u>Training images</u>

<u>Ground-truth labels</u>



What can we say at minimum?

1- When image is positive, at least one object instance from target category is present

2- When image is negative, no object instance from target category is present

Assumptions

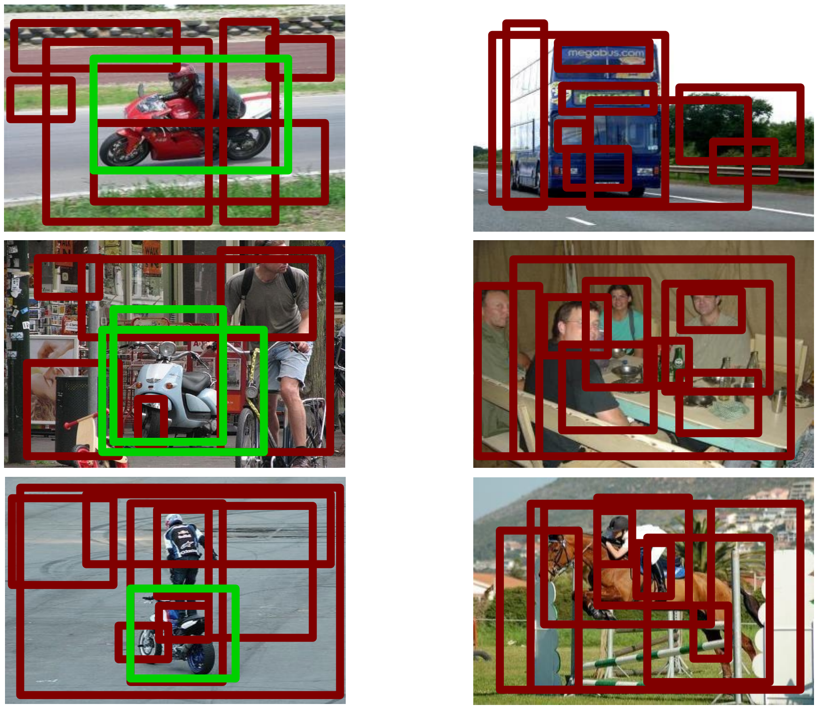
1- There exists a set of features present in positive images and absent in negative images

2- The same features are only present on the target object instances

Multiple-instance learning (MIL)

Dietterich et al. Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence

Positive bags **Negative bags**



bags = images

instances = *m* windows per image

Finally, the aggregated image-level prediction is computed as follows, where greater values of $\hat{p}_c \in [0, 1]$ mean higher likelihood that c is present in the image:

$$\hat{p}_c = \sigma\left(\sum_{i=1}^{m} p_{i,c}^{\text{det}} o_{i,c}^{\text{cls}}\right) \tag{3}$$

Assuming the label $y_c = 1$ if and only if class c is present, the **m**ultiple **i**nstance **d**etection loss used for training the model is defined as:

$$L_{\text{mid}} = -\sum_{c=1}^{C}\left[y_c \log \hat{p}_c + (1 - y_c)\log(1 - \hat{p}_c)\right] \tag{4}$$

[Blaschko NIPS 10, Cinbis CVPR 14, Deselaers ECCV 10, Nguyen ICCV 09, Bilen BMVC 11, Russakovsky ECCV 12, Siva ICCV 11, Siva ECCV 12, Song NIPS 14, Song ICML 14, Bilen BMVC 14]

Adapted from Vitto Ferrari, Hakan Bilen, equations from Ye et al., Cap2Det

Challenges

Intra-class variations:

Appearance

Transformations

Scale

Aspect ratio

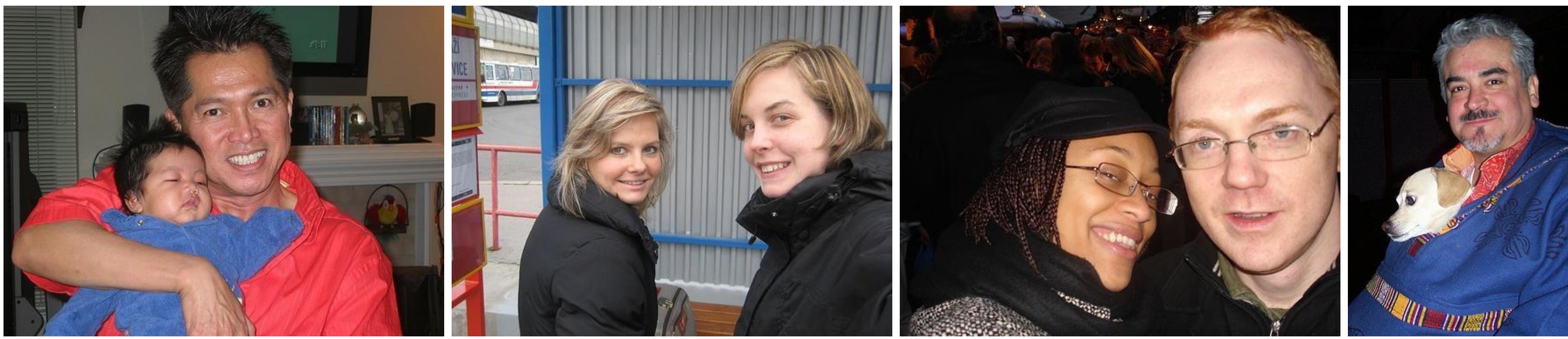
Background Clutter

Occlusions

Challenges

Ambiguity in defining commonality

- Parts



Question: What is a person?

a) Face

b) Face + upper body

c) Face + whole body

Challenges

Ambiguity in defining commonality

- Context




Question: What is a motorbike?

