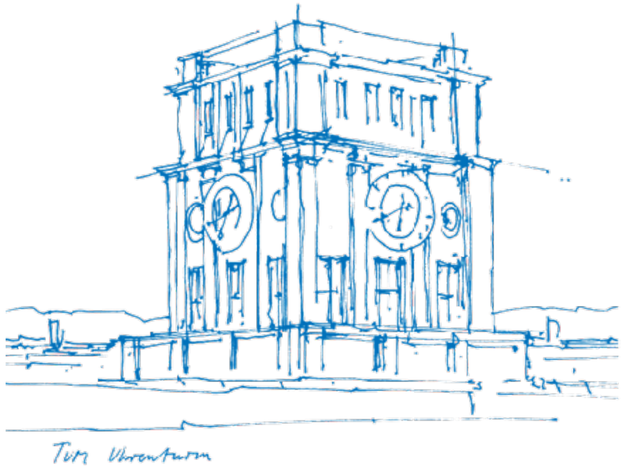
Computer Vision III:

Two-stage object detectors

Nikita Araslanov
08.11.2022

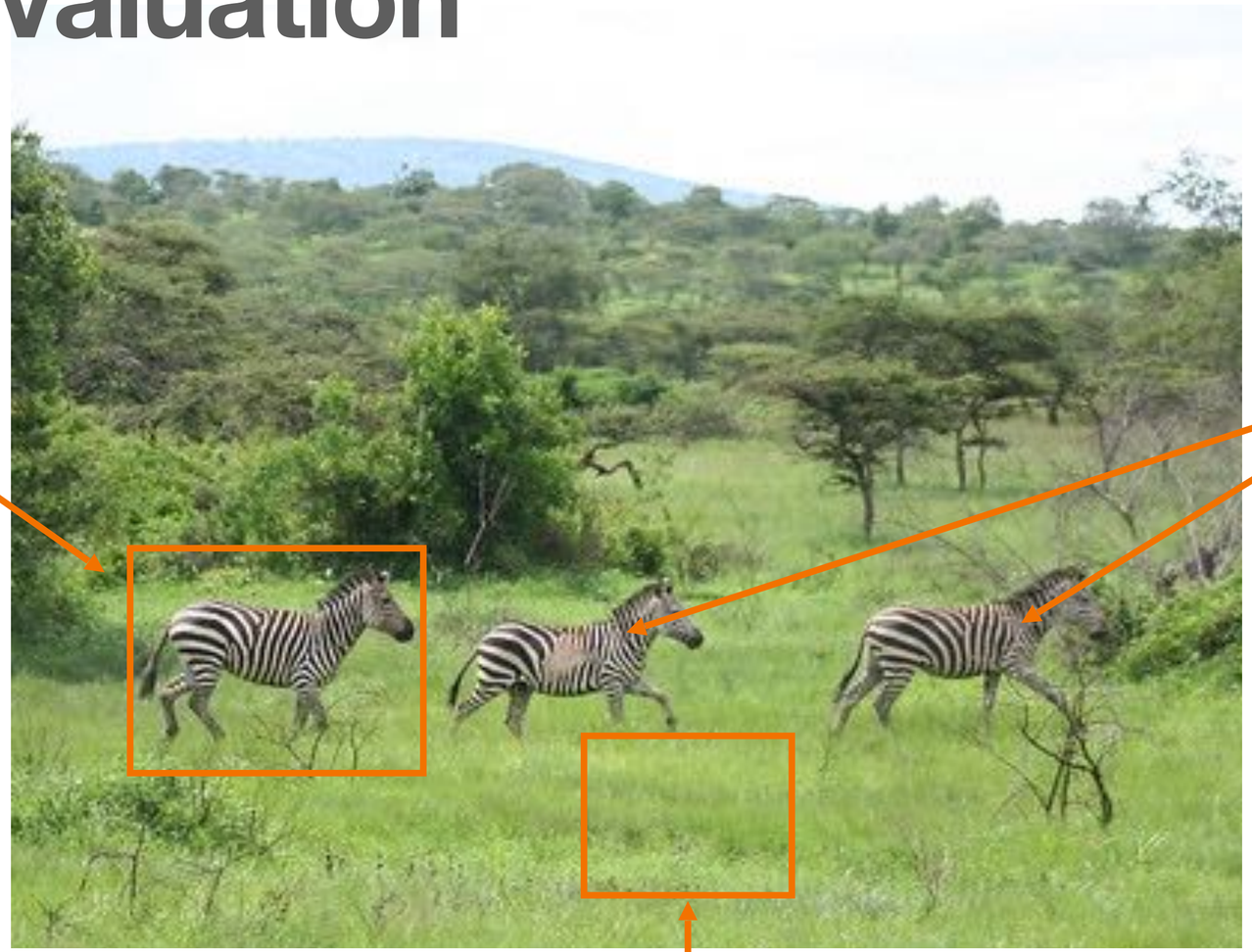
Content credit:
Prof. Laura Leal-Taixé
https://dvl.in.tum.de



Detection evaluation

Detection evaluation

TP = True positive



FN = False negative

FP = False positive

Detection evaluation

- Precision: how accurate your predictions are.

$$Precision = \frac{TP}{TP + FP}$$

# of predicted boxes

- Recall: how good you are at finding all positives

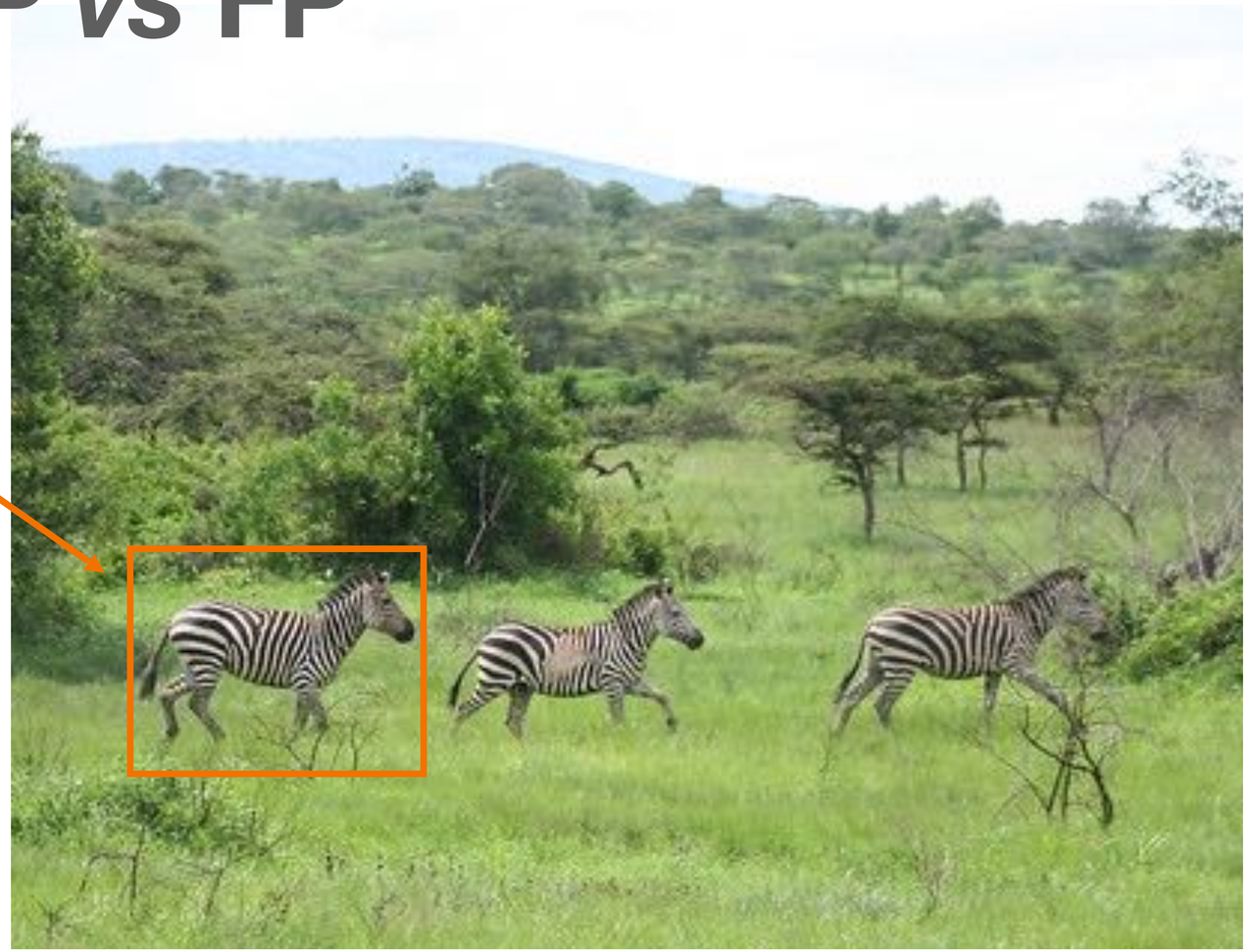
$$Recall = \frac{TP}{TP + FN}$$

# of ground-truth boxes

TP = True positives FP = False positives FN = False negatives

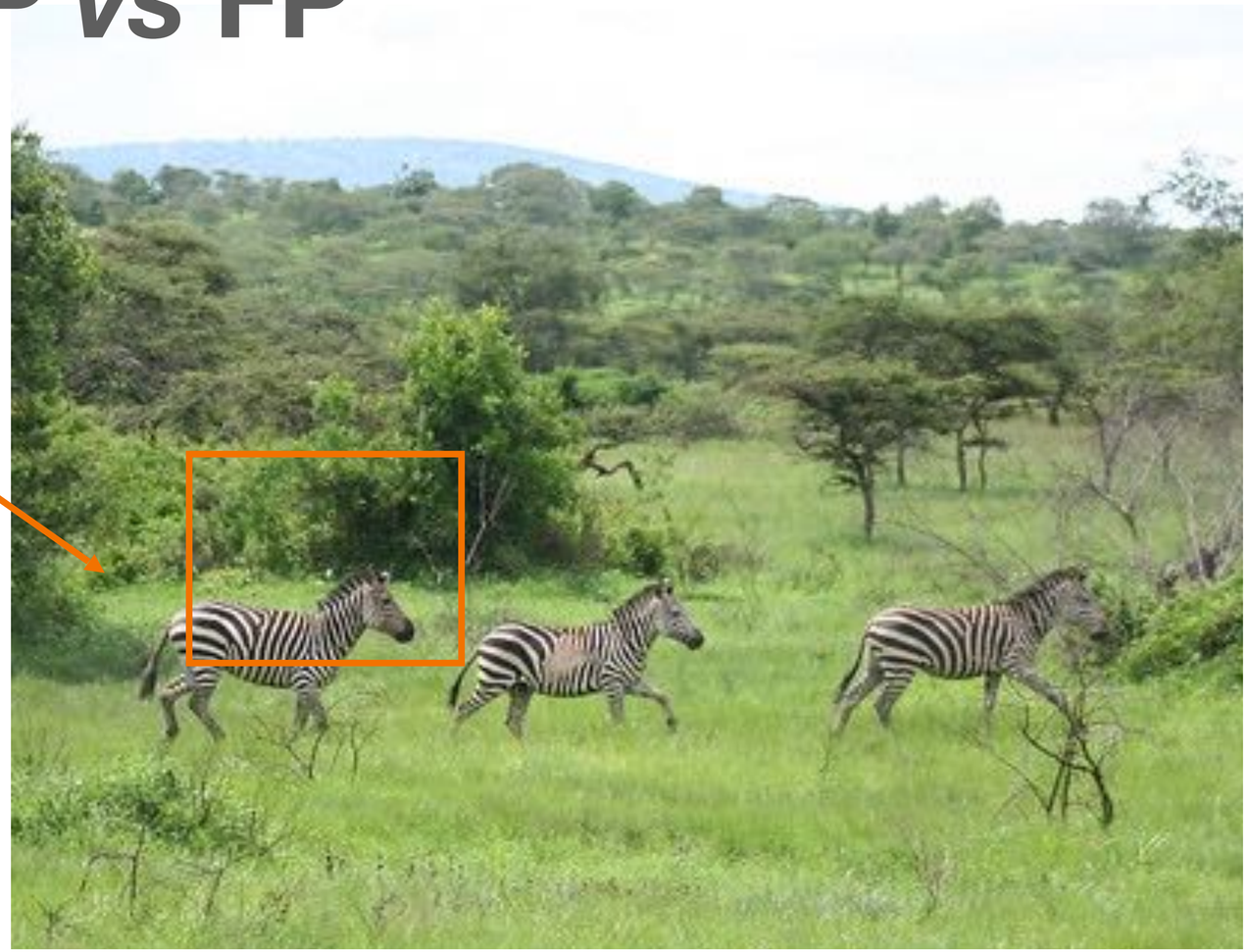
Counting TP *vs* FP

TP = True positive



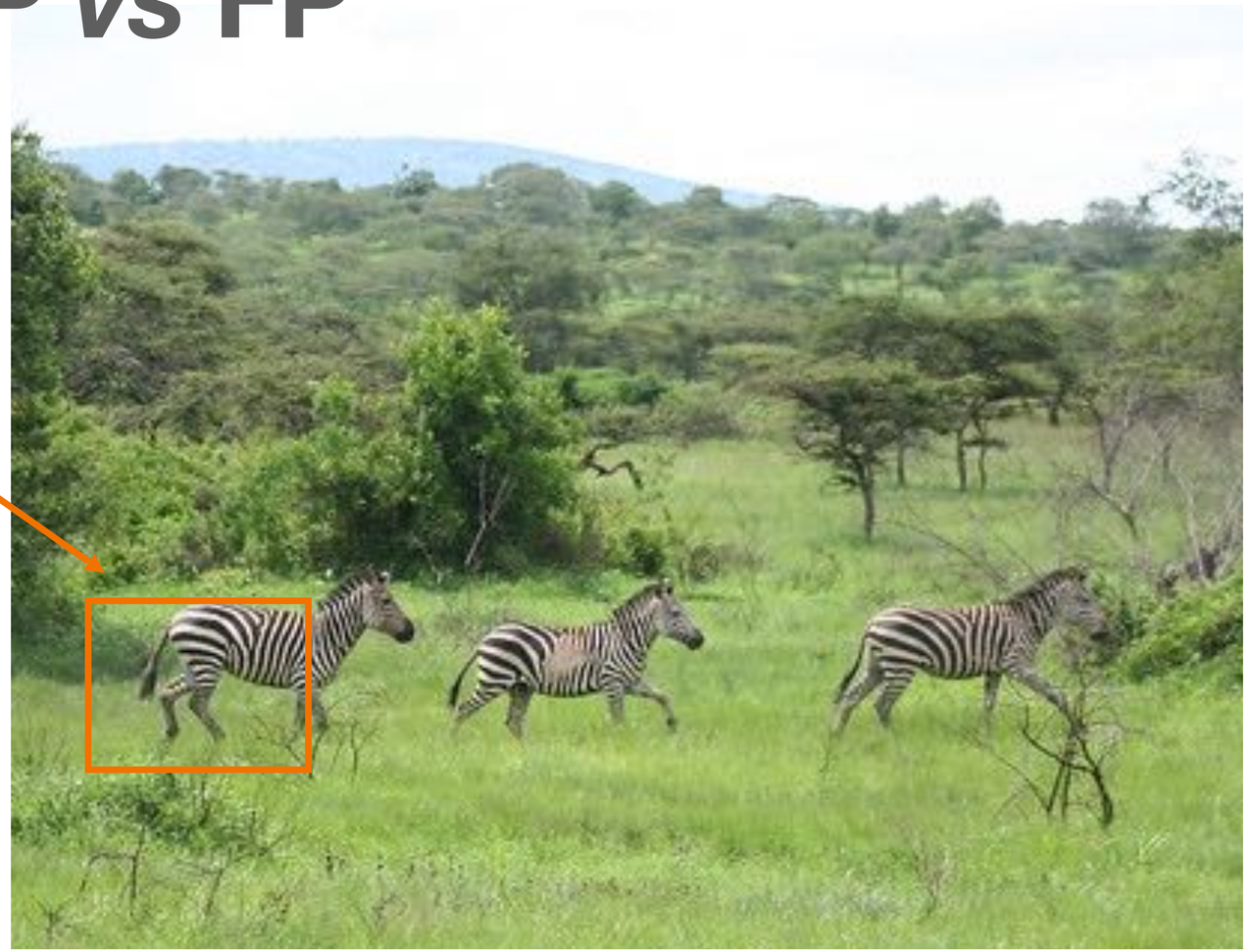
Counting TP *vs* FP

TP = True positive?



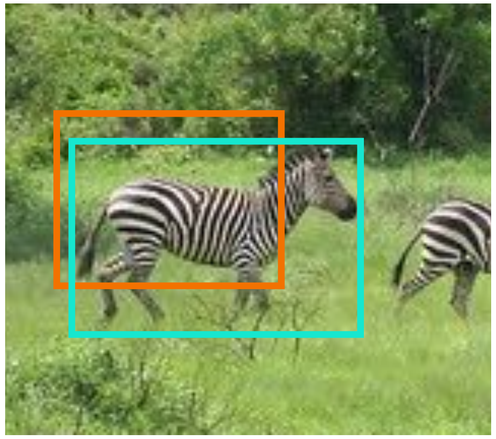
Counting TP *vs* FP

TP = True positive?