a) Motorbike + Person

b) Person

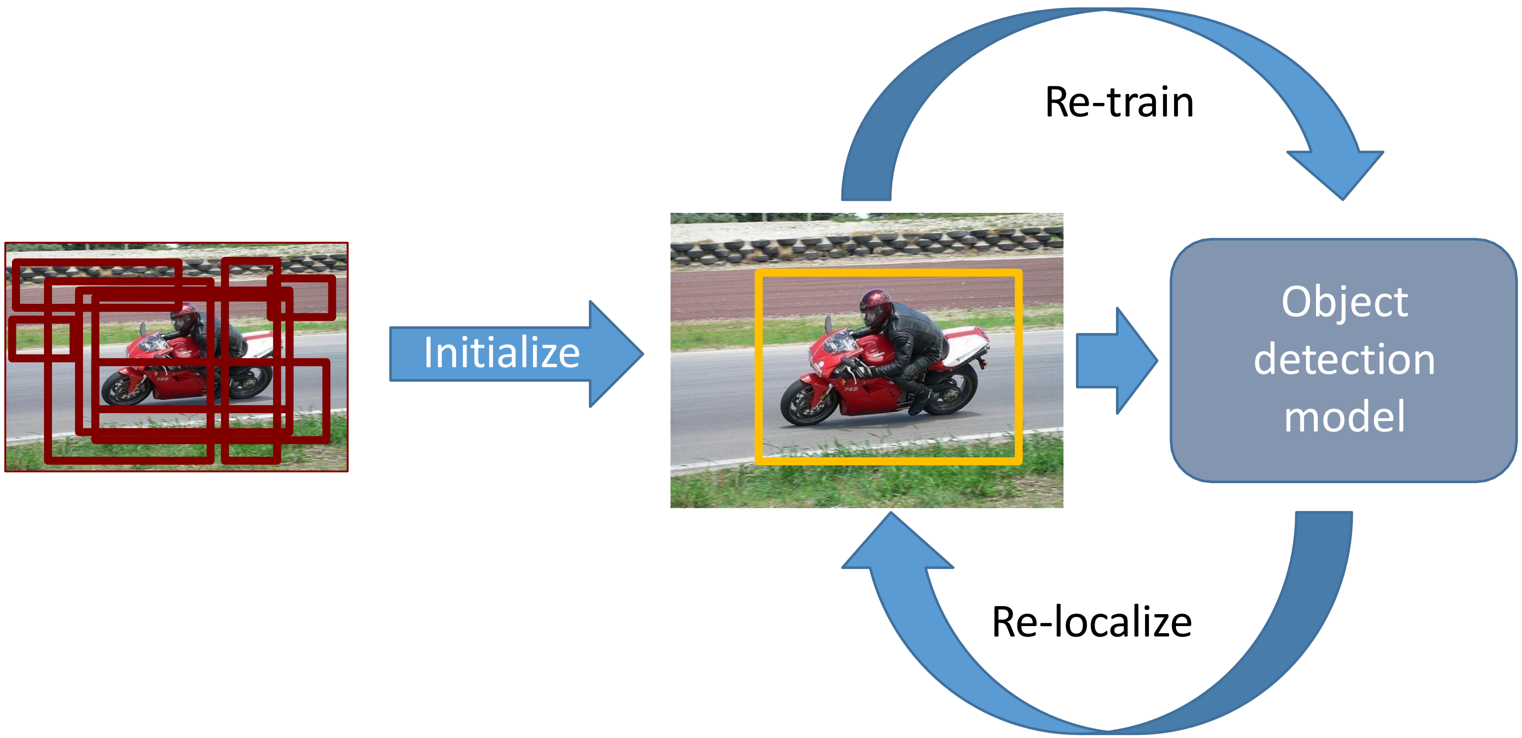
c) Motorbike + Motorbike

d) Motorbike ☺

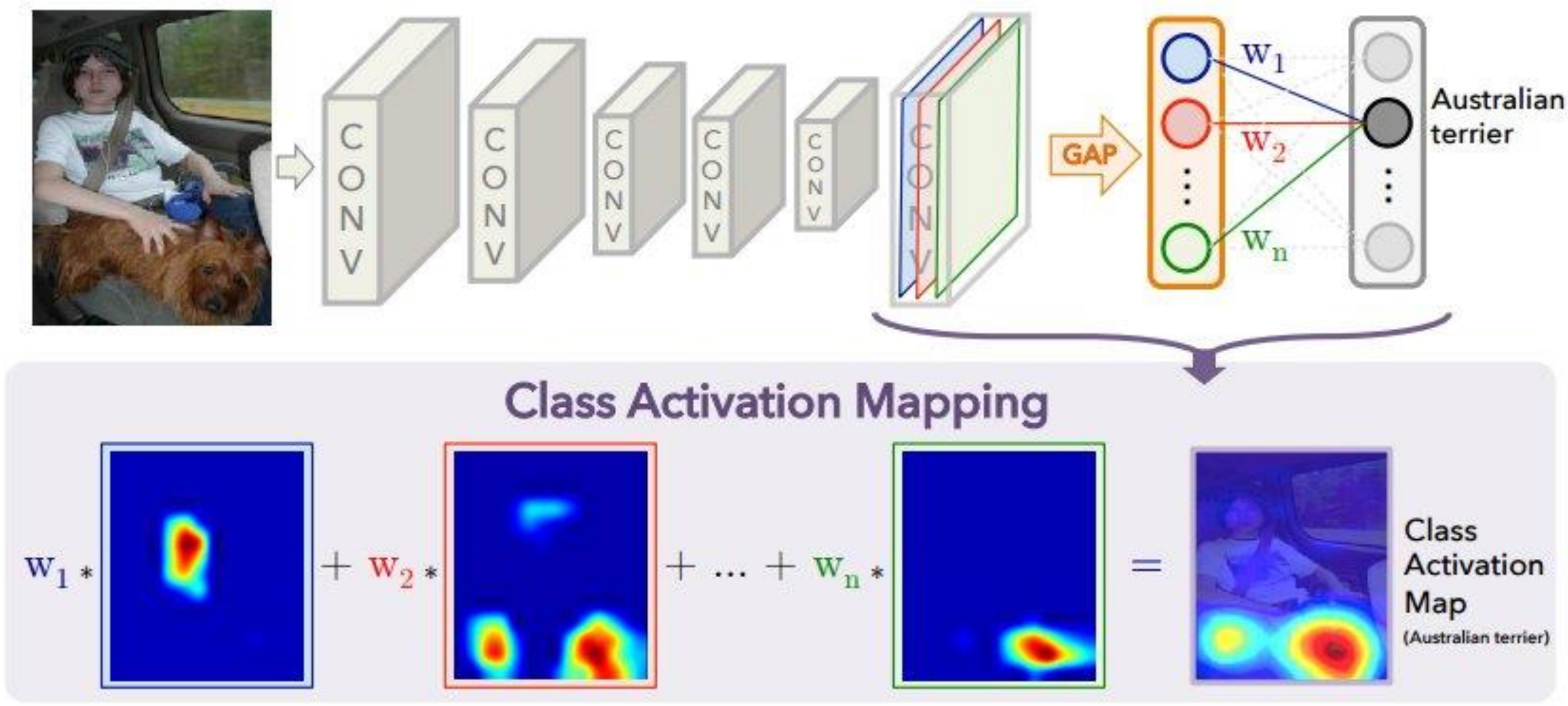
Challenges

Alternating optimization (Re-localize + Re-train)

- Sensitive to initialization (local minimum)
- Overfitting (locking) to predicted windows



Class activation maps



Zhou et al., "Learning Deep Features for Discriminative Localization", CVPR 2016

Class activation maps

- Let $f_k(x, y)$ be the activation in the k-th map at location (x, y)
- Global average pooling: $F^k = \Sigma_{x,y} f_k(x, y)$
- Input to softmax is $S_c = \Sigma_k w^c_k F^k$ where w^c_k is the weight for class c and map k

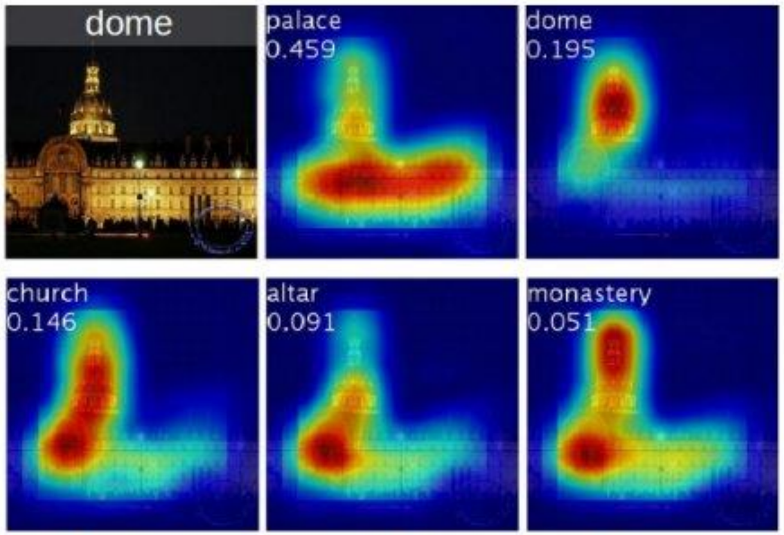
$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x,y) = \sum_{x,y} \sum_k w_k^c f_k(x,y)$$

- Map for class c:

$$M_c(x,y) = \sum_k w_k^c f_k(x,y)$$

Zhou et al., "Learning Deep Features for Discriminative Localization", CVPR 2016

Class activation maps



Class activation maps of top 5 predictions

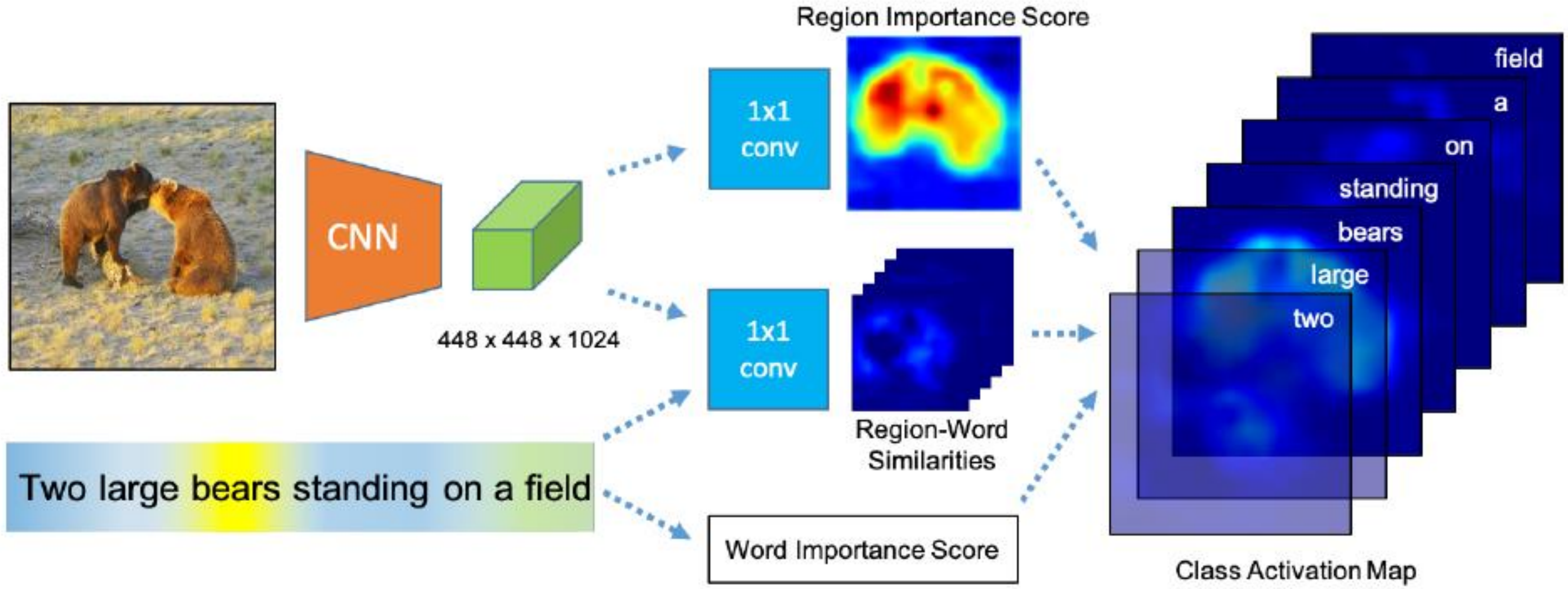
Class activation maps for one object class

Table 3. Localization error on the ILSVRC test set for various weakly- and fully- supervised methods.

Method	supervision	top-5 test error
GoogLeNet-GAP (heuristics)	weakly	**37.1**
GoogLeNet-GAP	weakly	42.9
Backprop [23]	weakly	46.4
GoogLeNet [25]	full	26.7
OverFeat [22]	full	29.9
AlexNet [25]	full	34.2

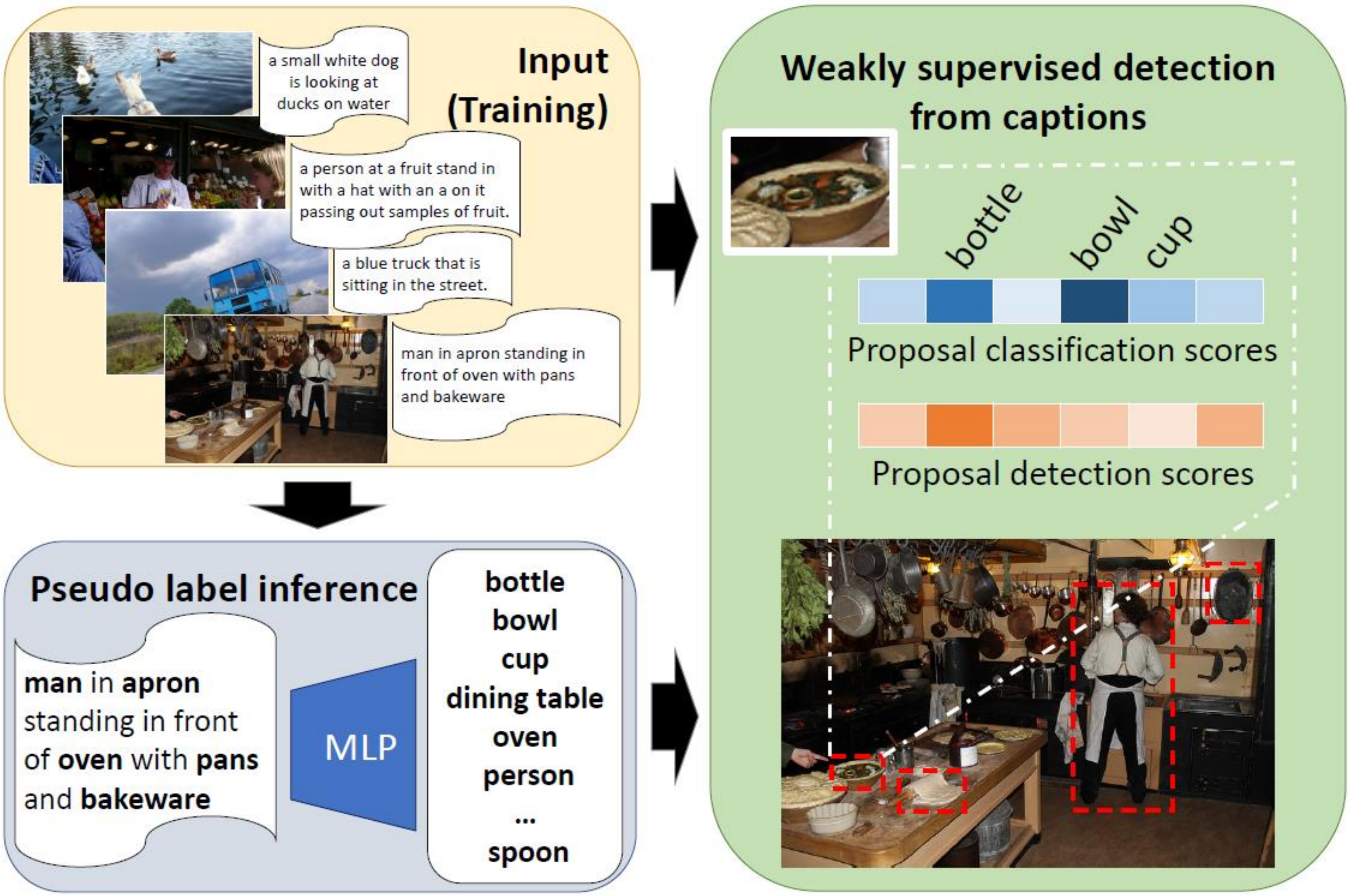
Zhou et al., "Learning Deep Features for Discriminative Localization", CVPR 2016