Counting TP *vs* FP

- What is a true positive?
  - Use the Intersection over Union (IoU)
  - e.g. if IoU $> 0.5 \rightarrow$ positive match
- The criterion is defined by the benchmarks (MS-COCO, Pascal VOC)

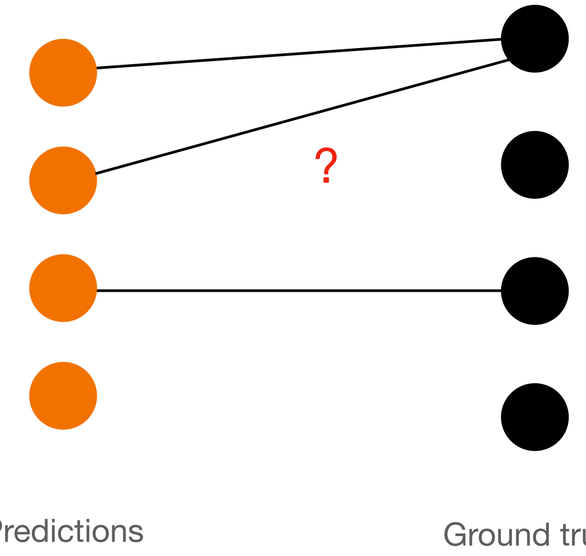


if

eger kesisim / union 0.5 ten buyukse

Resolving conflicts

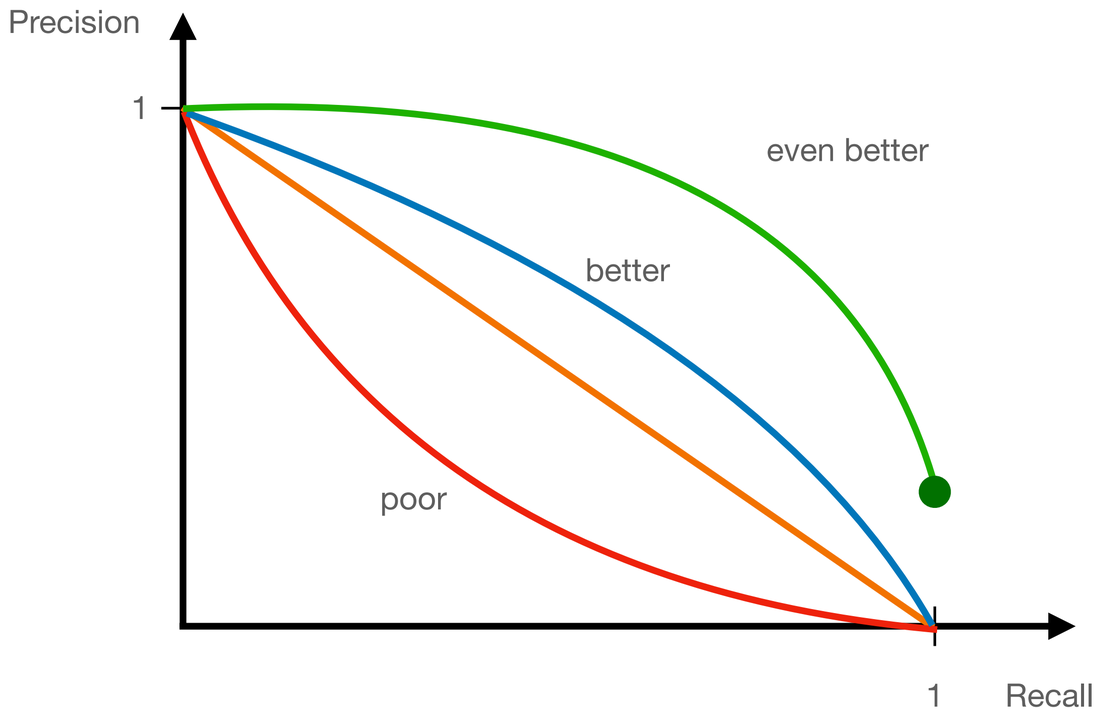
- Each prediction can match at most 1 ground-truth box
- Each ground-truth box can match at most 1 prediction



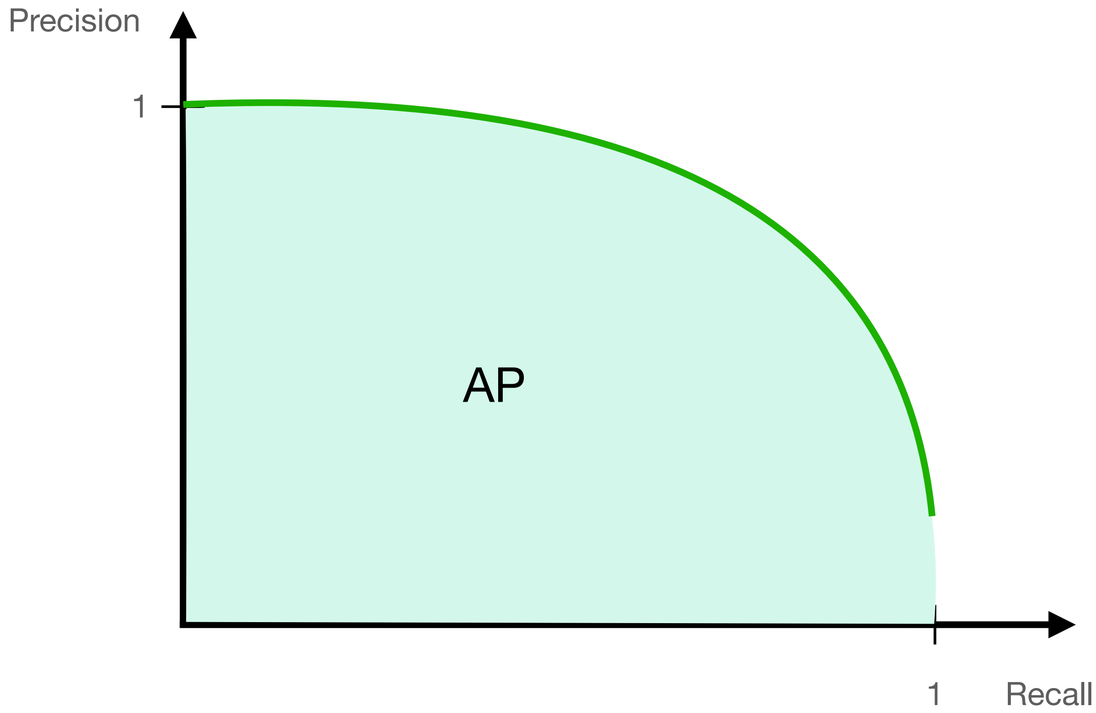
Select the most confident prediction first

All-in-one metric: Average Precision (AP)

- There is often a trade-off between Precision and Recall



Average Precision (AP)



- AP = the area under the Precision-Recall curve (AUC)
- How to fix the a particular Recall value?

so we can take a fixed recall like 0.5 and calculate mean of precision

Leverage confidence of model predictions

Computing Average Precision

1. Rank the predicted boxes by confidence score
2. For each prediction find the associated ground truth
   - The ground-truth should be unassigned and pass IoU threshold
3. Compute cumulative TP and FP
   - TP: [1,0,1] FP: [0,1,0] → cTP = [1,1,2], cFP = [0,1,1]

in here we cumulate true positives 1 = 1 , 1+0 = 1 , 1+1 = 2 , toplayarak gidiyoruz yani

in order true positive , false positive, true positive

4. Compute Precision and Recall (#GT = 3 → FN + TP = 3)
   - Precision: [1/1, 1/2, 2/3]; Recall (#GT = 3): [1/3,1/3,2/3]
5. Find (max) Precision such that Recall is at least $R \in \{0, 0.1, \ldots, 1\}$

Average Precision: Example

Rank	True/false	Precision	Recall
1	T		
2	T		
3	F		
4	T		
5	F		
6	F		
7	T		
8	F		
9	T		
10	T		
11	F		
12	F		

My method predicts 12 boxes;
There are 6 objects (ground-truth)

Average Precision: Example

Rank	True/false	Precision	Recall
1	T	1.0	0.17
2	T		
3	F		
4	T		
5	F		
6	F		
7	T		
8	F		
9	T		
10	T		
11	F		
12	F		

My method predicts 12 boxes;
There are 6 objects (ground-truth)

Average Precision: Example

Rank	True/false	Precision	Recall
1	T	1.0	0.17
2	T	1.0	0.33
3	F		
4	T		
5	F		
6	F		
7	T		
8	F		
9	T		
10	T		
11	F		
12	F		

My method predicts 12 boxes;
There are 6 objects (ground-truth)

Average Precision: Example

Rank	True/false	Precision	Recall
1	T	1.0	0.17
2	T	1.0	0.33
3	F	0.67	0.33
4	T		
5	F		
6	F		
7	T		
8	F		
9	T		
10	T		
11	F		
12	F		

My method predicts 12 boxes;
There are 6 objects (ground-truth)

Average Precision: Example

Rank	True/false	Precision	Recall
1	T	1.0	0.17
2	T	1.0	0.33
3	F	0.67	0.33
4	T	0.75	0.5
5	F	0.6	0.5
6	F	0.5	0.5
7	T	0.57	0.67
8	F	0.5	0.67
9	T	0.56	0.83
10	T	0.6	1.0
11	F	0.55	1.0
12	F	0.5	1.0

My method predicts 12 boxes;
There are 6 objects (ground-truth)

Average Precision: Example

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

- Plot Precision vs Recall

1	2	3	4	5	6	7	8	9	10	11	12
T	T	F	T	F	F	T	F	T	T	F	F

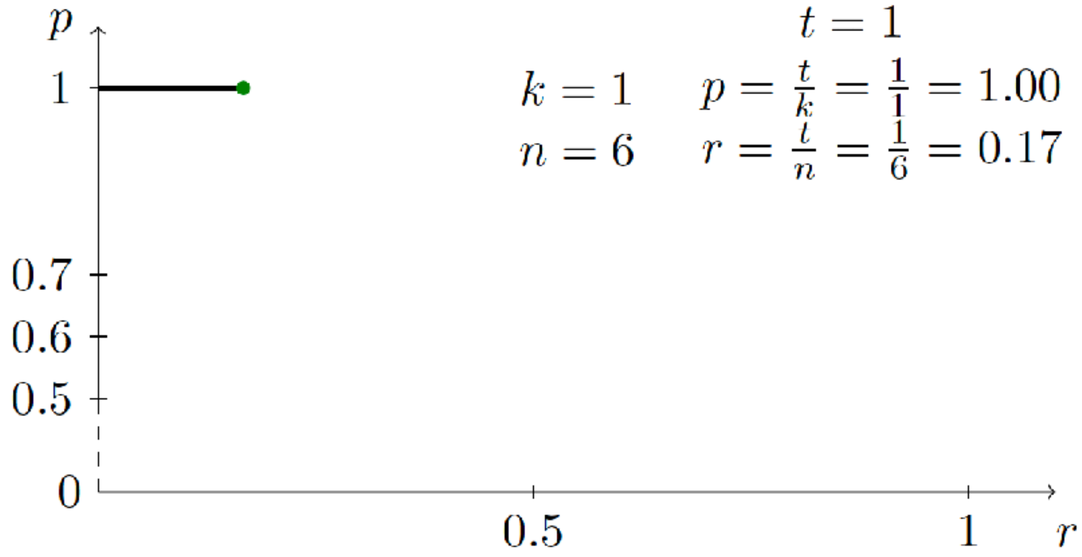


Image Credit: Y. Avrithis, Object detection lecture. INRIA.

Average Precision: Example

- Plot Precision vs Recall

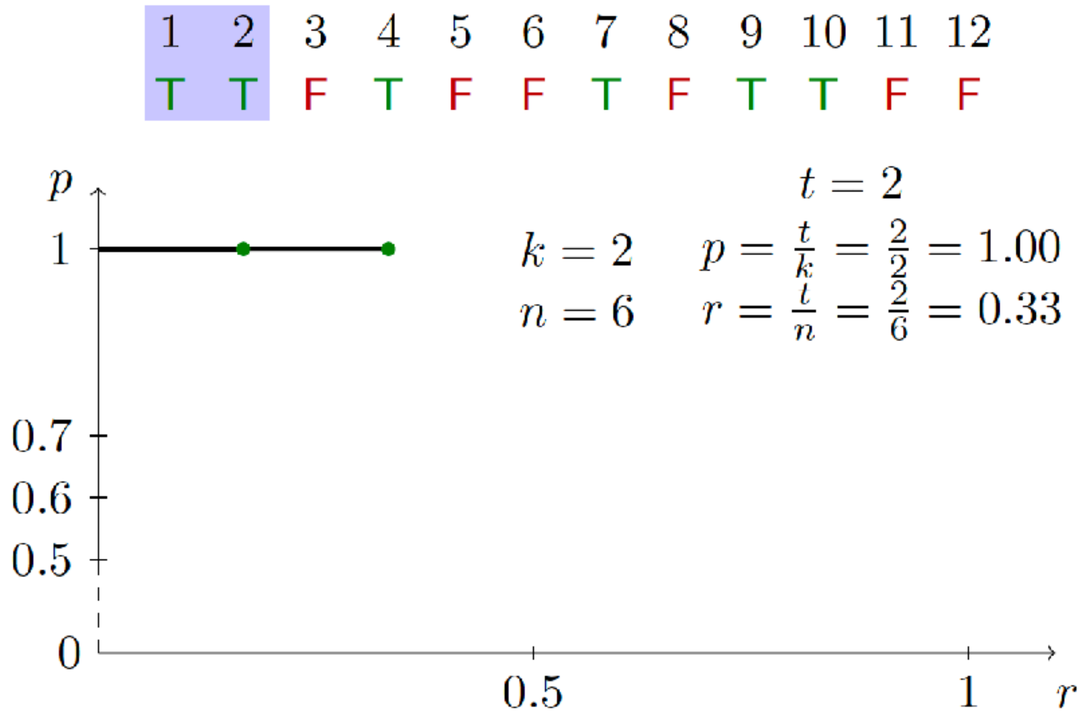


Image Credit: Y. Avrithis, Object detection lecture. INRIA.

Average Precision: Example

- Plot Precision vs Recall

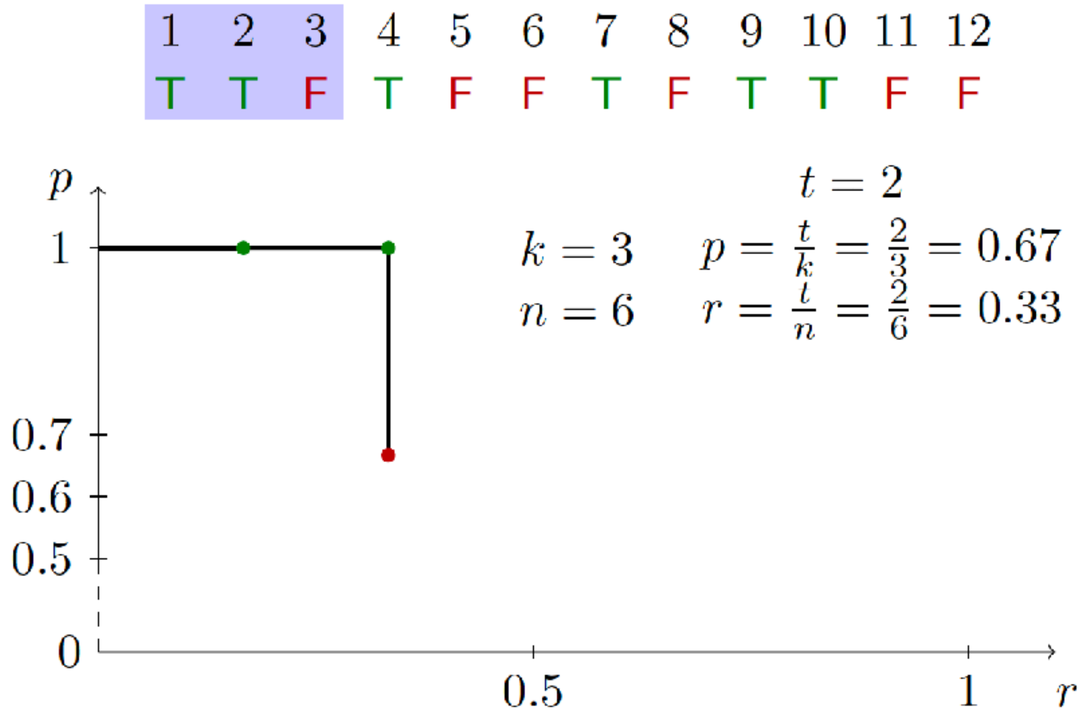


Image Credit: Y. Avrithis, Object detection lecture. INRIA.