Localization from captions



Ye et al., “Learning to discover and localize visual objects with open vocabulary”, arxiv 2018

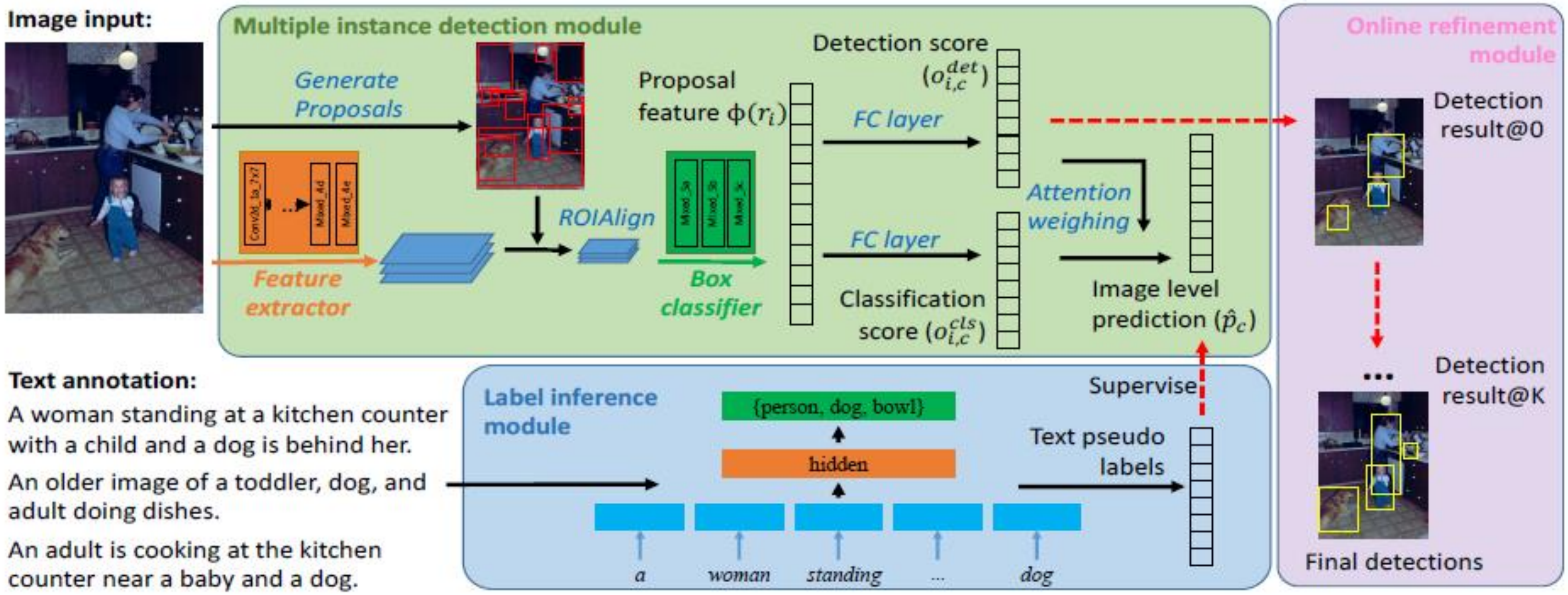
Learning object detectors from captions



Ye et al., ICCV 2019

Learning to amplify weak caption signal

- Use pseudo labels in free-form text as supervision
- Label inference module performs basic reasoning based on the textual context
- Multiple instance detection module predicts detection/classification scores from proposal features



Ye et al., ICCV 2019

Weakly supervised object detection results

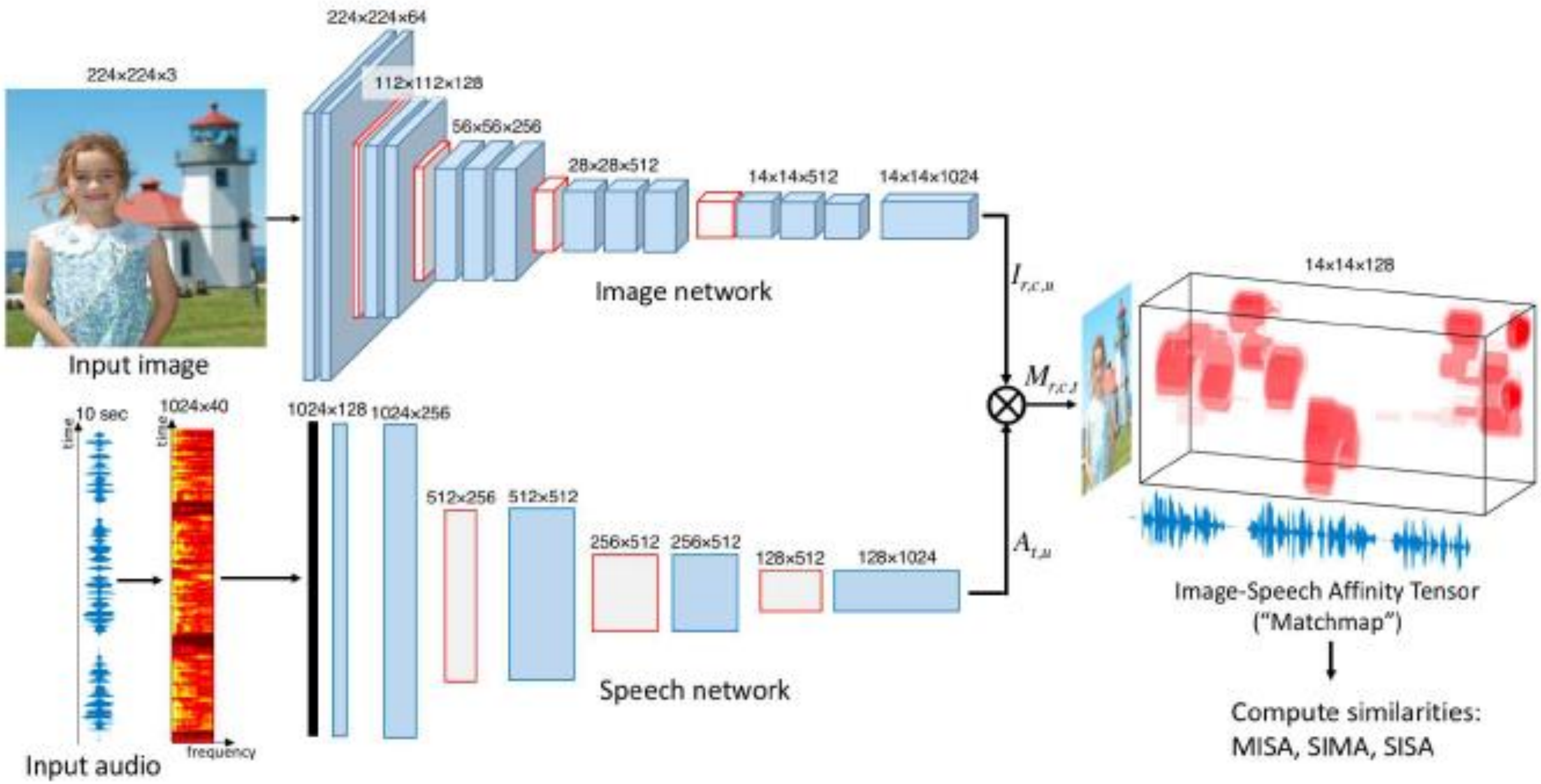
Methods	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
Training on different datasets using ground-truth labels:																					
GT-LABEL VOC	68.7	49.7	53.3	27.6	14.1	64.3	58.1	76.0	23.6	59.8	50.7	57.4	48.1	63.0	15.5	18.4	49.7	55.0	48.4	67.8	48.5
GT-LABEL COCO	65.3	50.3	53.2	25.3	16.2	68.0	54.8	65.5	20.7	62.5	51.6	45.6	48.6	62.3	7.2	24.6	49.6	34.6	51.1	69.3	46.3
Training on COCO dataset using captions:																					
EXACTMATCH (EM)	63.0	**50.3**	50.7	25.9	**14.1**	64.5	50.8	33.4	17.2	49.0	48.2	46.7	44.2	59.2	10.4	14.3	49.8	37.7	21.5	47.6	39.9
EM + GLOVEPSEUDO	**66.6**	43.7	53.3	29.4	13.6	65.3	**51.6**	33.7	15.6	50.7	46.6	45.4	47.6	**62.1**	8.0	**15.7**	48.6	46.3	30.6	36.4	40.5
EM + LEARNEDGLOVE	64.1	49.9	**58.6**	24.9	13.2	**66.9**	49.2	26.9	13.1	57.7	**52.8**	42.6	**53.2**	58.6	14.3	15.0	45.2	50.3	34.1	43.5	41.7
EM + EXTENDVOCAB	65.0	44.9	49.2	**30.6**	13.6	64.1	50.8	28.0	**17.8**	59.8	45.5	**56.1**	49.4	59.1	16.8	15.2	**51.1**	**57.8**	14.0	**61.8**	42.5
EM + TEXTCLSF	63.8	42.6	50.4	29.9	12.1	61.2	46.1	**41.6**	16.6	**61.2**	48.3	55.1	51.5	59.7	**16.9**	15.2	50.5	53.2	**38.2**	48.2	**43.1**
Training on Flickr30K dataset using captions:																					
EXACTMATCH (EM)	**46.6**	**42.9**	42.0	9.6	7.7	31.6	44.8	53.2	13.1	28.0	39.1	43.2	31.9	**52.5**	4.0	**5.1**	38.0	28.7	**15.8**	41.1	31.0
EM + EXTENDVOCAB	37.8	37.6	35.5	11.0	**10.3**	18.0	47.9	51.3	**17.7**	25.5	37.0	47.9	**35.2**	46.1	**15.2**	0.8	27.8	35.6	5.8	42.0	29.3
EM + TEXTCLSF	24.1	38.8	**44.5**	**13.3**	6.2	**38.9**	**49.9**	**60.4**	12.4	**47.4**	**39.2**	**59.3**	34.8	48.1	10.7	0.3	**42.4**	**39.4**	14.1	**47.3**	**33.6**

Table 1: **Average precision (in %) on the VOC 2007 test set (learning from COCO and Flickr30K captions)**. We learn the detection model from the COCO captions describing the 80 objects, but evaluate on only the overlapping 20 VOC objects.

- Caption supervision model comparable to one trained with image-level labels
- Other ways of obtaining pseudo labels are inferior
- Results (and text classifier) generalize to other datasets

Ye et al., ICCV 2019

Localization from sound



Harwath et al., "Jointly Discovering Visual Objects and Spoken Words from Raw Sensory Input", ECCV 2018

Localization from sound

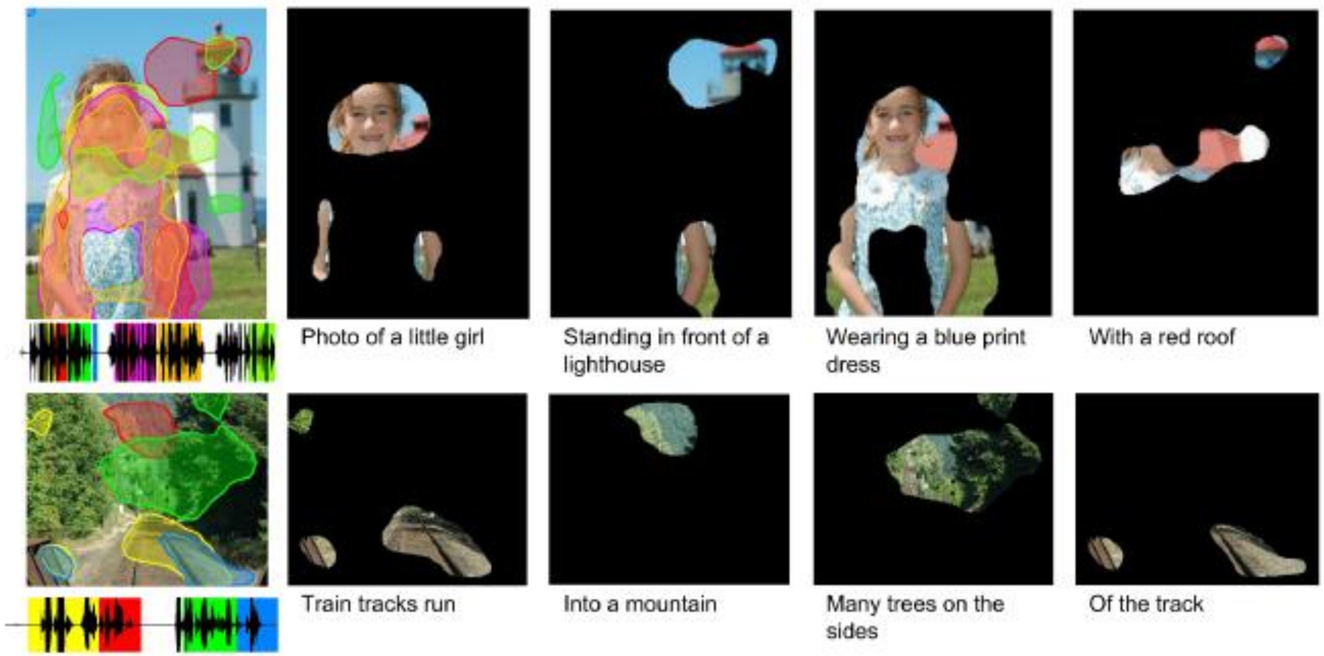
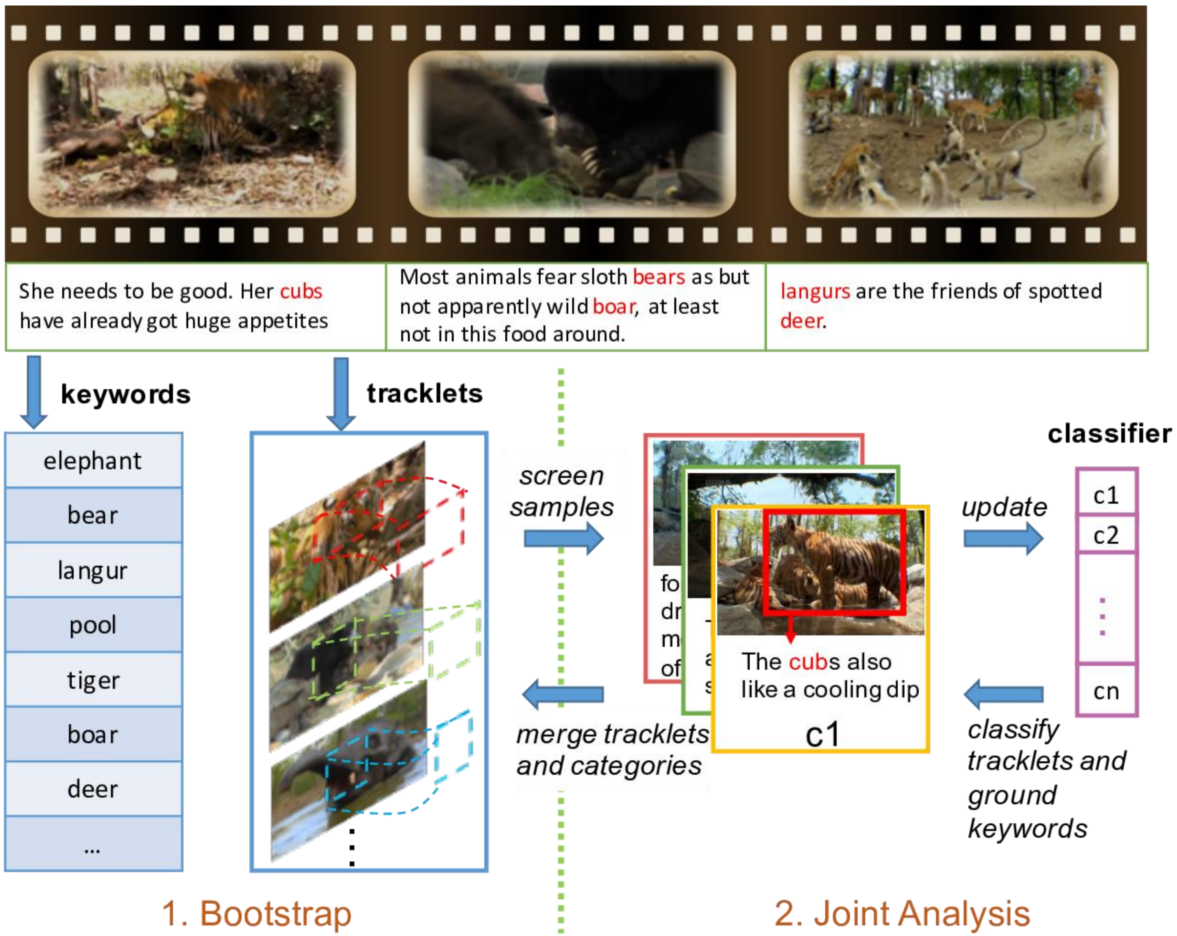


Fig. 7: On the left are shown two images and their speech signals. Each color corresponds to one connected component derived from two matchmaps from a fully random MISA network. The masks on the right display the segments that correspond to each speech segment. We show the caption words obtained from the ASR transcriptions below the masks. Note that those words were never used for learning, only for analysis.

Harwath et al., “Jointly Discovering Visual Objects and Spoken Words from Raw Sensory Input”, ECCV 2018

Detection from documentaries



Chen et al., "Discover and Learn New Objects from Documentaries", CVPR 2017

Domain adaptation

Standard Visual Recognition

Training data

Test data

Train a classifier on the training data and directly apply it to the test data

Domain Shift

Training data

Test data

Source domain

Target domain

A classifier trained on one domain may perform poorly on another domain

Semi-supervised vs Unsupervised

- Semi-supervised: Some labeled target data, but not enough to train from scratch