Average Precision: Example

- Plot Precision vs Recall

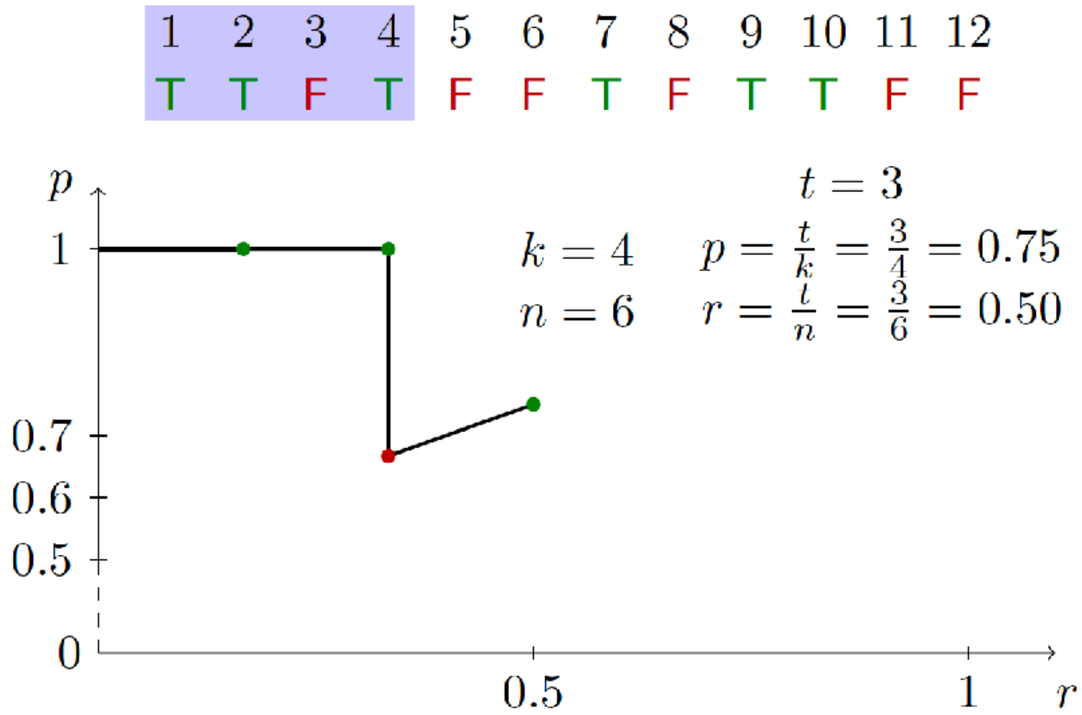


Image Credit: Y. Avrithis, Object detection lecture. INRIA.

Average Precision: Example

- Plot Precision vs Recall

1	2	3	4	5	6	7	8	9	10	11	12
T	T	F	T	F	F	T	F	T	T	F	F

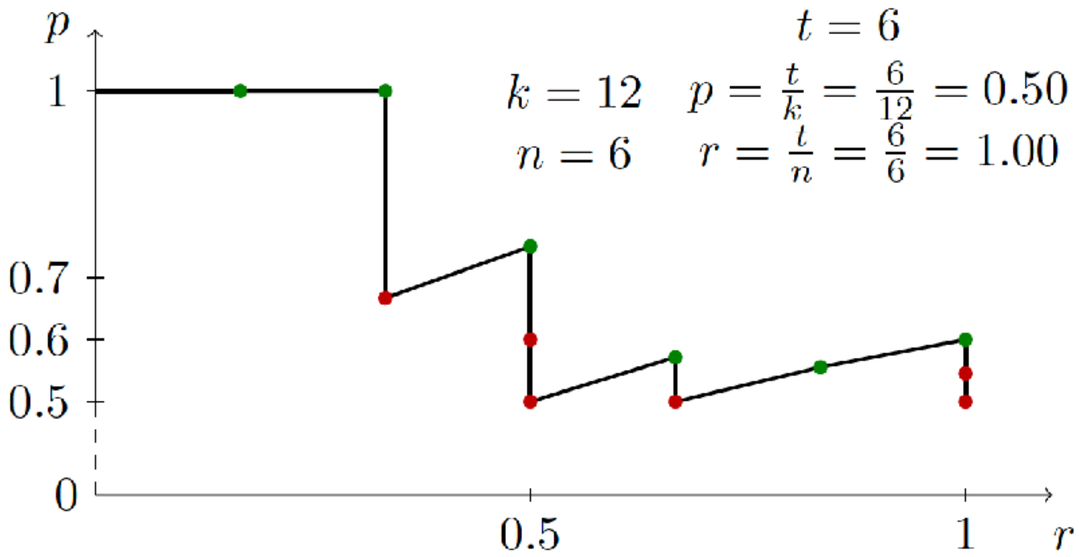


Image Credit: Y. Avrithis, Object detection lecture. INRIA.

Average Precision: Example

- Plot Precision vs Recall

1. $P = \frac{1}{1} = 1$

$r = \frac{1}{6} =$

1	2	3	4	5	6	7	8	9	10	11	12
T	T	F	T	F	F	T	F	T	T	F	F

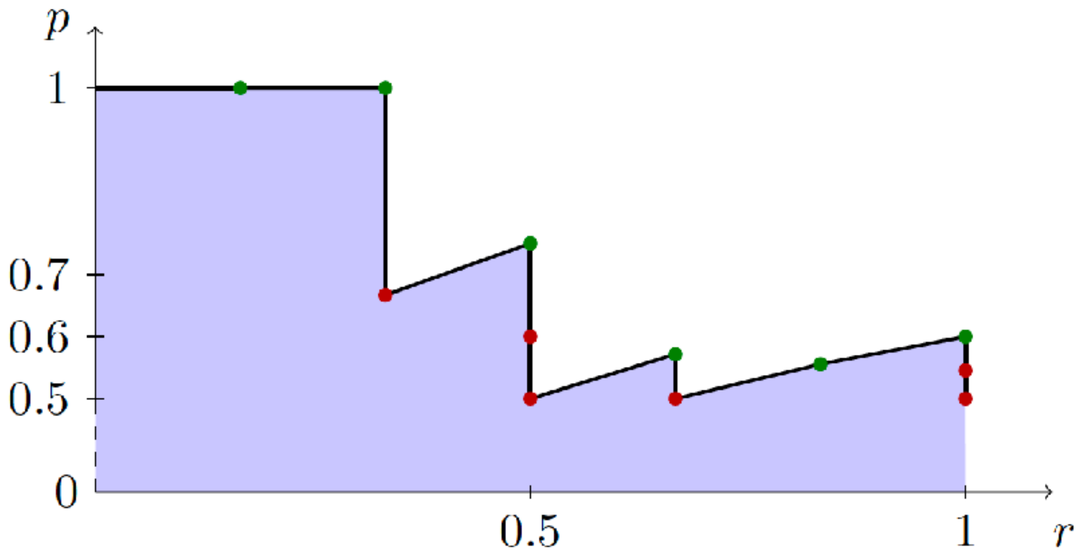


Image Credit: Y. Avrithis, Object detection lecture. INRIA.

Average Precision: Example

- Plot Precision vs Recall

1	2	3	4	5	6	7	8	9	10	11	12
T	T	F	T	F	F	T	F	T	T	F	F

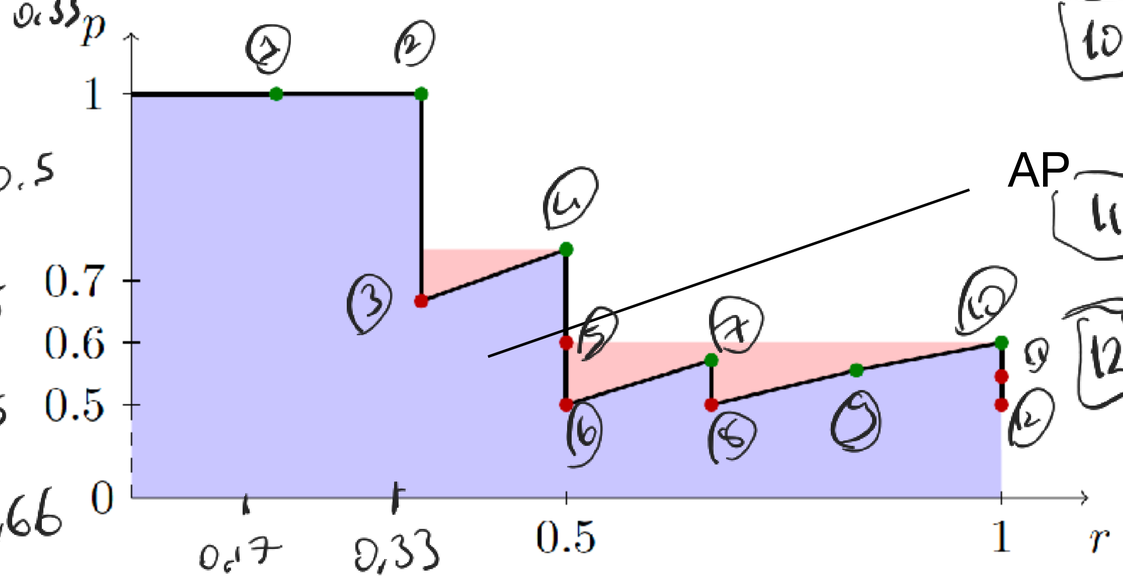


Image Credit: Y. Avrithis, Object detection lecture. INRIA.

AP derivatives

- AP may be averaged over multiple IoU thresholds
- mAP is the average over object categories
- Many other flavours:

MS-COCO

Average Precision (AP):

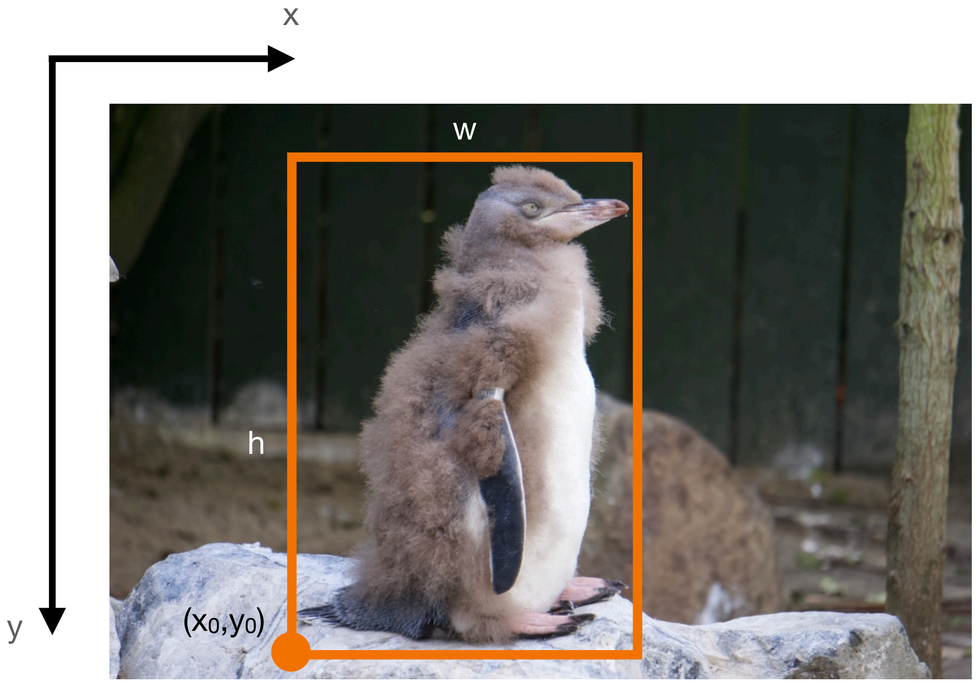
AP	% AP at IoU=.50:.05:.95 **(primary challenge metric)**
$AP^{IoU=.50}$	% AP at IoU=.50 (PASCAL VOC metric)
$AP^{IoU=.75}$	% AP at IoU=.75 (strict metric)

AP Across Scales:

AP^{small}	% AP for small objects: area < 32^2
AP^{medium}	% AP for medium objects: 32^2 < area < 96^2
AP^{large}	% AP for large objects: area > 96^2

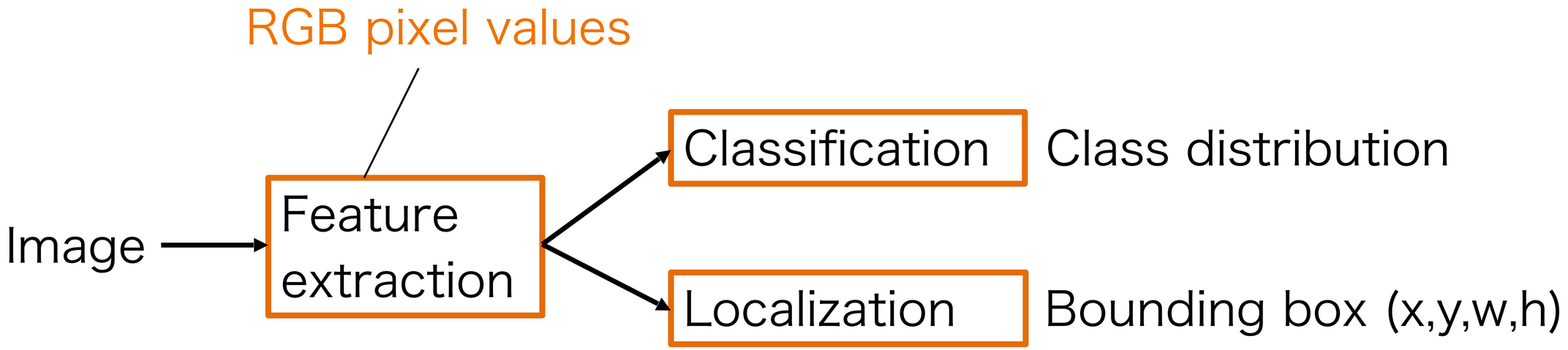
Lecture 01 recap

Object detection

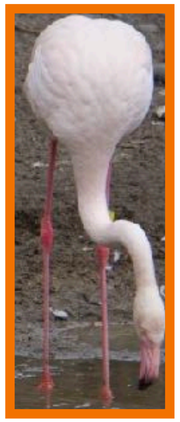


Types of object detectors

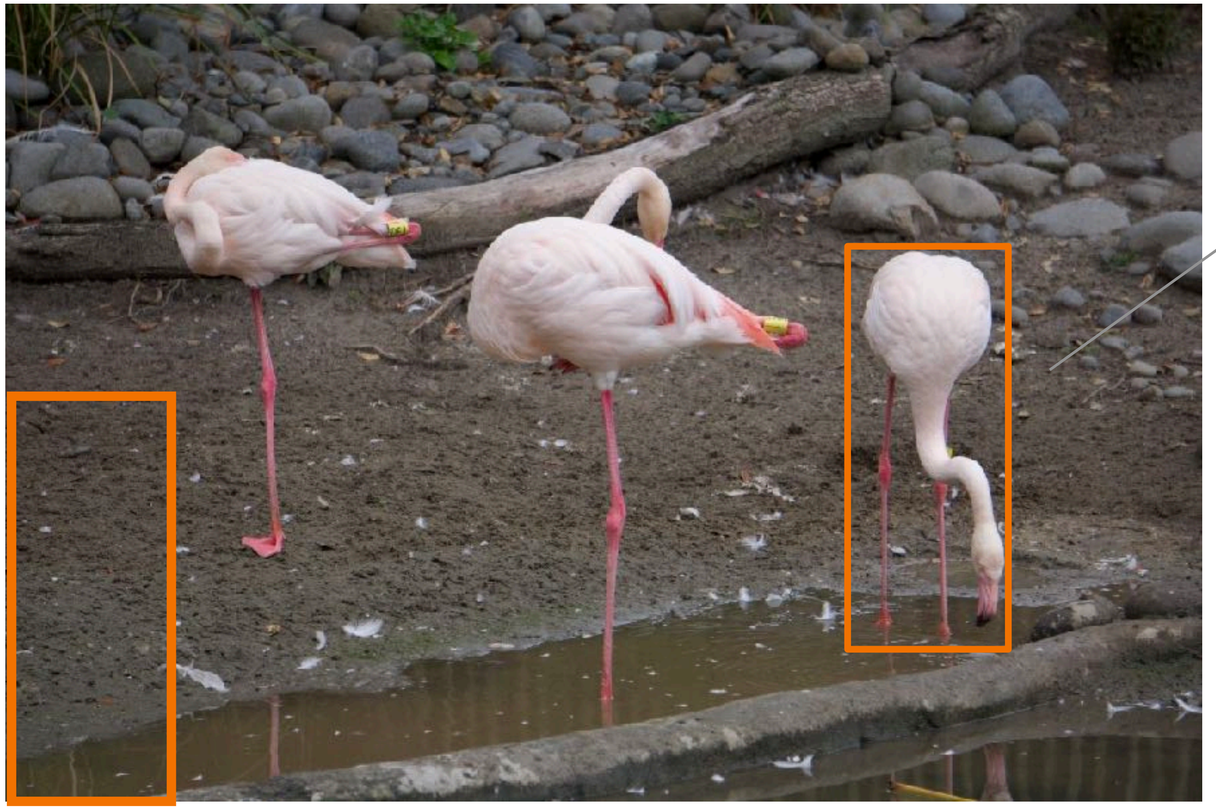
- One-stage detectors:

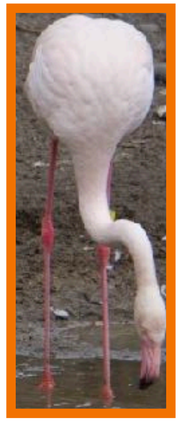


Template matching with sliding window



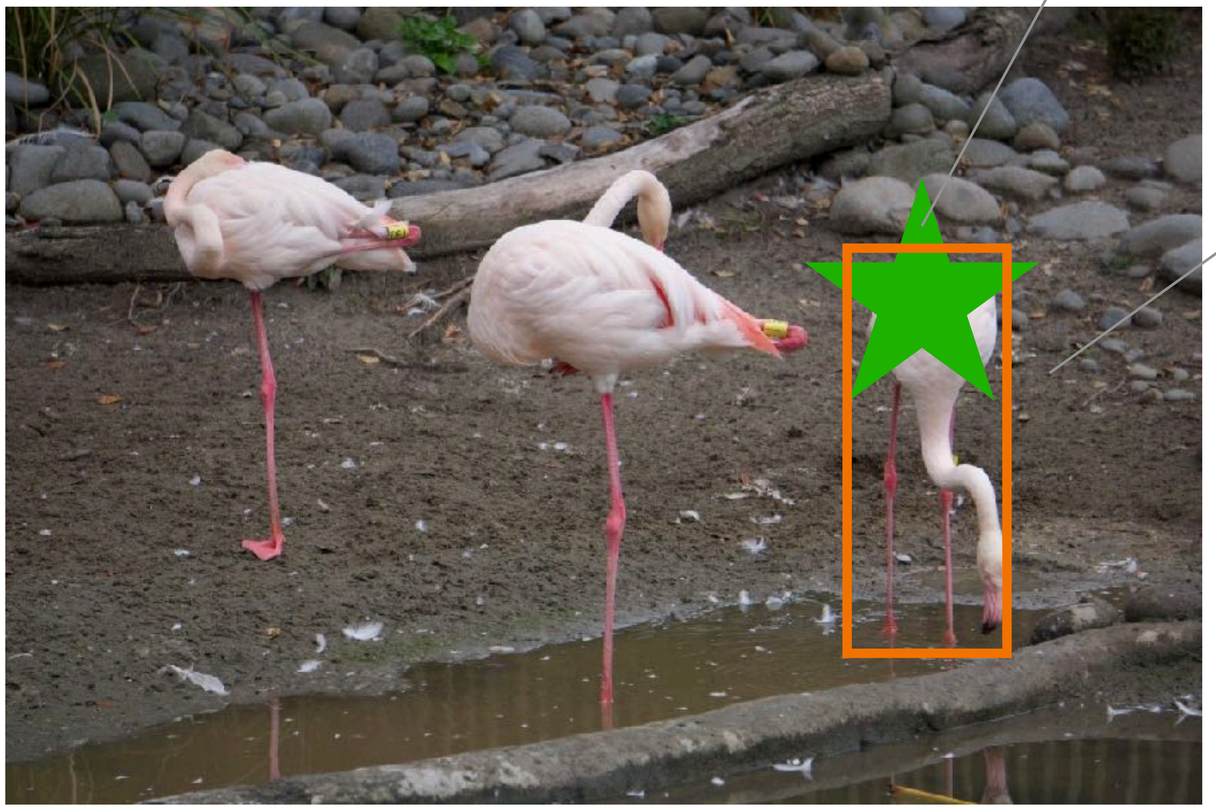
LOW correlation

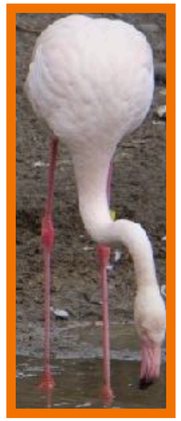




HIGH correlation

Template matching with sliding window





HIGH correlation

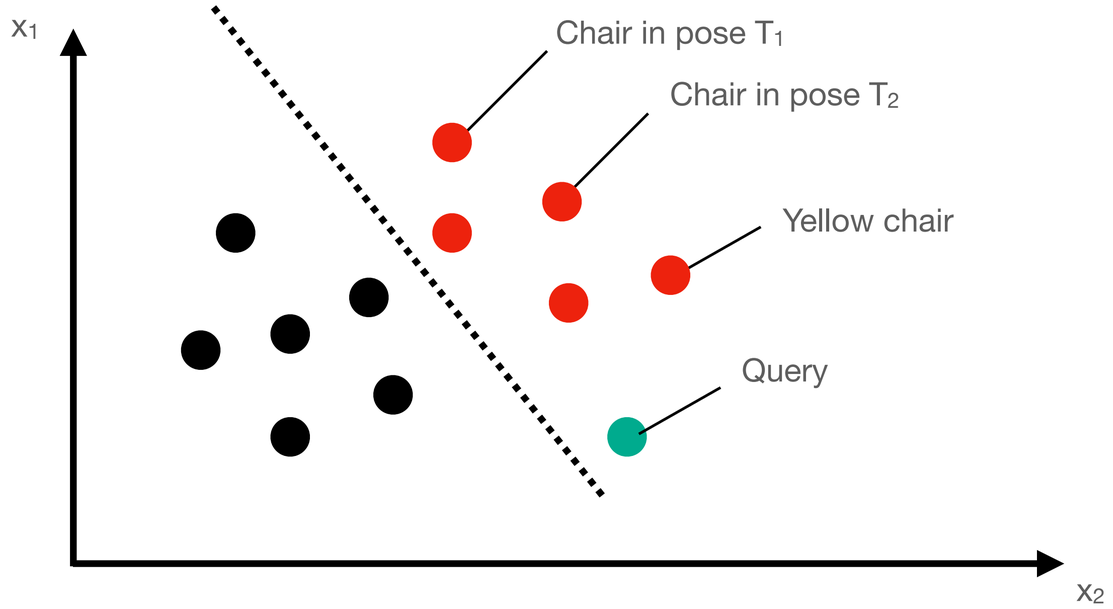
Template matching: disadvantages

- (Self-)occlusions (e.g. due to pose changes)

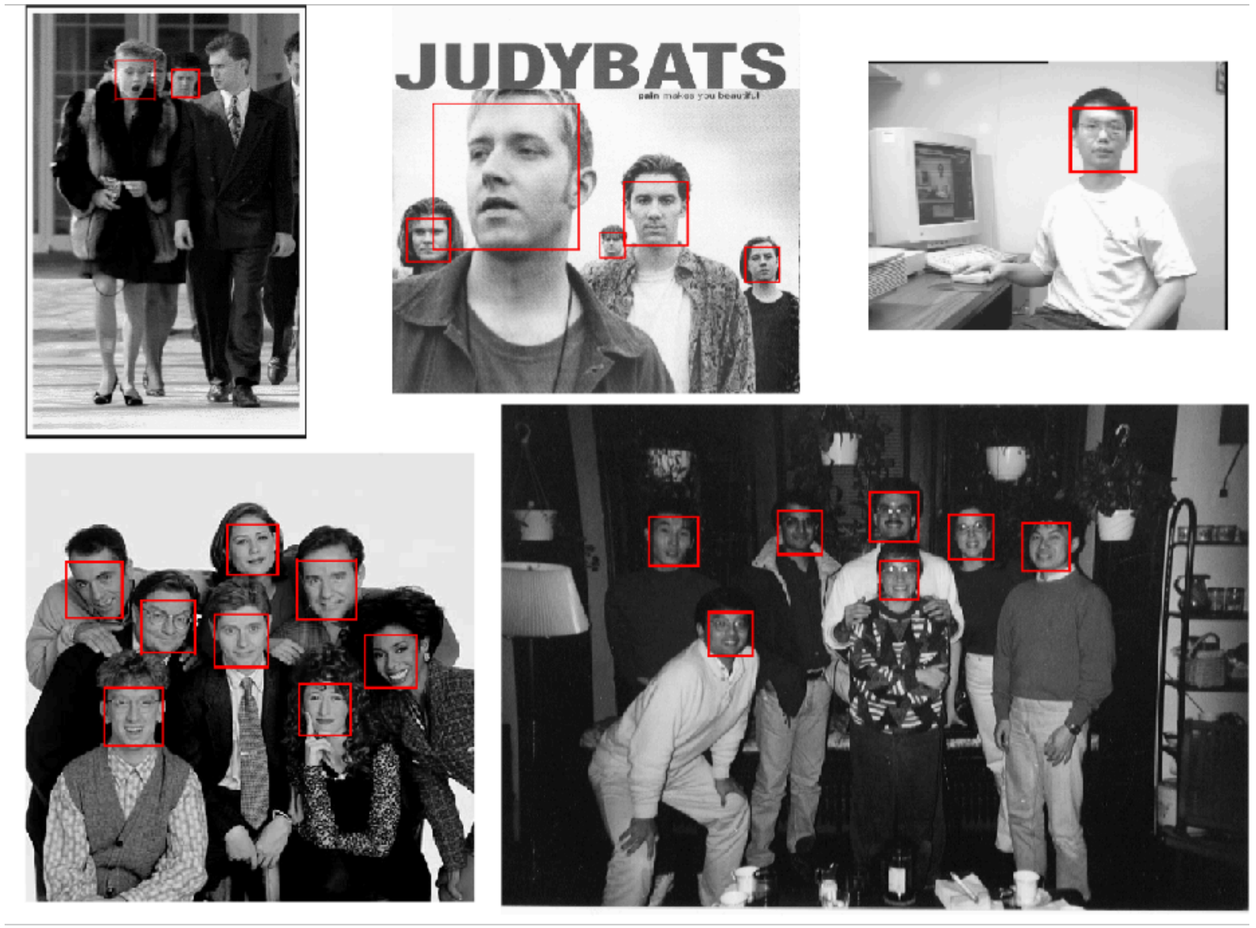
- Changes in appearance

Feature-based detection

Idea: Learn feature based classifiers invariant to natural object changes



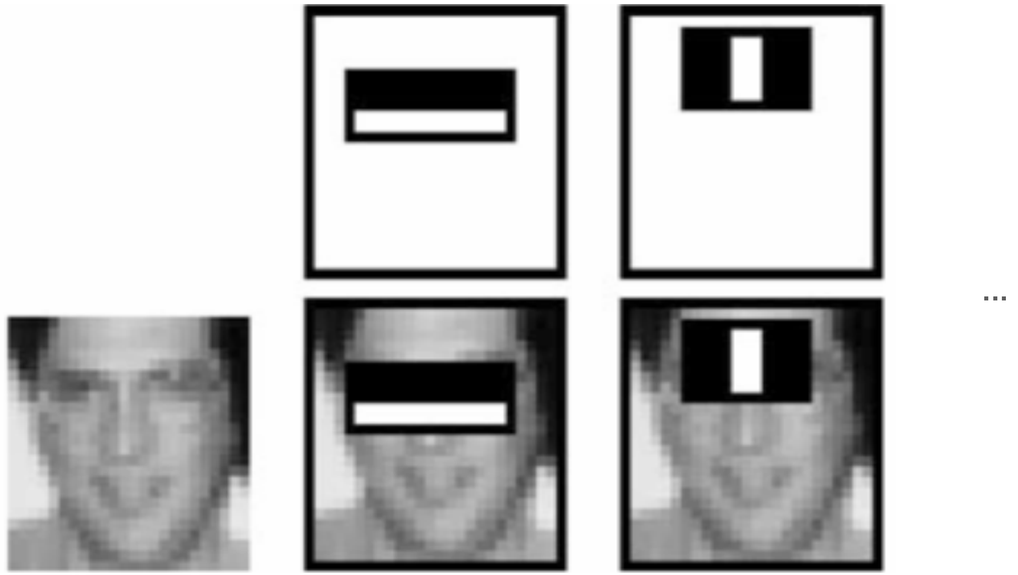
Viola-Jones detector



Viola and Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

Viola-Jones detector

Haar-like features



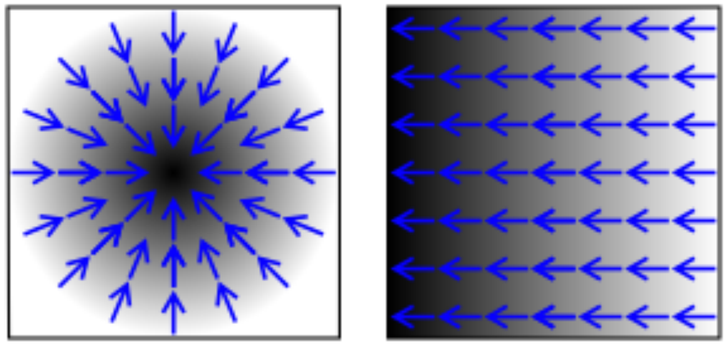
...

$h_1(\cdot)$ $h_2(\cdot)$

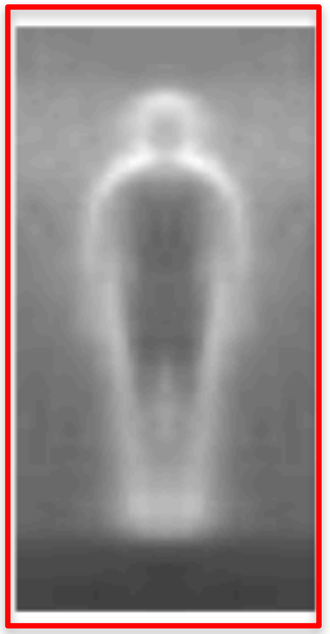
“weak” learners

Viola and Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

Histogram of Oriented Gradients



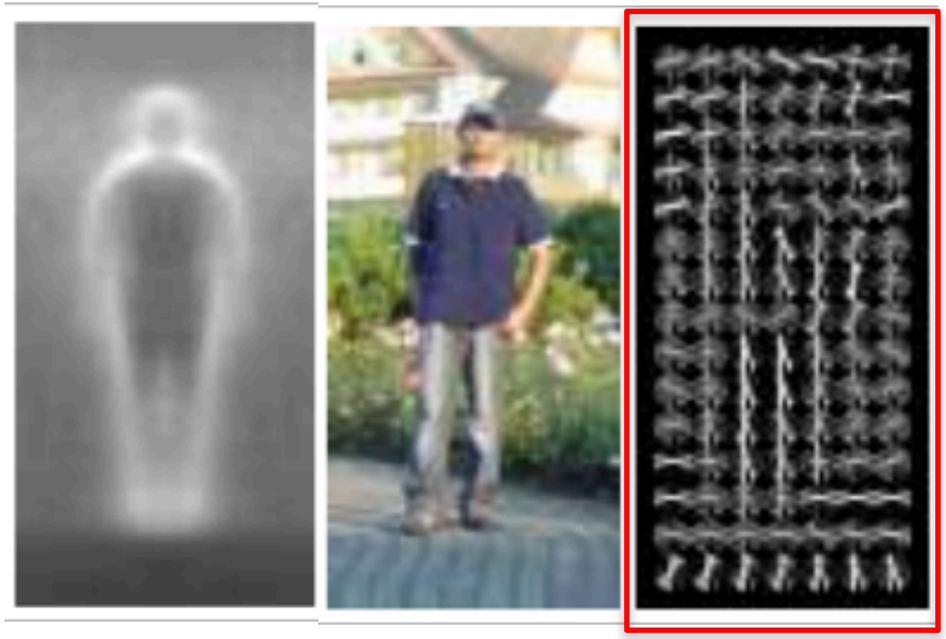
Gradient: blue arrows show the gradient, i.e., the direction of greatest change of the image.



Average gradient image over training samples → gradients provide shape information.

Dalal and Triggs. Histogram of oriented gradients for human detection. CVPR 2005.

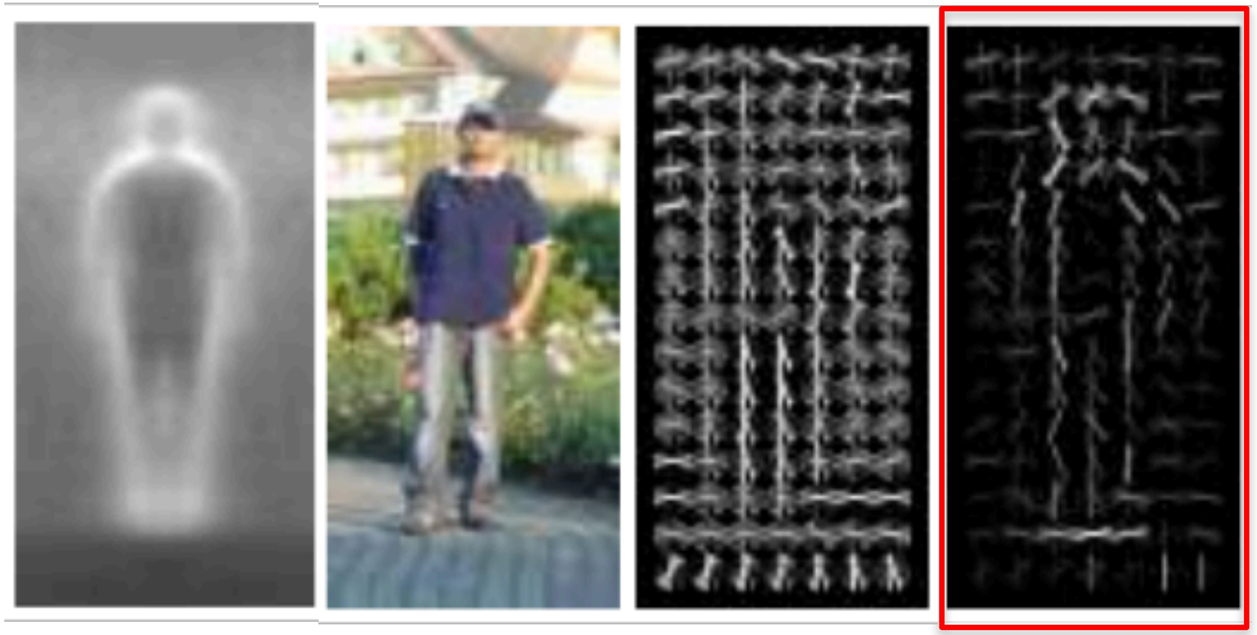
Histogram of Oriented Gradients



HOG descriptor → Histogram of oriented gradients.