Source data

Fully-labeled

Target data

A few labels

Semi-supervised vs Unsupervised

- Unsupervised: No labels for the target data

Source data

Fully-labeled

Target data

Single vs Multiple Source Domains

Source domain 1

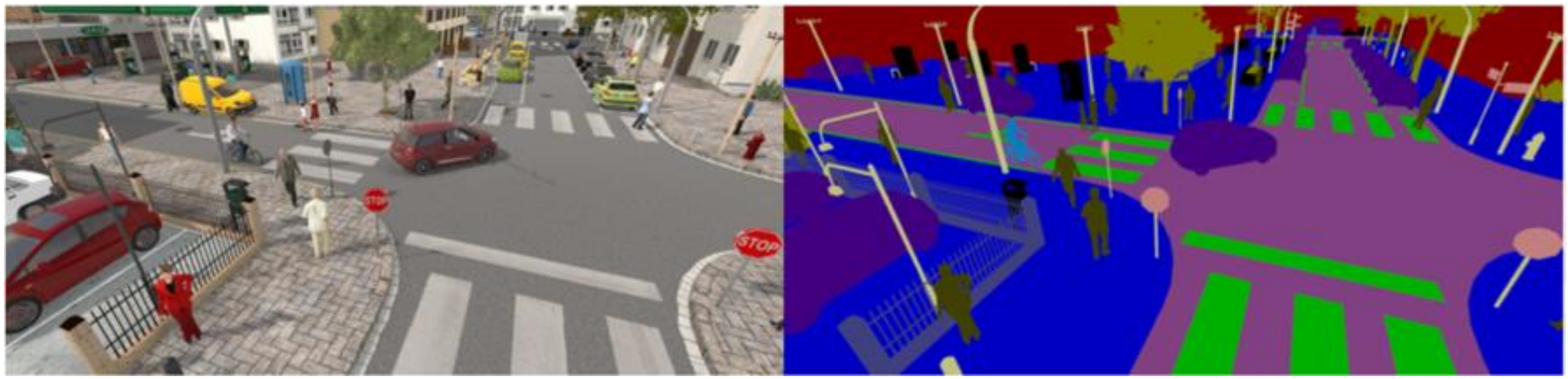
Source domain 2

Target domain

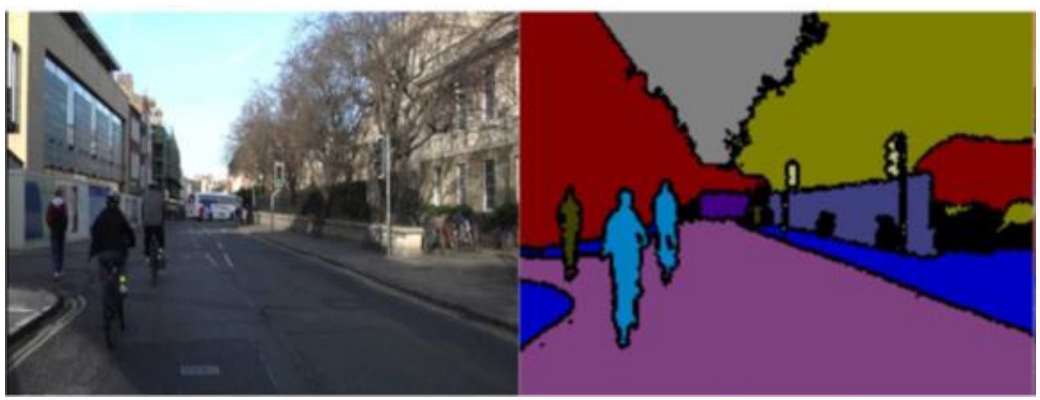
- Moving towards domain generalization

Domain Adaptation: Other Scenarios

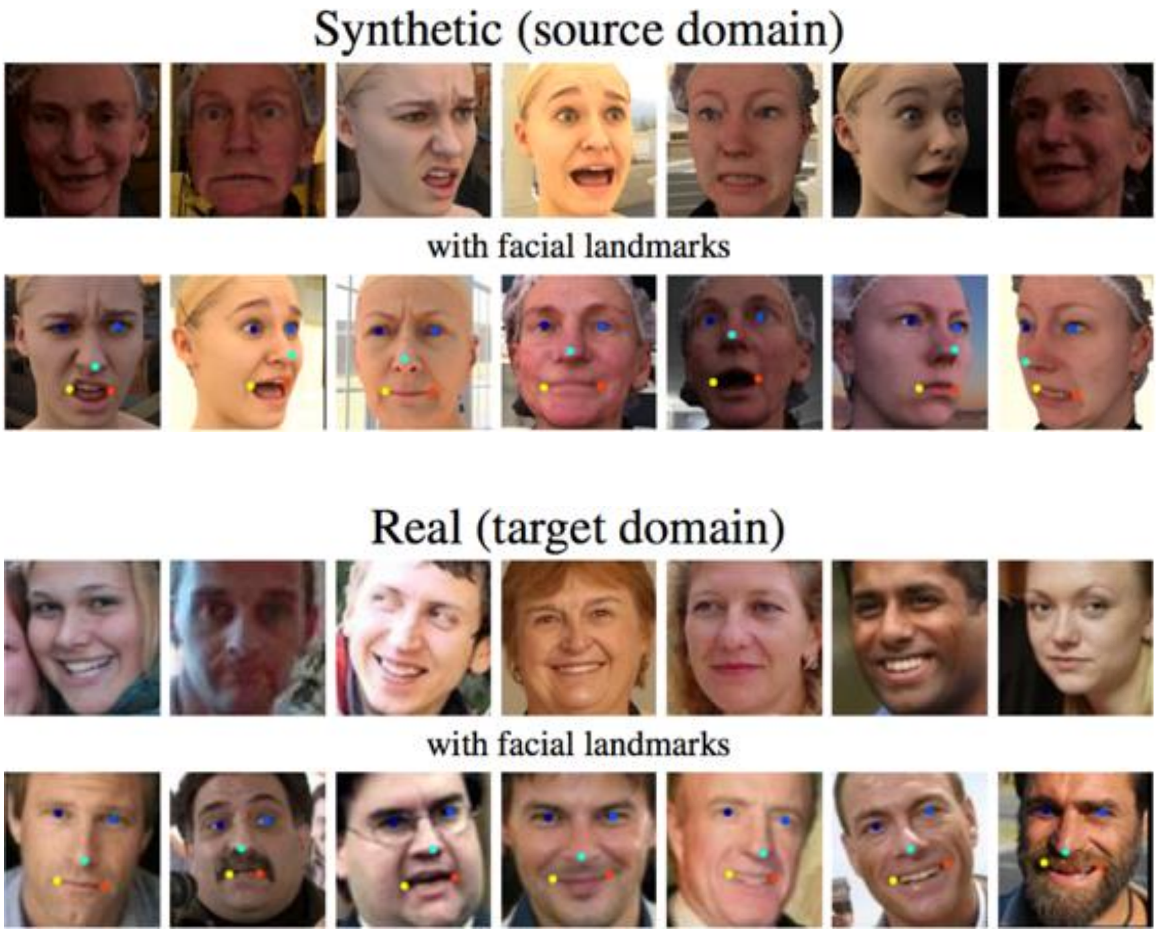
Synthetic (source domain)



Real (target domain)

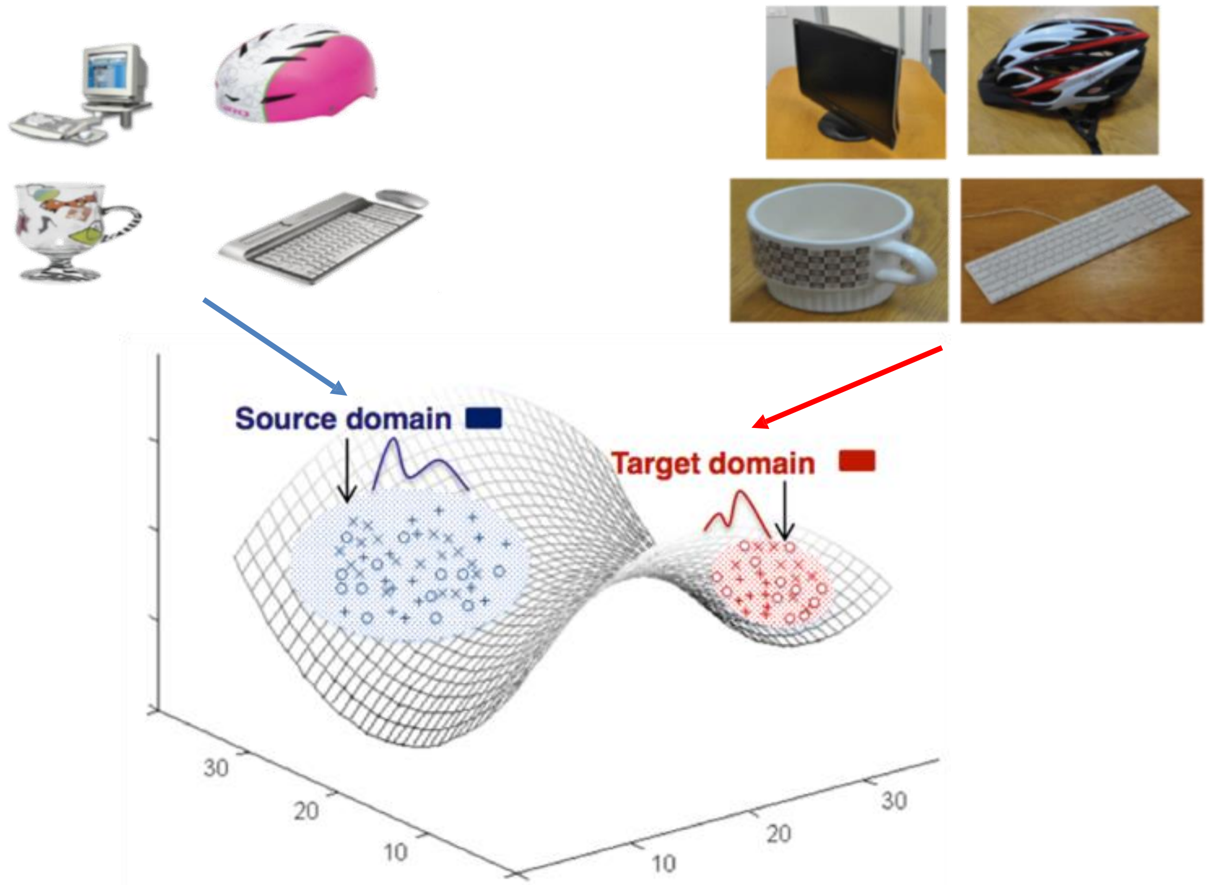


Domain Adaptation: Other Scenarios



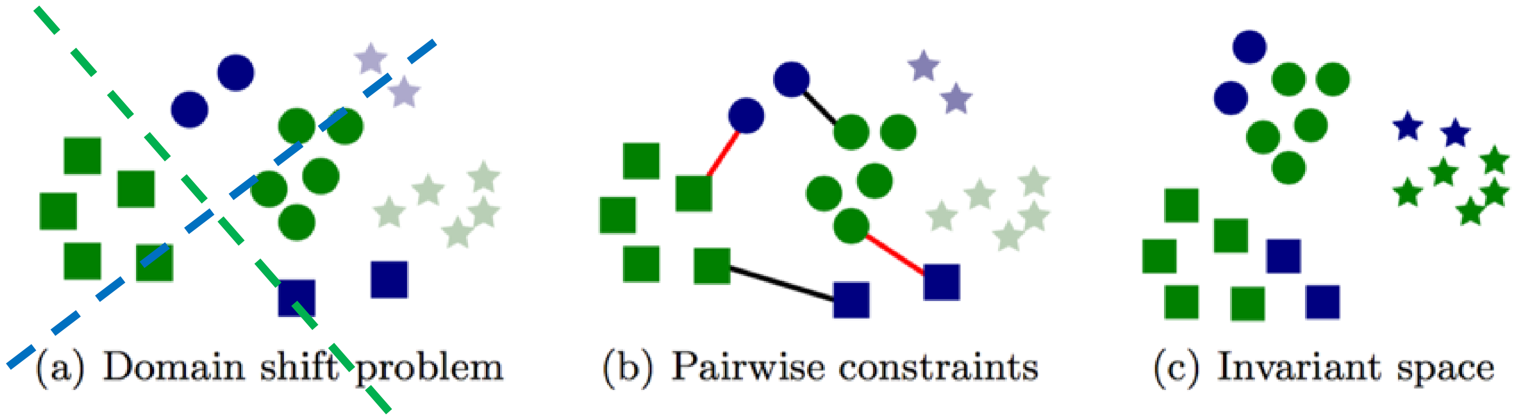
Domain Shift

- The domain shift is defined as a difference in the distribution of the source and target samples