Dalal and Triggs. Histogram of oriented gradients for human detection. CVPR 2005.

Histogram of Oriented Gradients

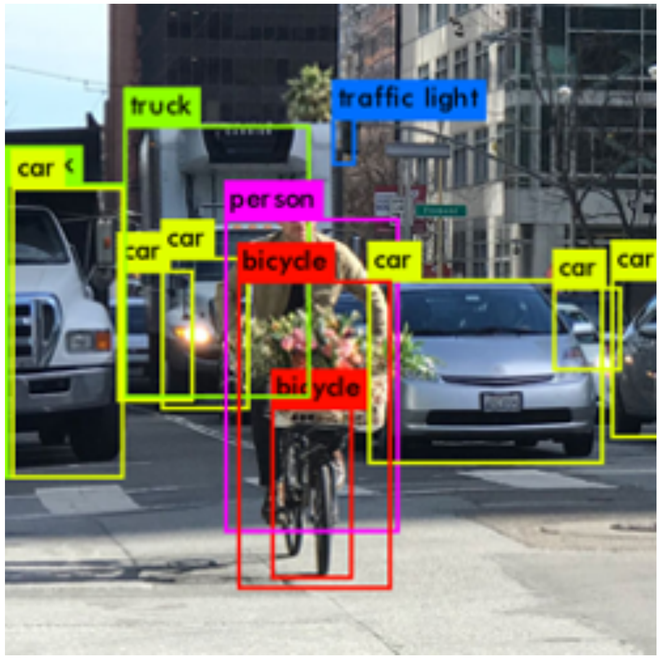


HOG features weighted by the positive SVM weights – the ones used for the pedestrian object classifier.

Dalal and Triggs. Histogram of oriented gradients for human detection. CVPR 2005.

Towards real-world object detection

Can we design features that work everywhere?



Probably not, but we can learn them from data.

↓

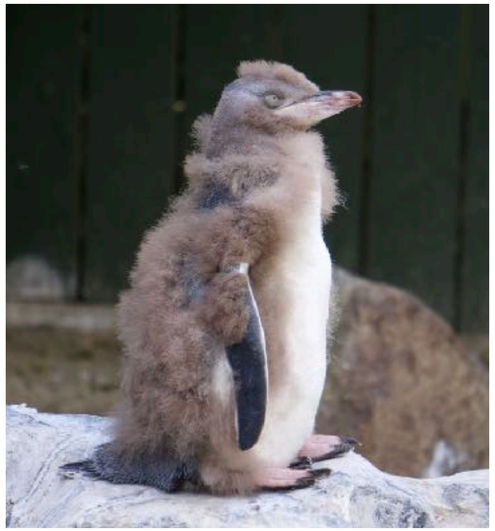
Deep Learning

Object detection with deep networks

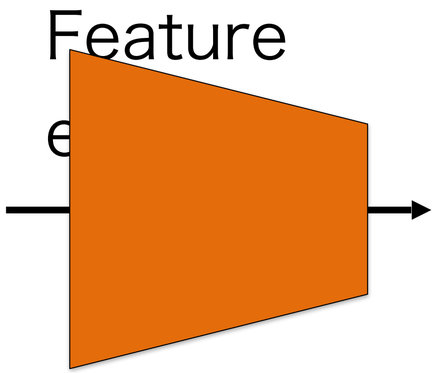
08.11.2022 | Nikita Araslanov

Localisation

- Bounding box regression



Image



CNN

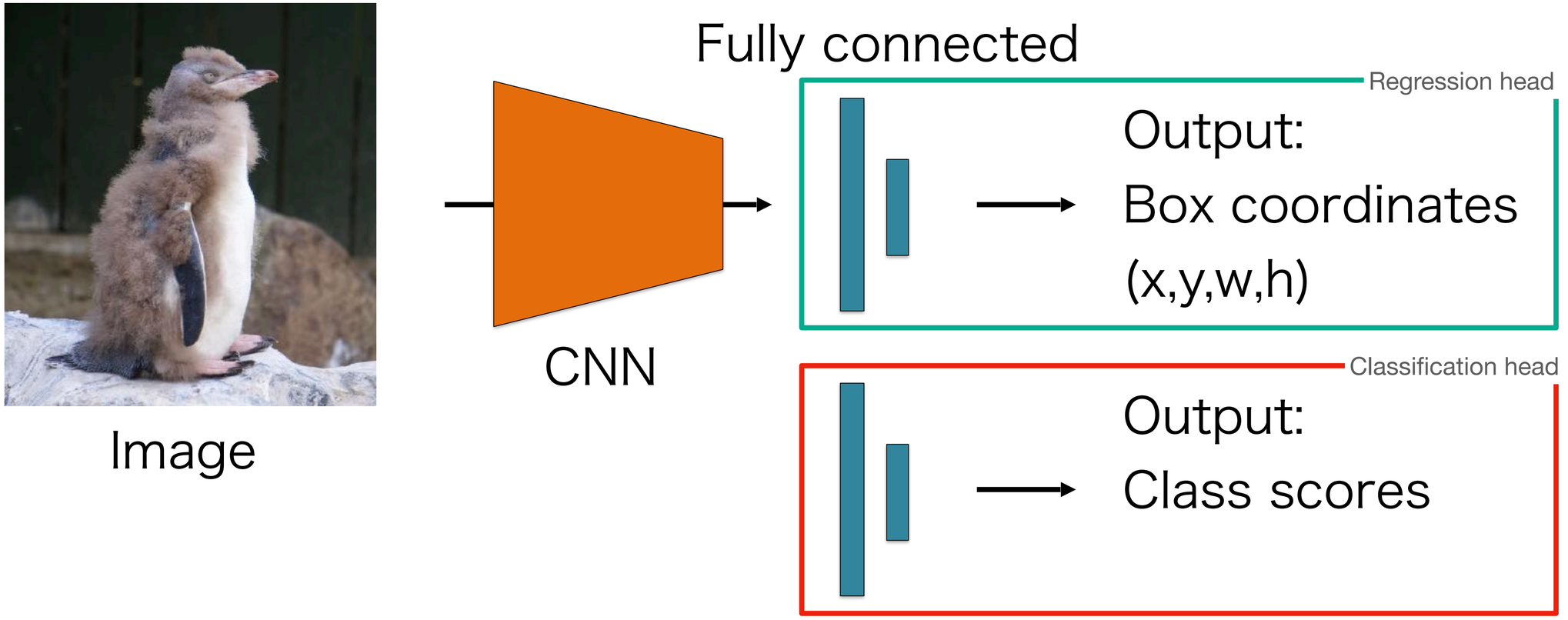
Box coordinates
(x,y,w,h)

L2 loss
function

Ground truth:
Box coordinates

Localisation and classification

- Bounding box regression



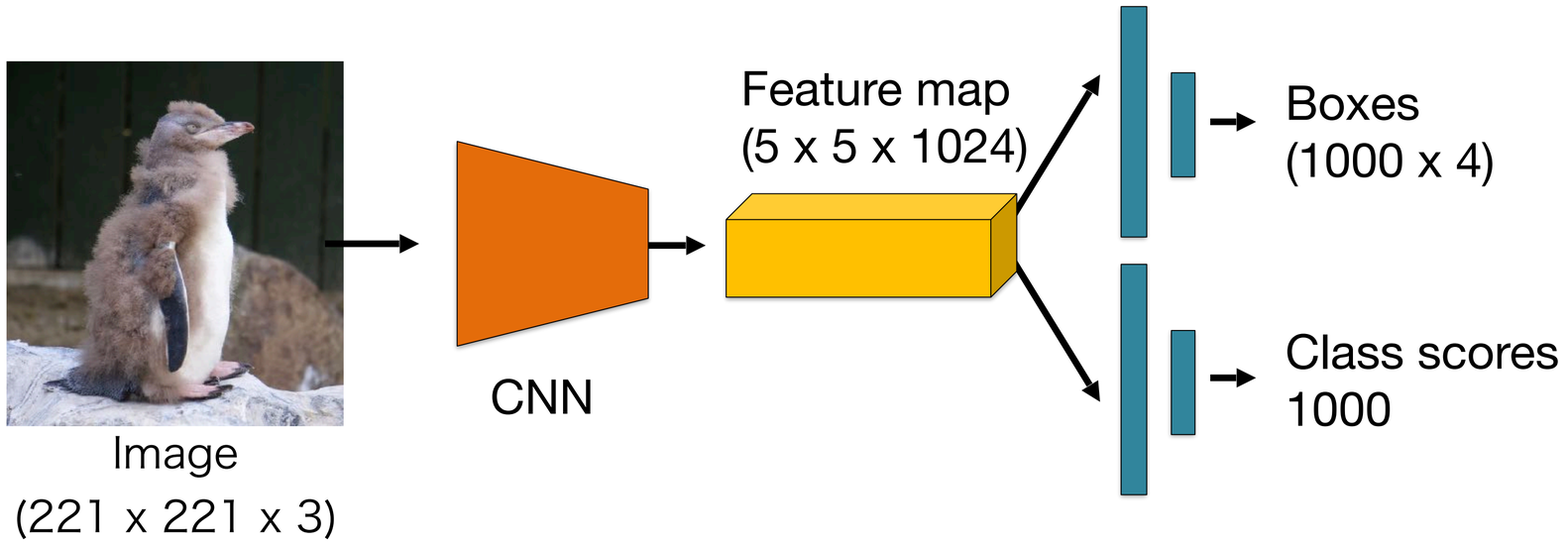
Overfeat

- Train the classification head first, freeze the layers;
- Then train the regression head.
- At test time, we use both.

Sermanet et al, “Overfeat: Integrated Recognition, Localization and Detection using Convolutional Networks”, ICLR 2014

Overfeat

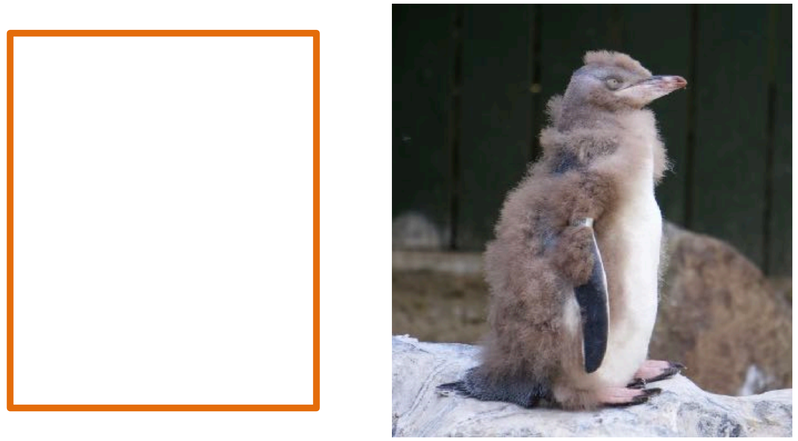
- Sliding window + box regression + classification



Sermanet et al, “Overfeat: Integrated Recognition, Localization and Detection using Convolutional Networks”, ICLR 2014

Overfeat

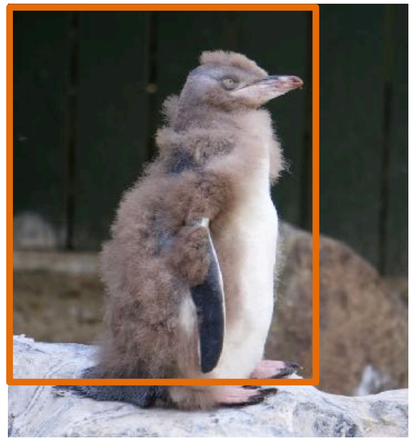
- Sliding window + box regression + classification



Sermanet et al, “Overfeat: Integrated Recognition, Localization and Detection using Convolutional Networks”, ICLR 2014

Overfeat

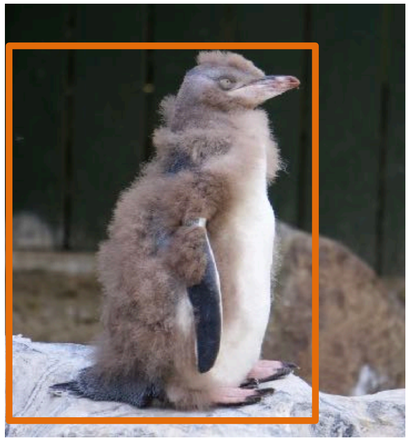
- Sliding window + box regression + classification



Sermanet et al, “Overfeat: Integrated Recognition, Localization and Detection using Convolutional Networks”, ICLR 2014

Overfeat

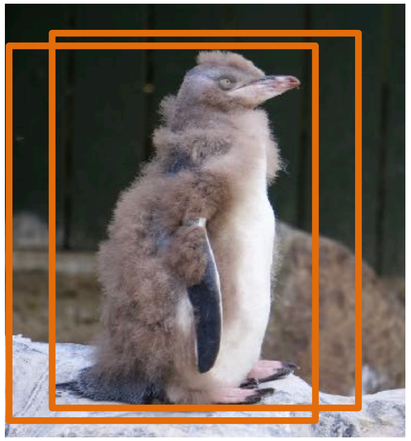
- Sliding window + box regression + classification



Sermanet et al, “Overfeat: Integrated Recognition, Localization and Detection using Convolutional Networks”, ICLR 2014

Overfeat

- Sliding window + box regression + classification



Sermanet et al, “Overfeat: Integrated Recognition, Localization and Detection using Convolutional Networks”, ICLR 2014

Overfeat

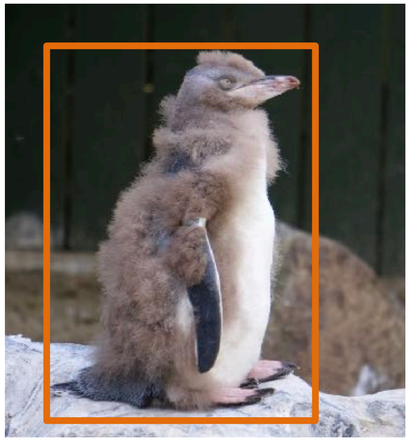
- Sliding window + box regression + classification

We end up with many predictions and we have to combine them for a final detection

Sermanet et al, “Overfeat: Integrated Recognition, Localization and Detection using Convolutional Networks”, ICLR 2014

Overfeat

- Sliding window + box regression + classification



We end up with many predictions and we have to combine them for a final detection

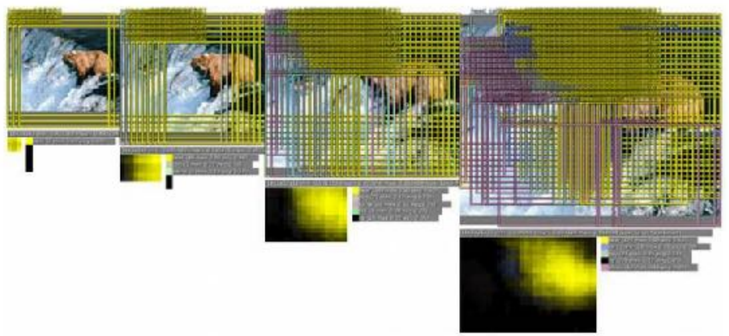
e.g. with non-max supression

Sermanet et al, “Overfeat: Integrated Recognition, Localization and Detection using Convolutional Networks”, ICLR 2014