Metric Learning for Domain Adaptation

- Saenko et al., Adapting Visual Category Models to New Domains, ECCV 2010



- Learning a distance:

$$d_W(\mathbf{x}_s^i, \mathbf{x}_t^j) = (\mathbf{x}_s^i - \mathbf{x}_t^j)^T W (\mathbf{x}_s^i - \mathbf{x}_t^j)$$

Metric Learning for Domain Adaptation

- Semi-supervised domain adaptation: Pairwise constraints based on labels

$$
\begin{aligned}
d_W(\mathbf{x}_s^i, \mathbf{x}_t^j) &\leq u \text{ if } y^i = y^j \\
d_W(\mathbf{x}_s^i, \mathbf{x}_t^j) &\geq l \text{ if } y^i \neq y^j
\end{aligned}
$$

- Learning formulation:

$$
\begin{aligned}
\min_{\mathbf{W} \succeq \mathbf{0}} \quad & \mathrm{tr}(W) - \log\det W \\
\text{s.t.} \quad & d_W(\mathbf{x}_s^i, \mathbf{x}_t^j) \leq u \text{ if } y^i = y^j \\
& d_W(\mathbf{x}_s^i, \mathbf{x}_t^j) \geq l \text{ if } y^i \neq y^j
\end{aligned}
$$

From Semi-supervised to Unsupervised DA

- Early approaches require labeled target samples
- The unsupervised scenario assumes no target labels are available

Source data

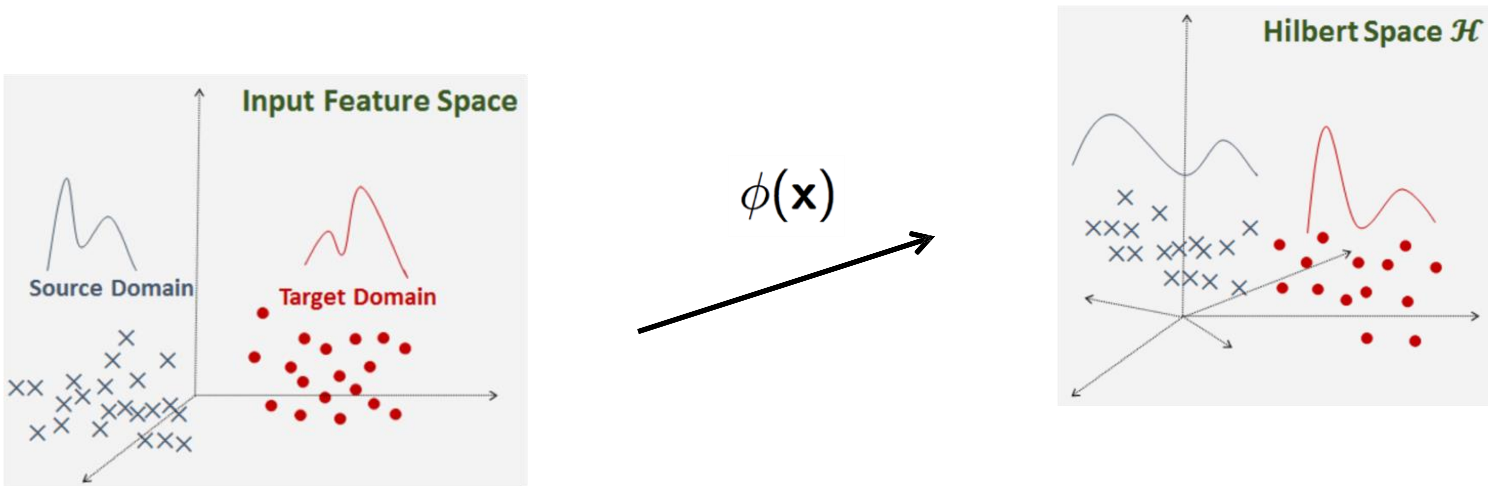
Fully-labeled

Target data

~~A few labels~~

Maximum Mean Discrepancy

- Compare the mean of two samples
  - Gretton et al., JMLR 2012



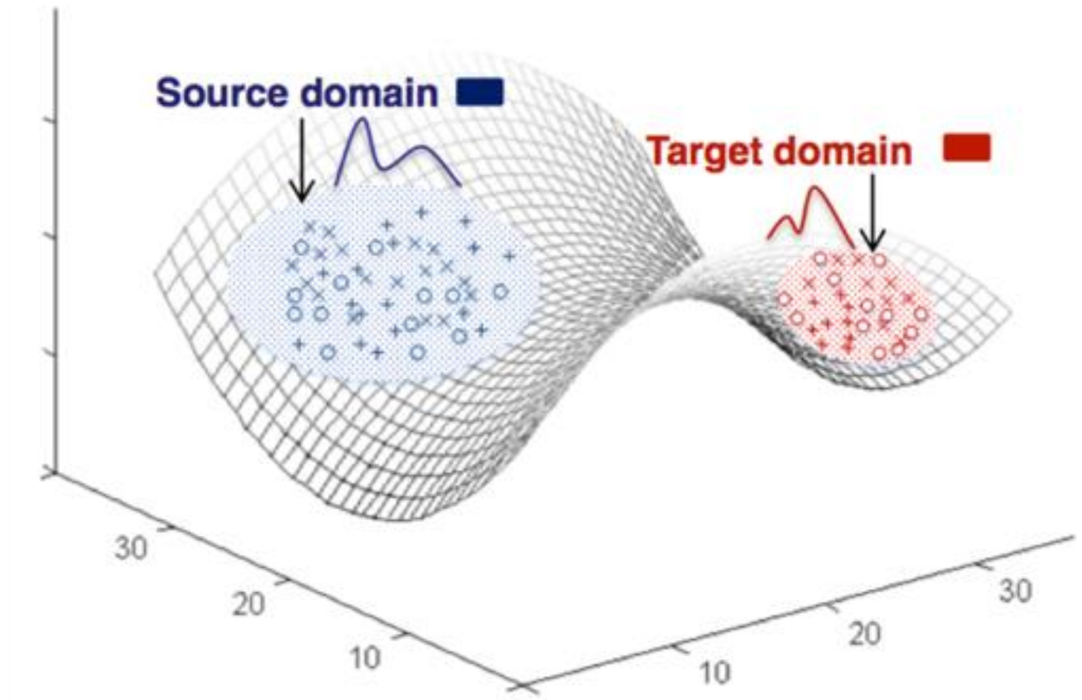
$$D_{MMD}(X_s, X_t) = \left\| \frac{1}{n} \sum_{i=1}^{n} \phi(x_s^i) - \frac{1}{m} \sum_{j=1}^{m} \phi(x_t^j) \right\|_{\mathcal{H}}$$

$$= \left(\sum_{i,j=1}^{n} \frac{k(x_s^i, x_s^j)}{n^2} + \sum_{i,j=1}^{m} \frac{k(x_t^i, x_t^j)}{m^2} - 2 \sum_{i,j=1}^{n,m} \frac{k(x_s^i, x_t^j)}{nm} \right)^{\frac{1}{2}}$$

Adapted from Mathieu Salzmann

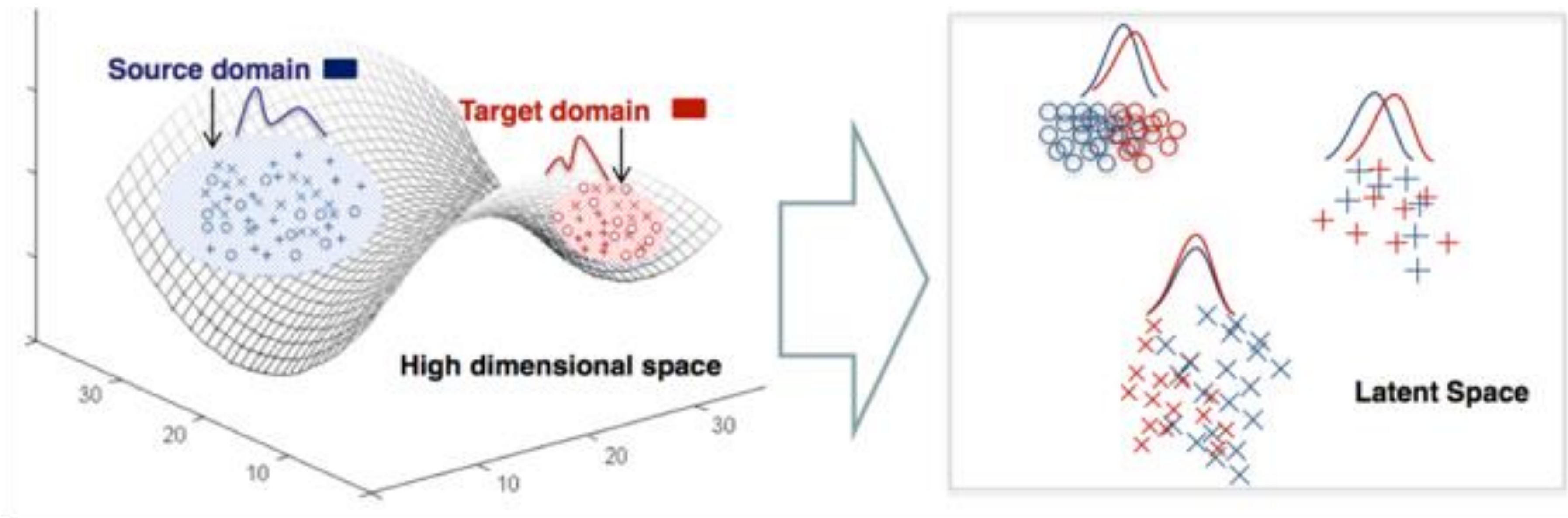
Sample Reweighting/Selection

- What happens if the original distributions are very different?



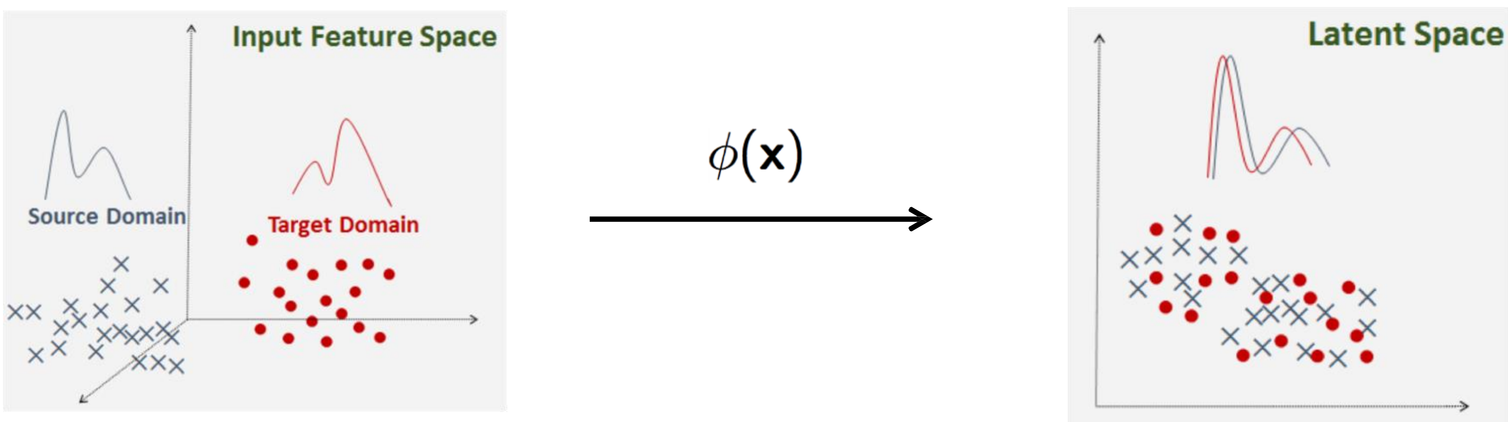
Transformation Learning

- Learn a mapping to a latent space where the distributions are similar



Transfer Component Analysis (TCA)

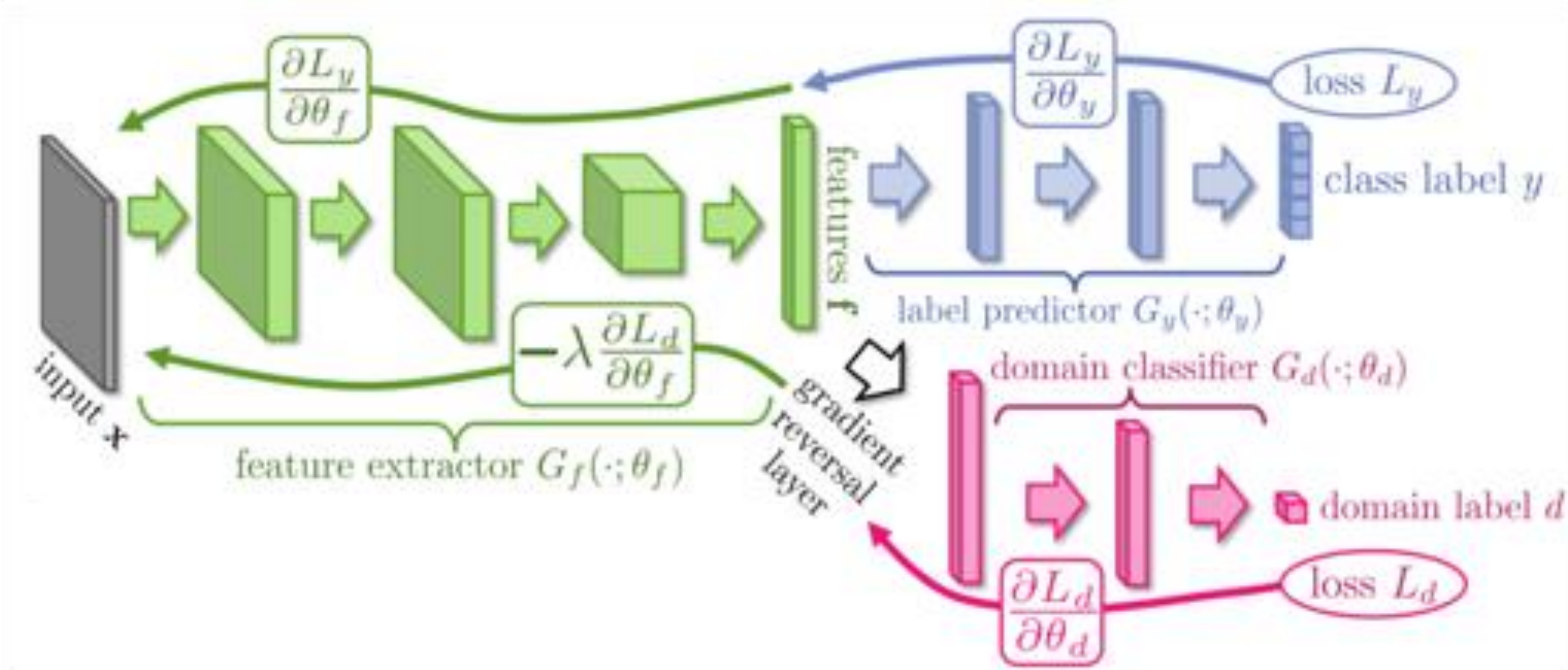
- Pan et al., TNN 2011
  - Motivation: Learn a nonlinear mapping that minimizes the MMD



- Would involve learning a kernel matrix

Domain Adversarial Networks

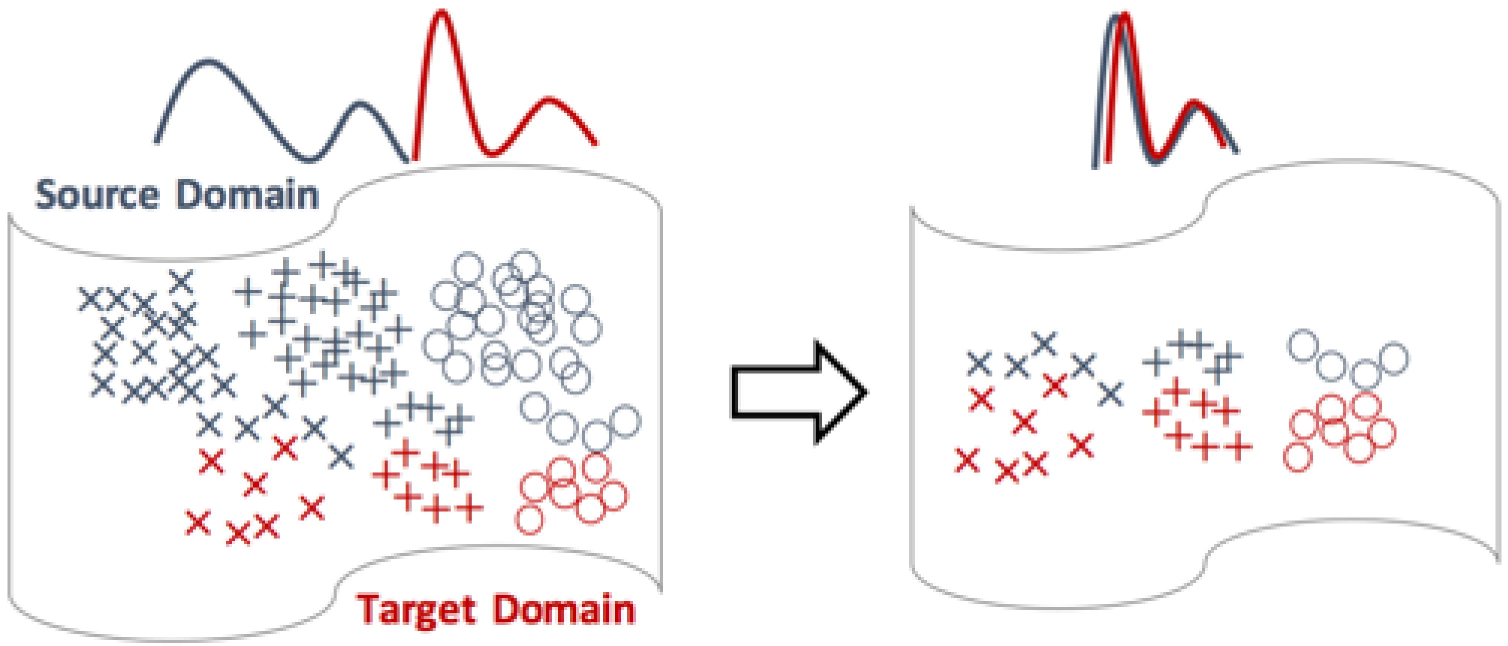
- Ganin & Lempitsky, ICML 2015; Ajakan et al., 2014
  - With domain-invariant features, classifying from which domain a sample comes should be difficult



- Shown to optimize a H-divergence between the source and target data

Sample Reweighting/Selection

- Assign a weight to each source sample to make the distributions similar



Sample Reweighting/Selection

- Gretton et al., JRSS 2012: Sample reweighting

$$\min_{\beta} \quad \left\| \frac{1}{n} \sum_{i=1}^{n} \beta_i \phi(\mathbf{x}_s^i) - \frac{1}{m} \sum_{i=1}^{m} \phi(\mathbf{x}_t^i) \right\|^2$$