Overfeat

- In practice: use many sliding window locations and multiple scales

Window positions + score maps



Box regression outputs

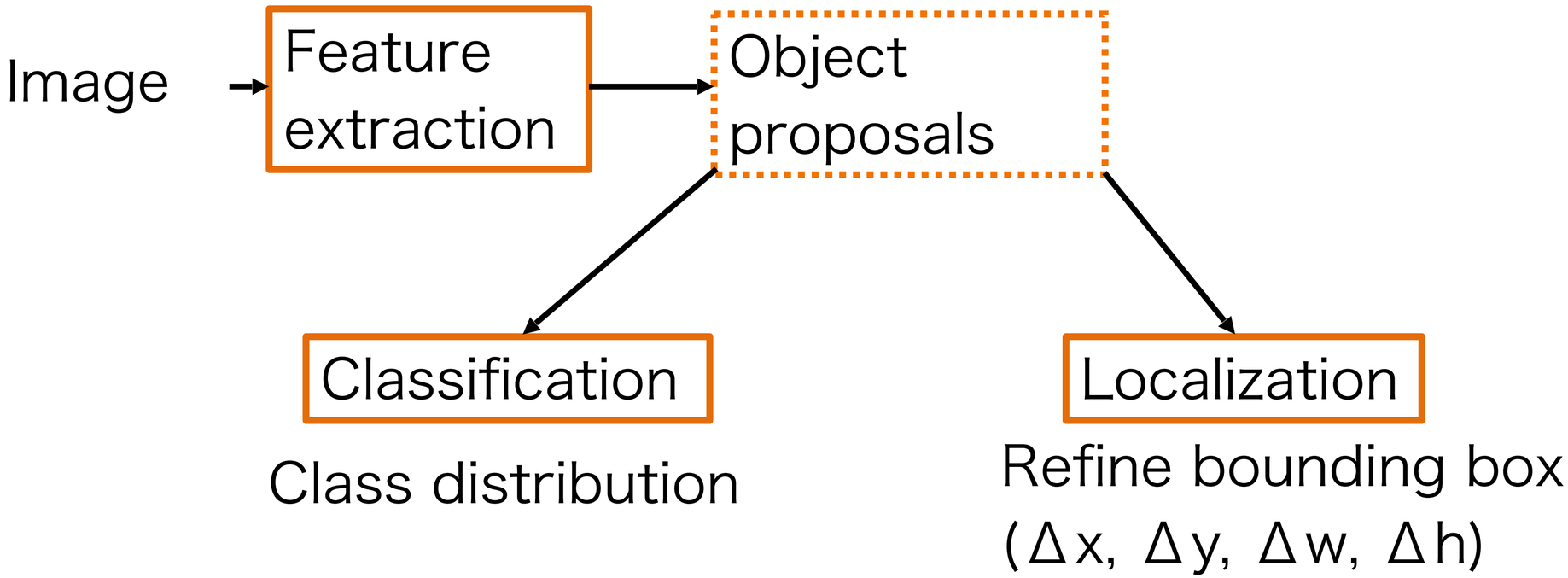
Final Prediction

Sermanet et al, “Overfeat: Integrated Recognition, Localization and Detection using Convolutional Networks”, ICLR 2014

Overfeat

- Cons:
  - Expensive to try all possible positions, scales and aspect ratios
  - Network works on a fixed input

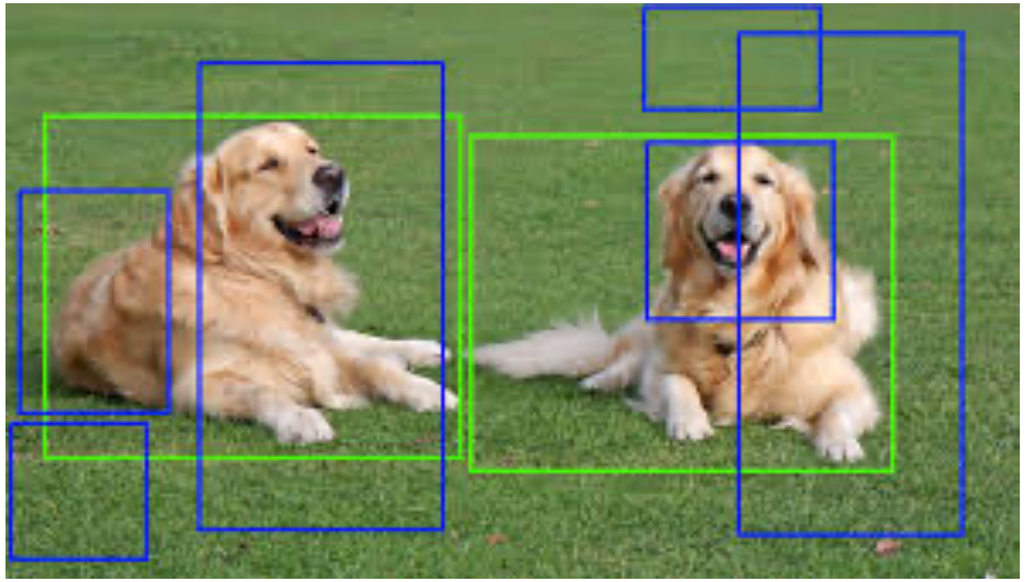
Two-stage detectors



Region proposals

- We heuristic-based methods that gives us “interesting” regions in an image

1. Obtain region proposals.
2. Classify & refine them.

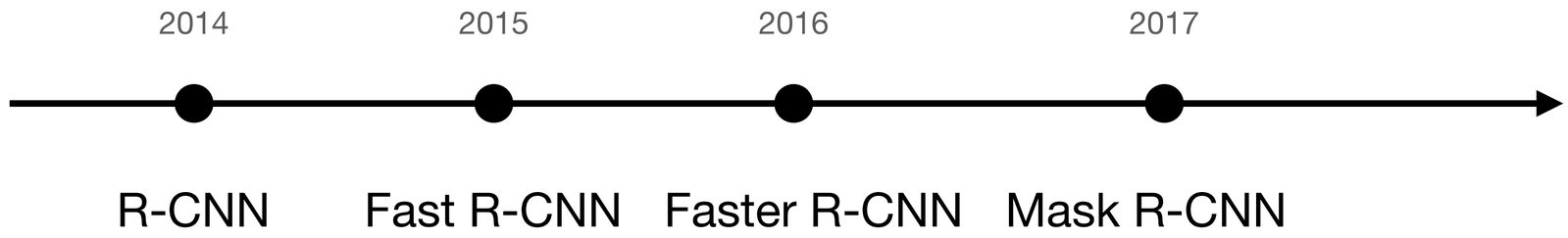


Uijlings et al. Selective Search for Object Recognition. IJCV 2013

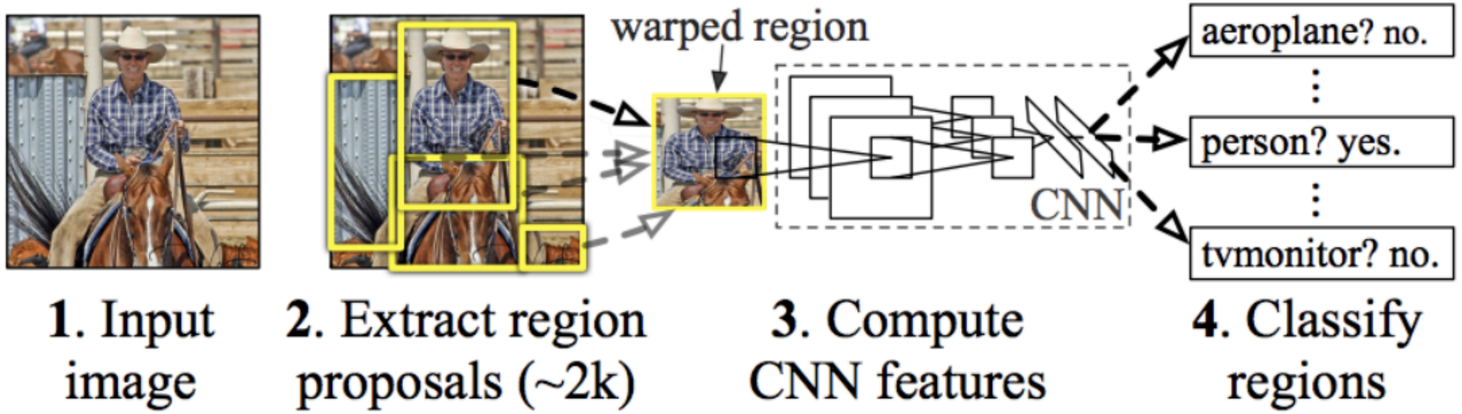
R-CNN family

R-CNN family

- “Regions with CNN features”
- One of the most impactful lines of work on multi-object detection and segmentation
- … and in CV
  - >120K total citations: R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN



R-CNN

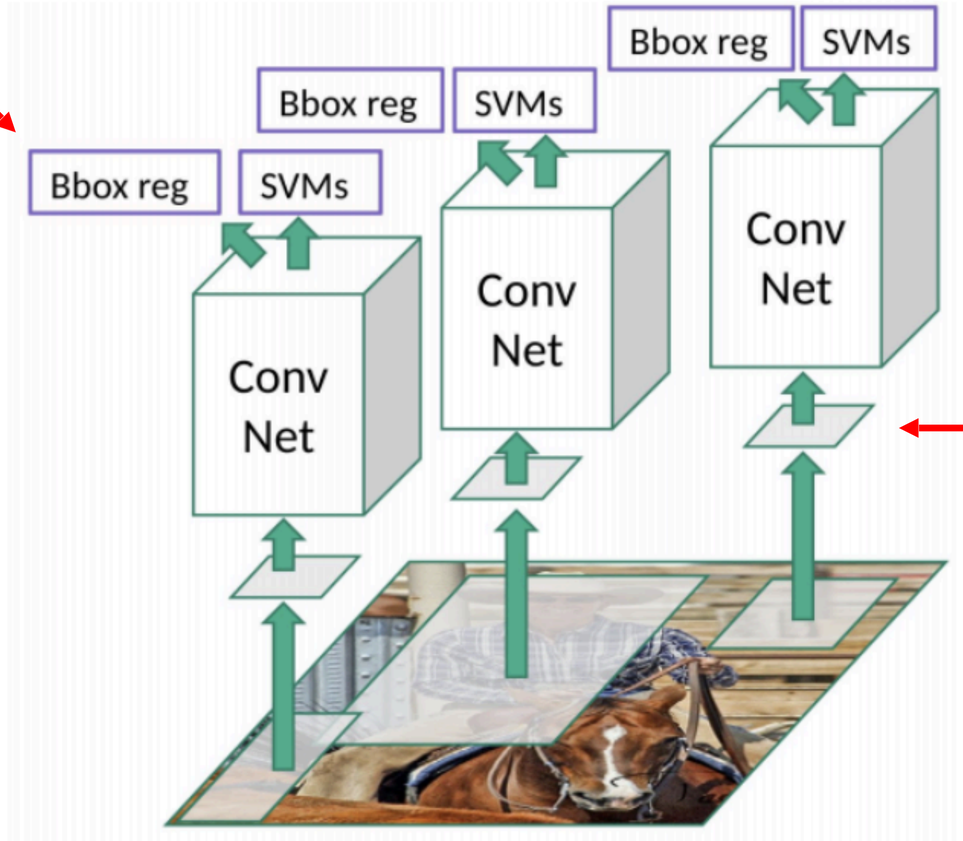


Girschick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014

R-CNN

Regression head to refine the bounding box location

Classification head



Extract features

Warping to a fix size 227 x 227

R-CNN

- Training scheme:
  1. Pre-train the CNN for image classification (ImageNet)
  2. Finetune the CNN on the number of classes the detector is aiming to classify
  3. Train a linear Support Vector Machine classifier to classify image regions. One SVM per class!
  4. Train the bounding box regressor

R-CNN

- Pros:
  - New: CNN features; the overall pipeline with proposals is heavily engineered → good accuracy.
  - CNN summarizes each proposal into a 4096 vector (much more compact representation compared to HOG)
  - Leverage transfer learning: The CNN can be pre-trained for image classification with C classes. One needs only to change the FC layers to deal with Z classes.

R-CNN

- Cons:

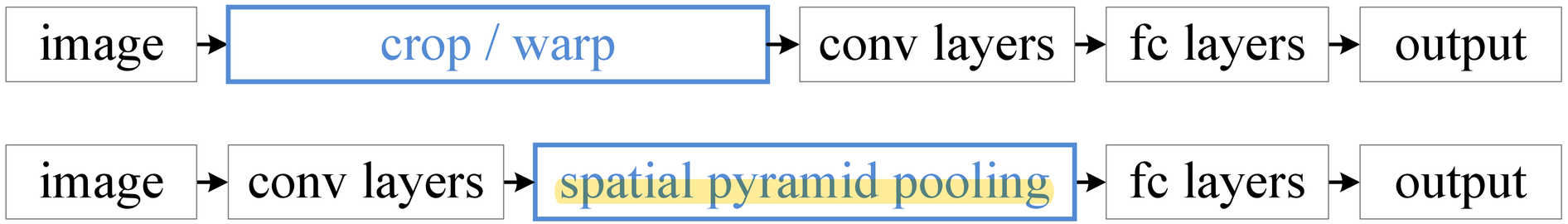
Let us try to solve this first

  - Slow: 47s/image with VGG16 backbone. One considers around 2000 proposals per image, they need to be warped and forwarded through the CNN.
  - Training is also slow and complex
  - The object proposal algorithm is fixed.
  - Feature extraction and SVM classifier are trained separately – features are not learned “end-to-end”

SPP-Net: Spatial Pyramid Pooling

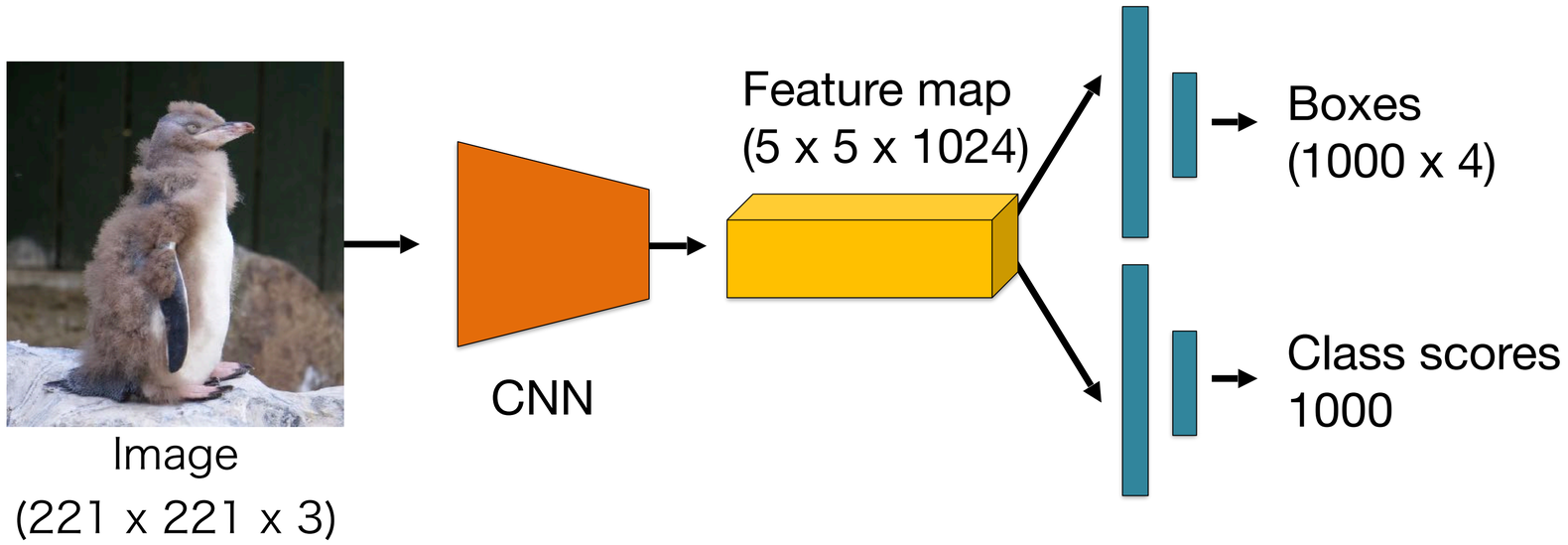
crop

warp



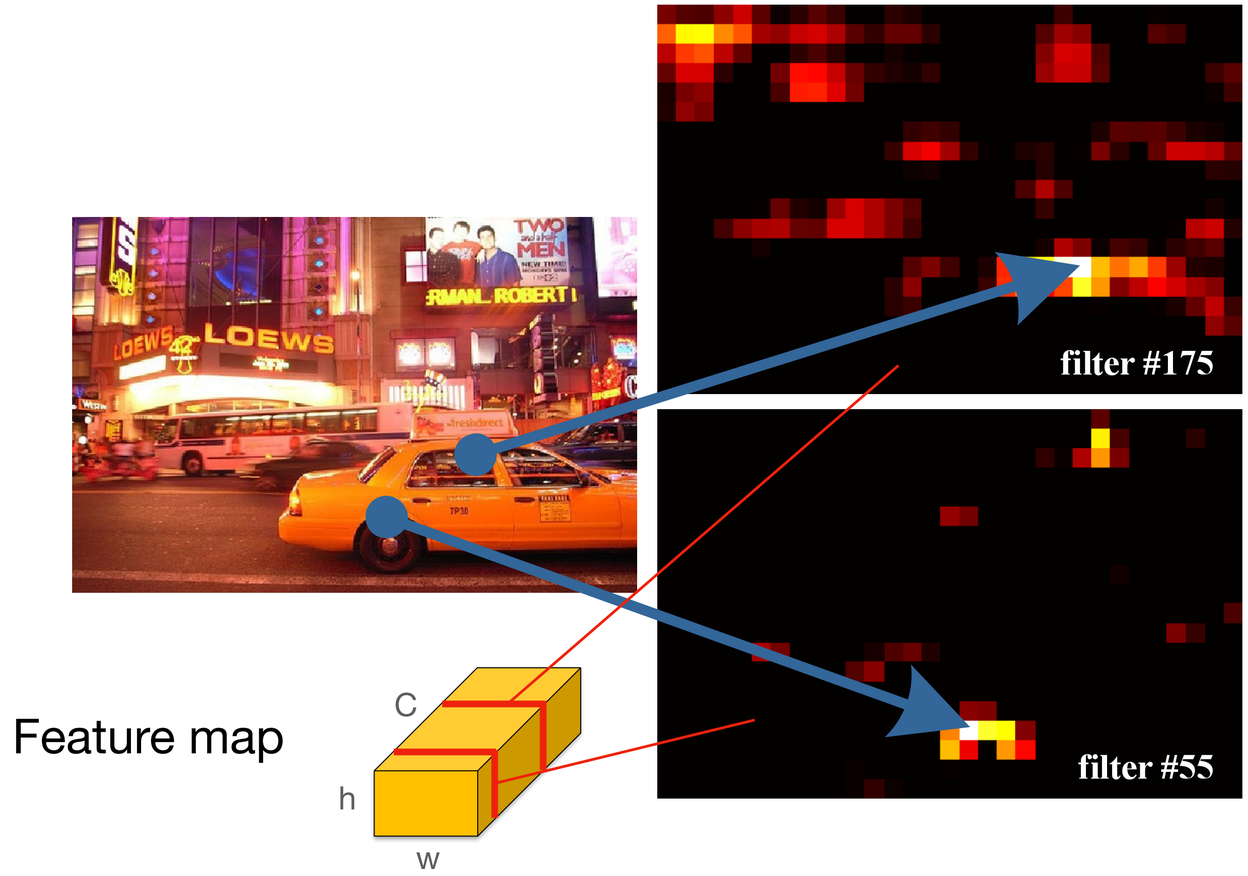
He et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. ECCV 2014.