MMD

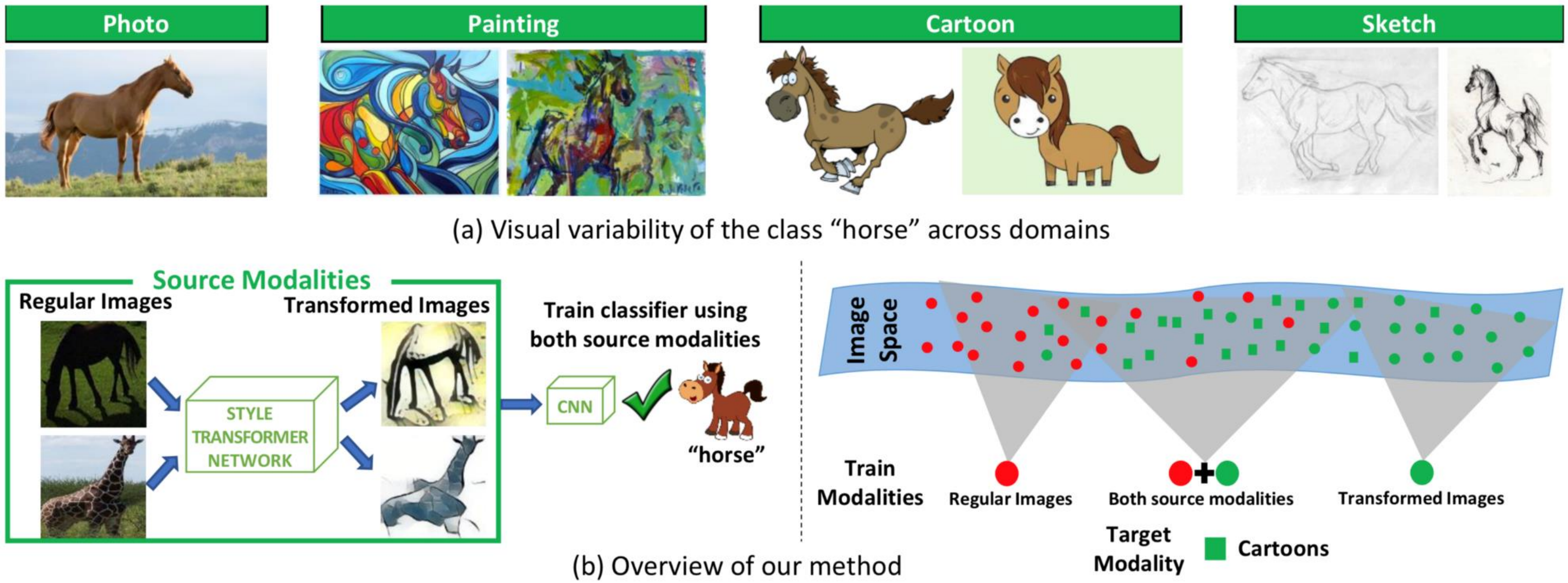
$$\text{s.t.} \quad \beta_i \in [0, B] \,, \; \forall \, 1 \leq i \leq n$$

Bound on the weights

$$\left| \sum_{i=1}^{n} \beta_i - n \right| \leq n\epsilon$$

Encourage the weights to define a probability distribution

Adapting classifiers



Thomas and Kovashka, "Artistic Object Recognition by Unsupervised Style Adaptation", ACCV 2018

Adapting classifiers

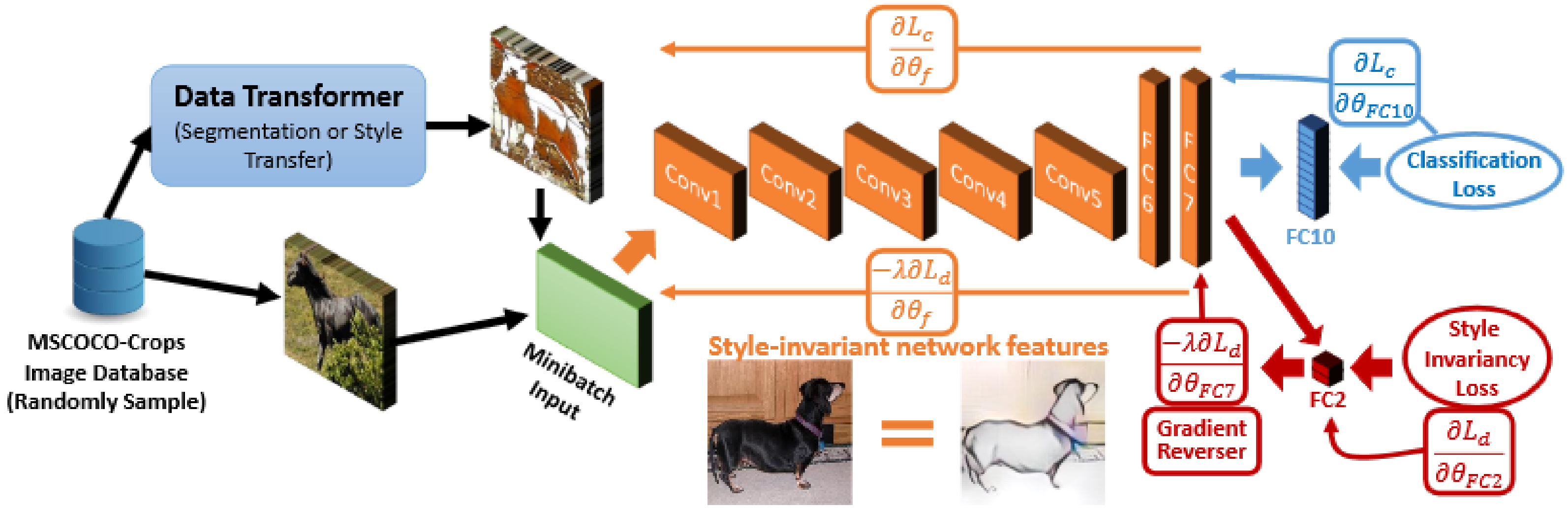


Fig. 2. Training with multiple modalities and style-invariance constraint. We train networks on real and synthetic data. We show an example of style transfer transforming photos into labeled synthetic cartoons. The style-invariance loss trains the FC2 layer to predict which modality the image came from. During backpropagation, we reverse its gradient before propagating it to the layers used by both classifiers. This encourages those layers to learn style-invariant features.

Thomas and Kovashka, "Artistic Object Recognition by Unsupervised Style Adaptation", ACCV 2018

Adapting detectors

Figure 1. **Illustration of different datasets for autonomous driving:** From top to bottom-right, example images are taken from: *KITTI*[17], *Cityscapes*[5], *Foggy Cityscapes*[49], *SIM10K*[30]. Though all datasets cover urban scenes, images in those dataset vary in style, resolution, illumination, object size, *etc*. The visual difference between those datasets presents a challenge for applying an object detection model learned from one domain to another domain.

Chen et al., "Domain Adaptive Faster R-CNN for Object Detection in the Wild", CVPR 2018

Adapting detectors

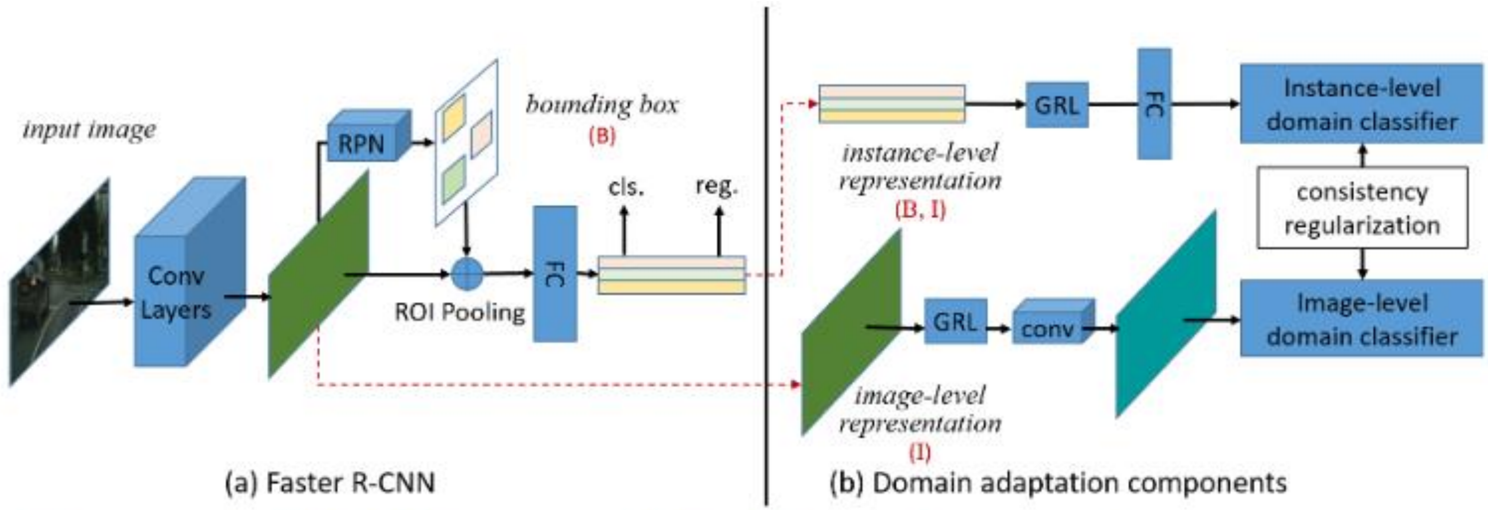


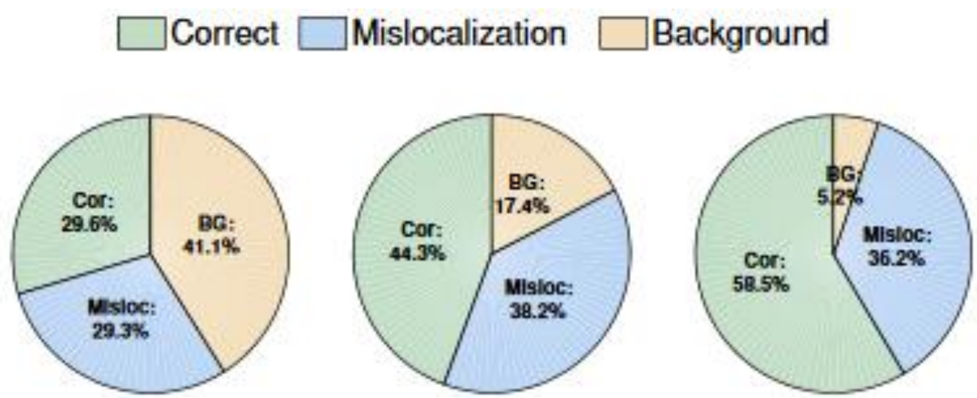
Figure 2. **An overview of our Domain Adaptive Faster R-CNN model:** we tackle the domain shift on two levels, the image level and the instance level. A domain classifier is built on each level, trained in an adversarial training manner. A consistency regularizer is incorporated within these two classifiers to learn a domain-invariant RPN for the Faster R-CNN model.

Chen et al., “Domain Adaptive Faster R-CNN for Object Detection in the Wild”, CVPR 2018

Adapting detectors

	img	ins	cons	car AP
Faster R-CNN				30.12
Ours	✓			33.03
		✓		35.79
	✓	✓		37.86
	✓	✓	✓	**38.97**

Table 1. The average precision (AP) of *Car* on the *Cityscapes* validation set. The models are trained using the *SIM 10k* dataset as the source domain and the *Cityscapes* training set as the target domain. *img* is short for *image-level alignment*, *ins* for *instance-level alignment* and *cons* is short for our *consistency loss*



(a) Faster RCNN (b) Ours (Ins. Only) (c) Ours (Img Only)

Figure 3. Error Analysis of Top Ranked Detections

Chen et al., “Domain Adaptive Faster R-CNN for Object Detection in the Wild”, CVPR 2018