Dealing with variable input size



What prevents us from dealing with any image size?

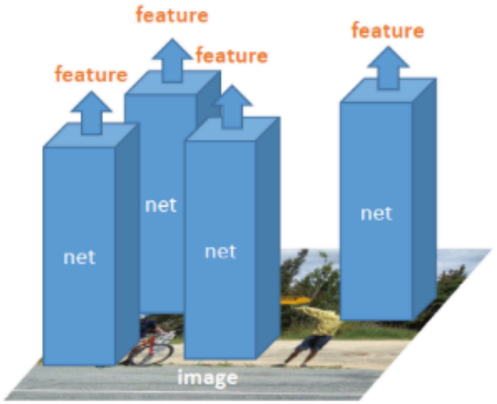
Object detection with deep nets



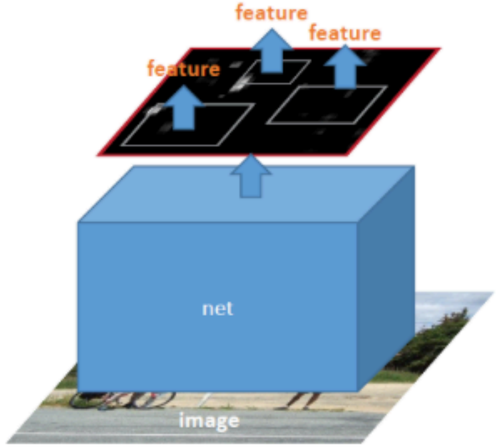
He et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. ECCV 2014.

SPP-Net: Overview

How do we “pool” these features into a common size



R-CNN
2000 nets on image regions



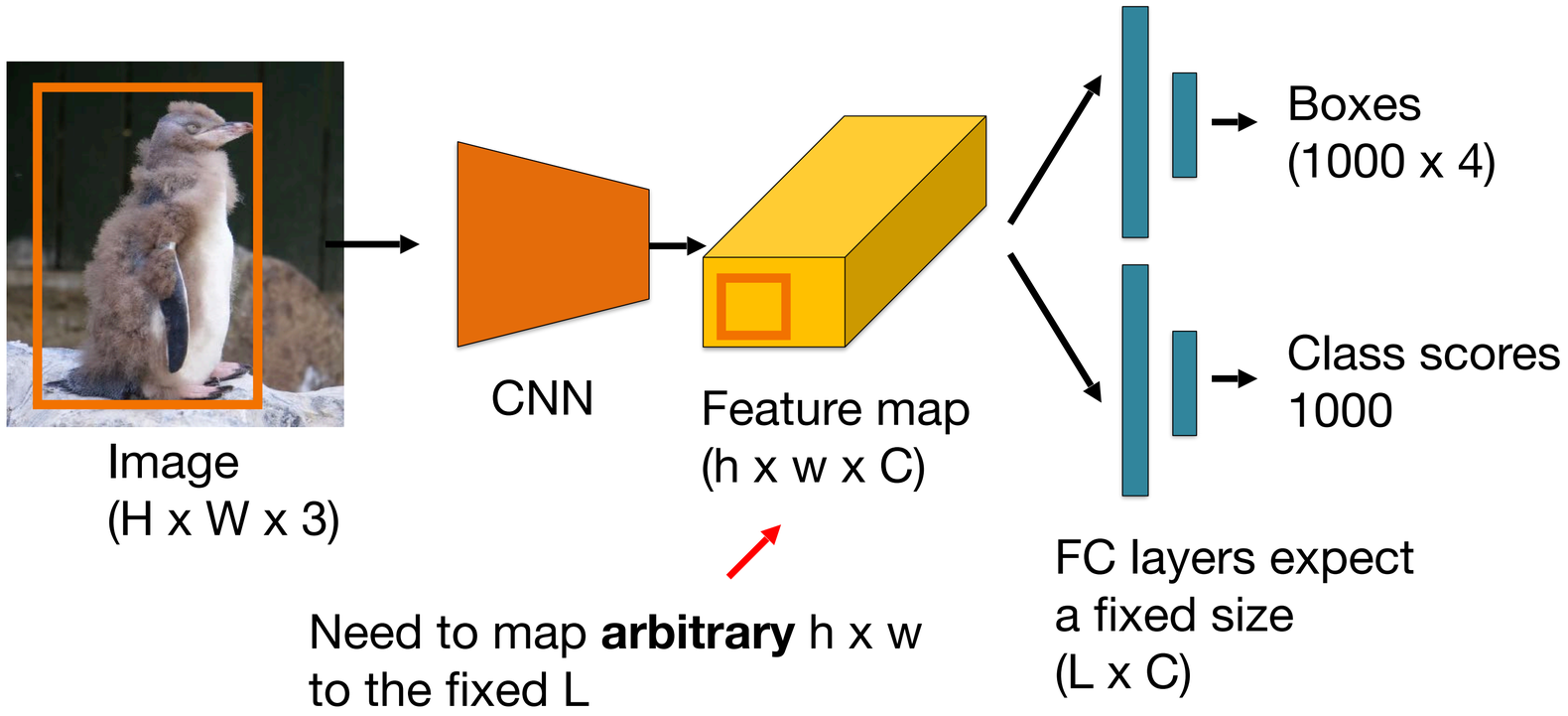
SPP-net
1 net on full image

Frozen

He et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. ECCV 2014.

Fast R-CNN: RoI pooling

- Region-of-Interest pooling

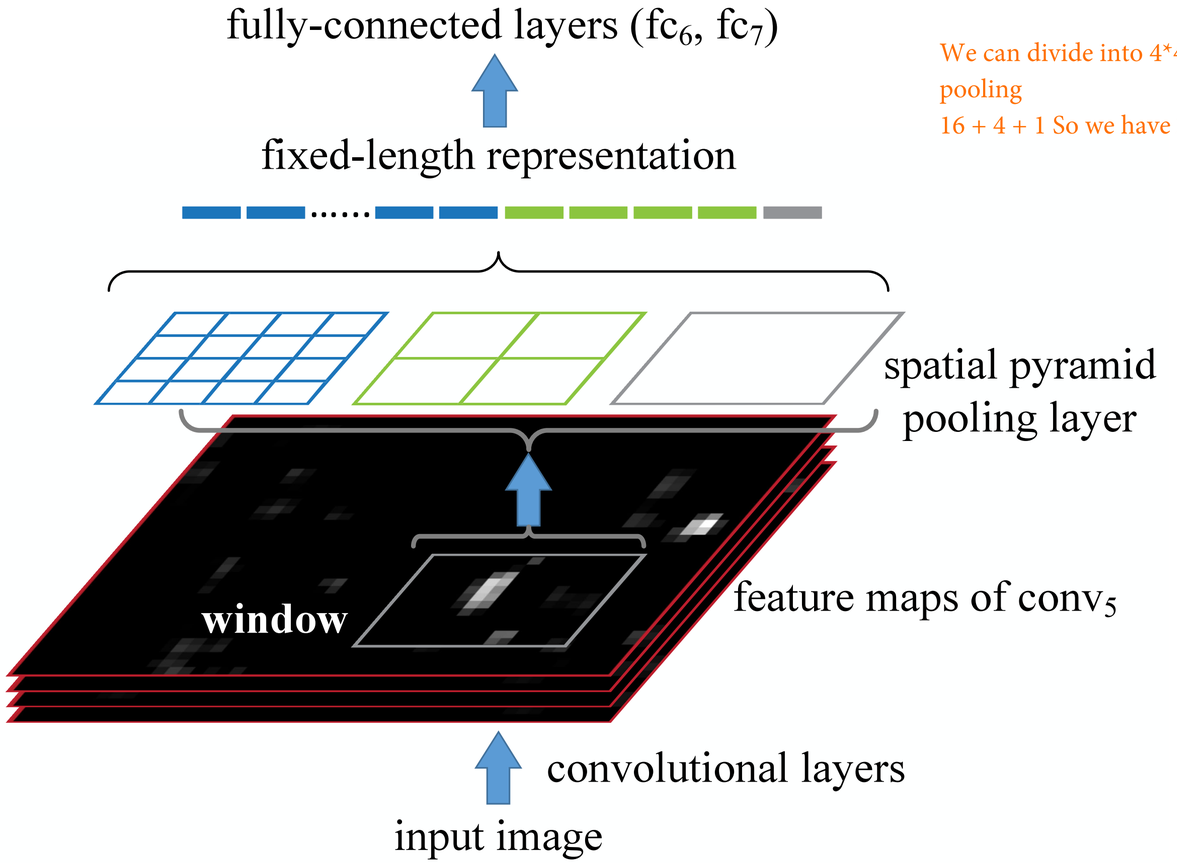


Sermanet et al, “Integrated Recognition, Localization and Detection using Convolutional Networks”, ICLR 2014

SPP-Net: Spatial Pyramid Pooling

suppose we find object proposal, and each proposal can be in different size and aspect raito. So it is a problem because we want to give it to fixed size dense layer.

We can divide into 4*4 windows, 2*2 window and 1*1 windows and take max pooling
16 + 4 + 1 So we have 21 fixed size

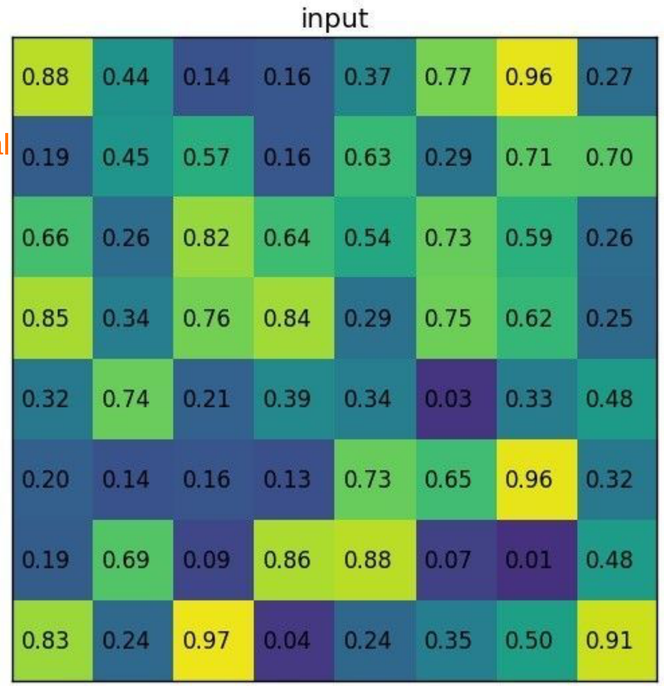


He et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. ECCV 2014.

Spatial Pooling

- Suppose we extract region proposal x=0, y=3, h = 5, w = 7

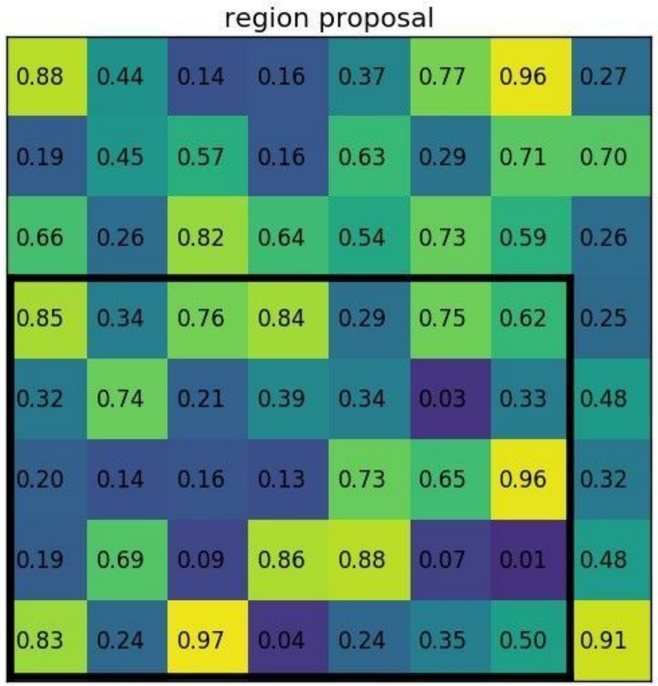
these are slice values of given proposal



Credit: deepsense.ai

Spatial Pooling

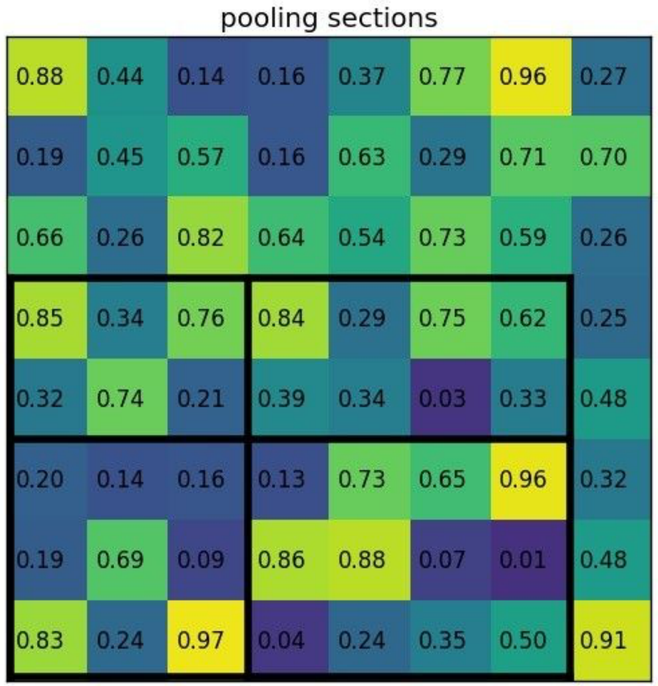
- Suppose we extract region proposal x=0, y=3, h = 5, w = 7



Credit: deepsense.ai

Spatial Pooling

- Suppose we extract region proposal x=0, y=3, h = 5, w = 7

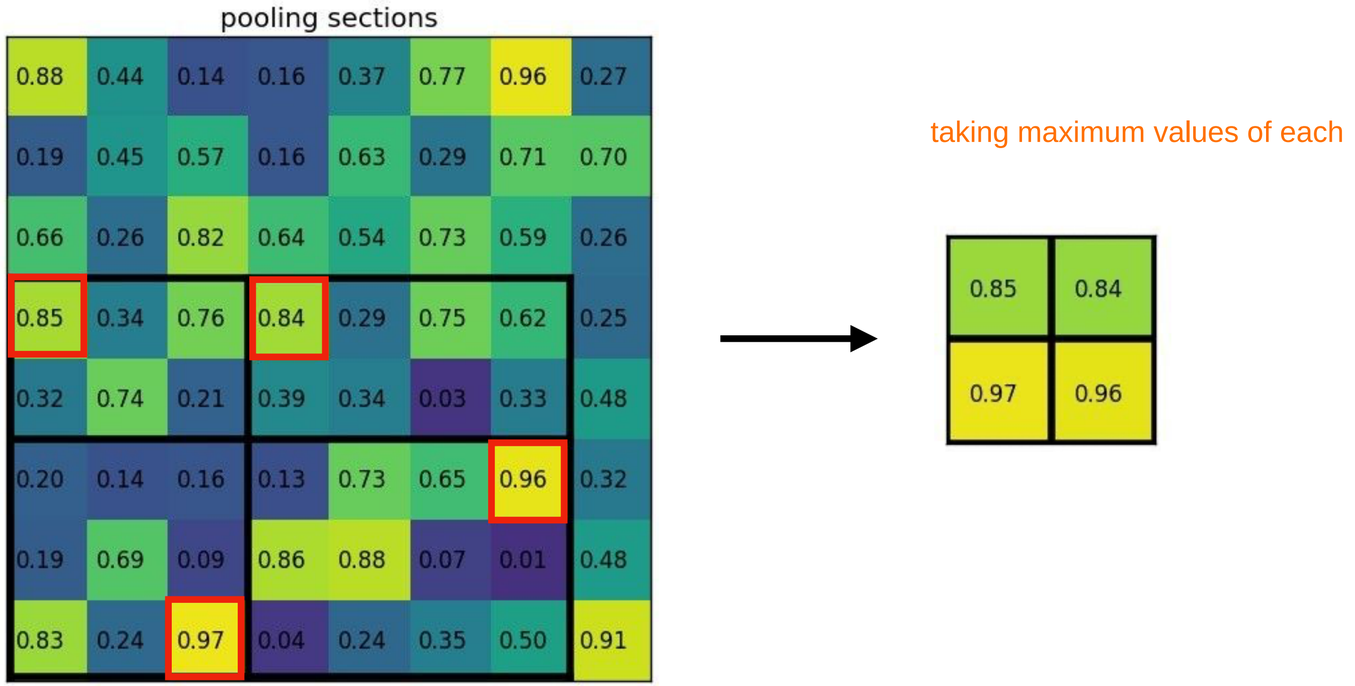


Discretise into 2x2 sub-windows

Credit: deepsense.ai

Spatial Pooling

- Suppose we extract region proposal x=0, y=3, h = 5, w = 7

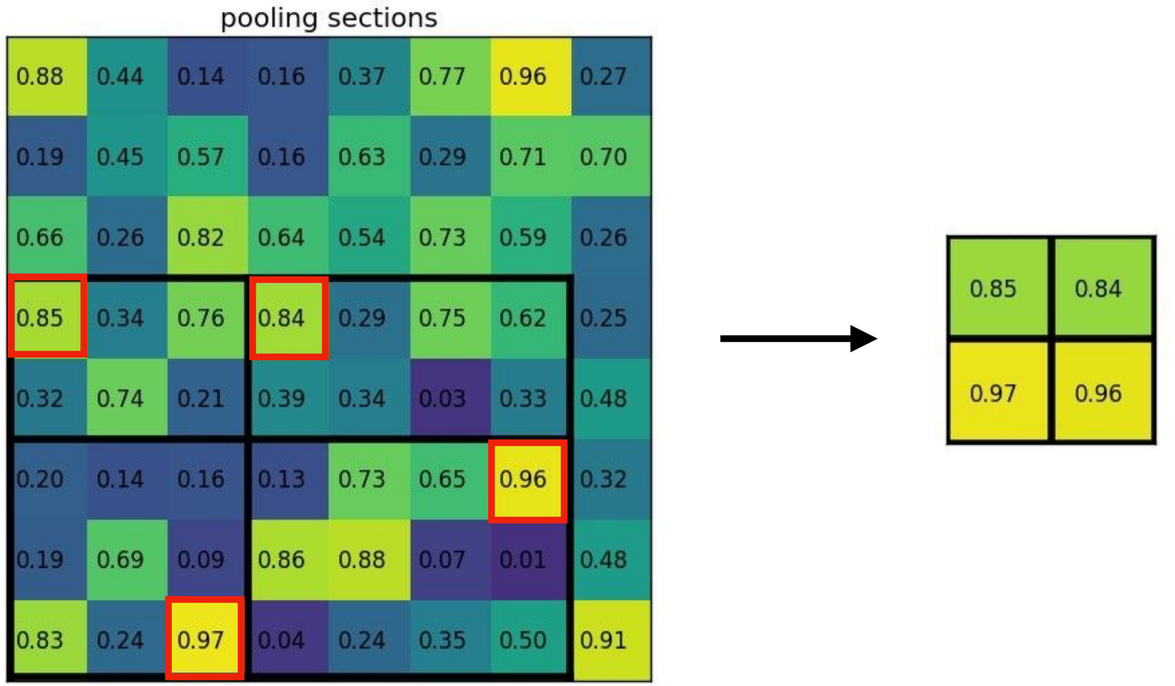


Max-pooling within each window

Credit: deepsense.ai

Spatial Pooling

- Suppose we extract region proposal x=0, y=3, h = 5, w = 7



Max-pooling within each window

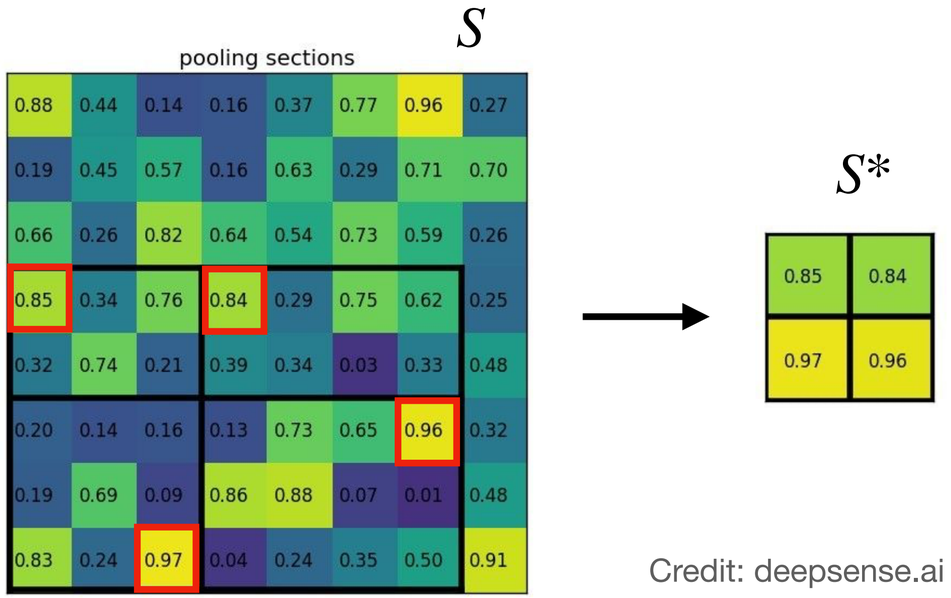
Is it differentiable?

Credit: deepsense.ai

Spatial Pooling

For the max value for each part is differentabile and other are not. Because we dont know previous value in S*

- Is it differentiable?
- $S*$ is differentiable w.r.t. S ?
  - yes and no – depends on pooling
- $S*$ is differentiable w.r.t. (x, y, h, w) ?
  - no



Max-pooling within each window

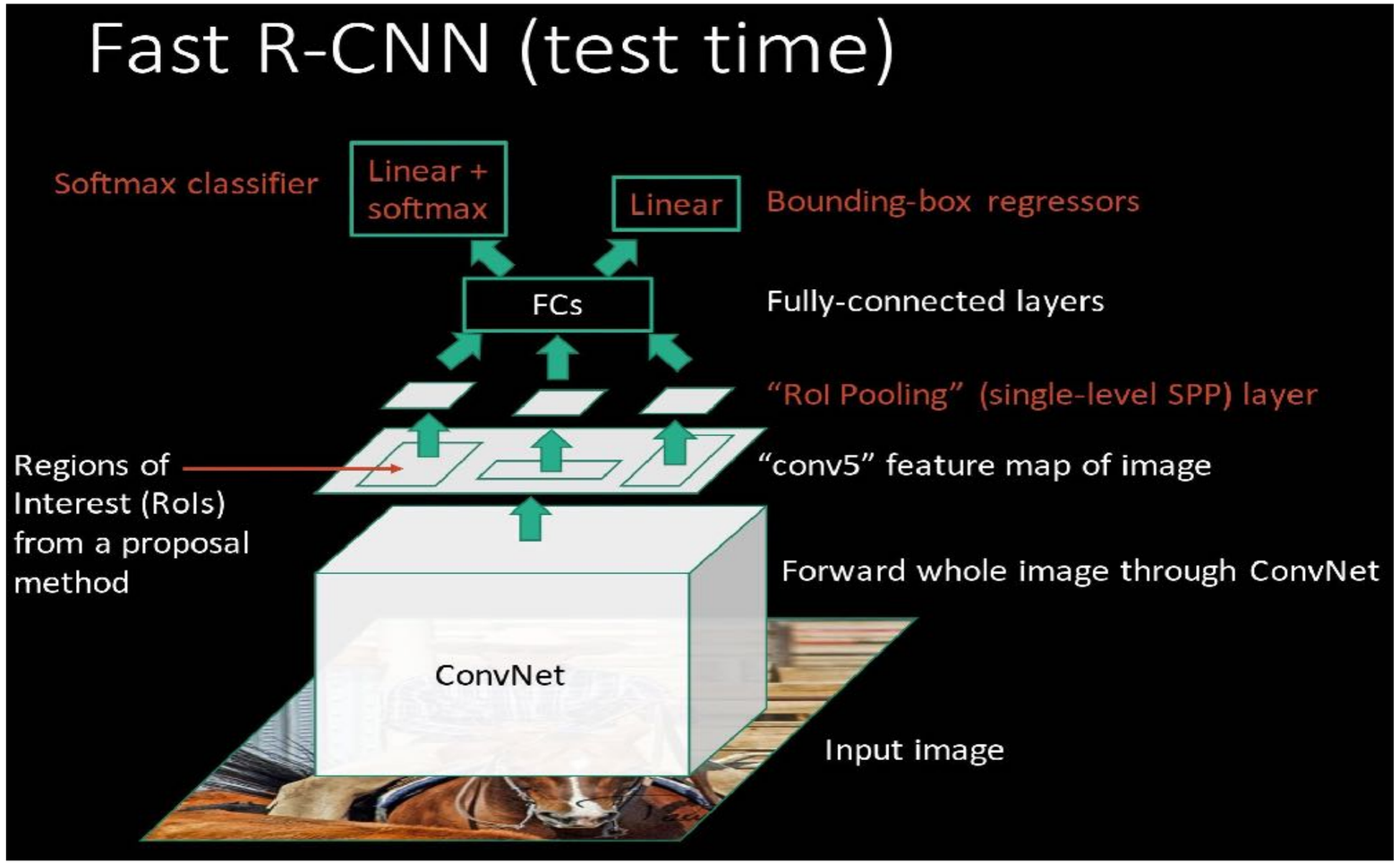
Credit: deepsense.ai

SPP-Net

- Faster training and testing than R-CNN
- Training scheme is still complex
- Still no end-to-end training (fixed convolutional layers)

- Integrate Spatial Pooling into R-CNN → Fast R-CNN

Fast R-CNN

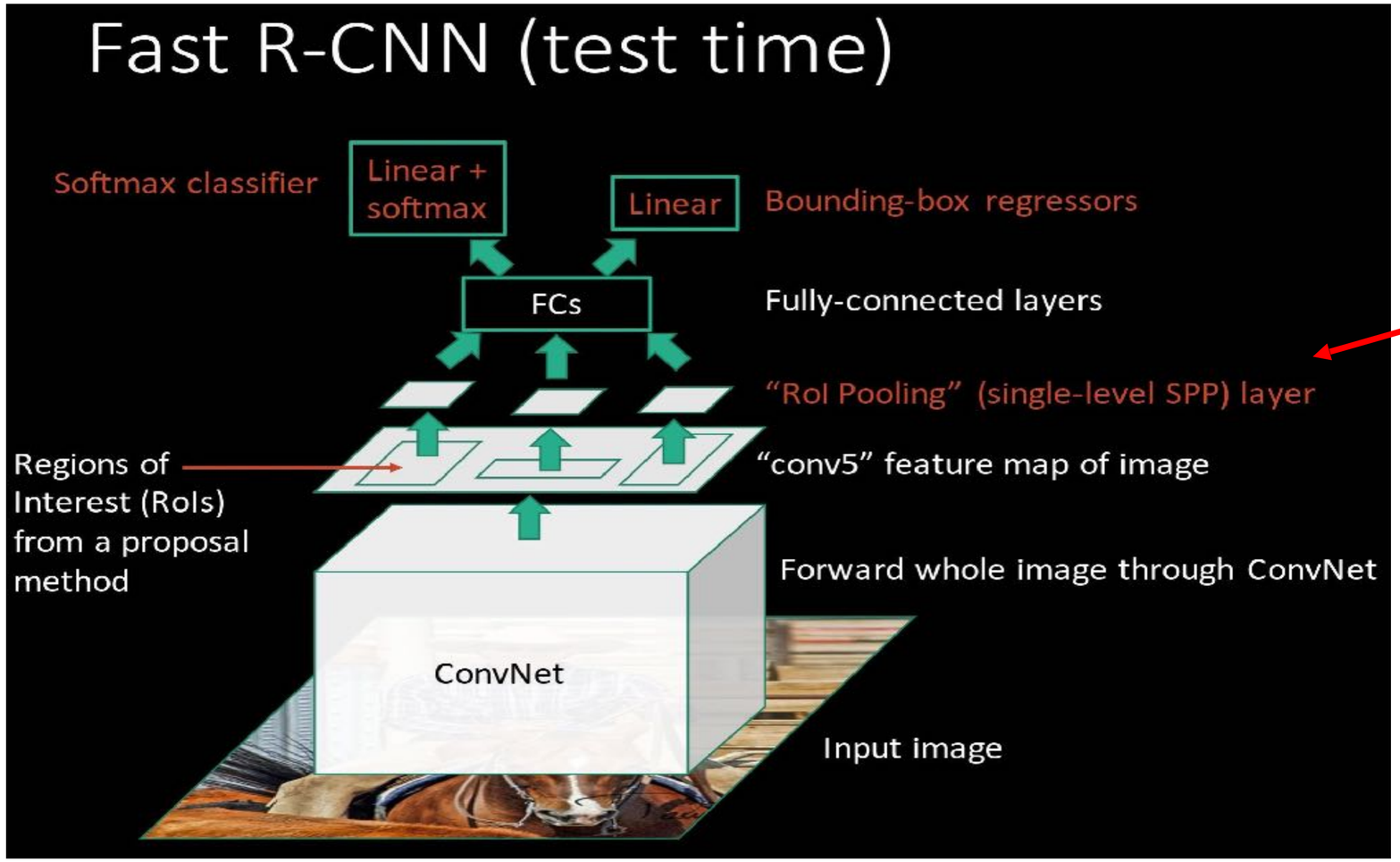


Shared computation at test time (like SPP)

Girschick, "Fast R-CNN", ICCV 2015

Slide credit: Ross Girschick

Fast R-CNN



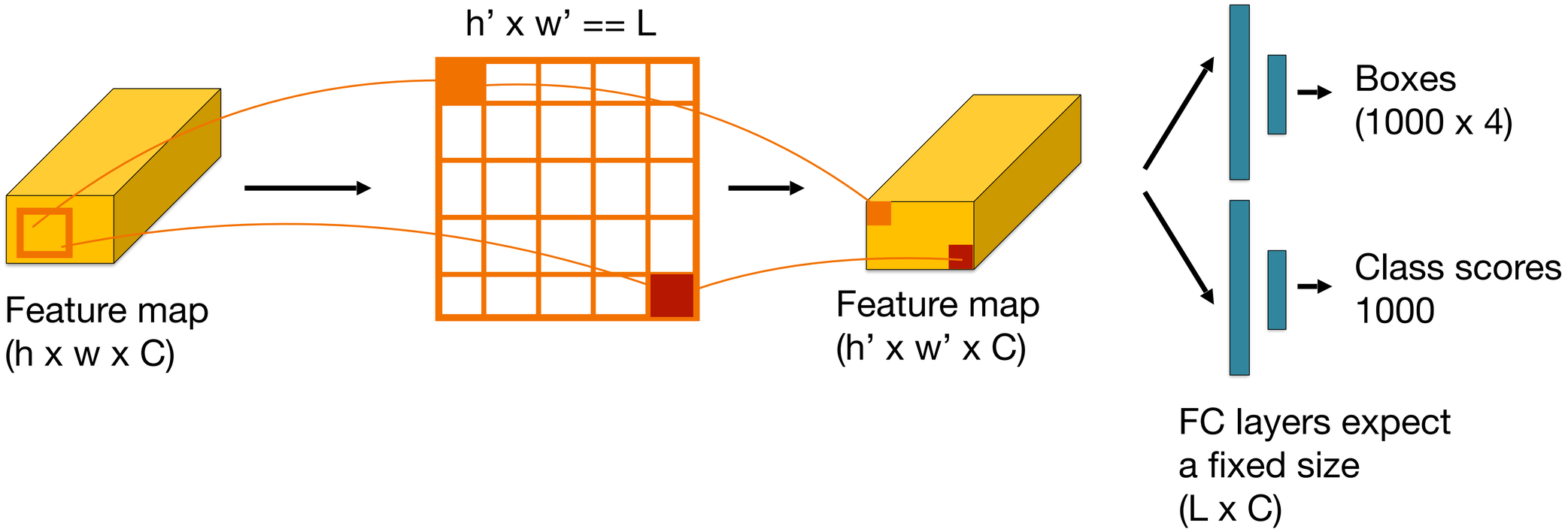
Region-of-Interest pooling

Girschick, "Fast R-CNN", ICCV 2015

Slide credit: Ross Girschick

Fast R-CNN: RoI pooling

- Region-of-Interest pooling



Sermanet et al, “Integrated Recognition, Localization and Detection using Convolutional Networks”, ICLR 2014

Fast R-CNN results

VGG-16 CNN on Pascal VOC 2007 dataset

	R-CNN	Fast R-CNN
Training Time:	84 hours	9.5 hours
Speed-up	1x	8.8x

Fast R-CNN results

VGG-16 CNN on Pascal VOC 2007 dataset

	R-CNN	Fast R-CNN
Training Time:	84 hours	9.5 hours
Speed-up	1x	8.8x
Test time per image	47 seconds	0.32 seconds
Speed-up	1x	146x

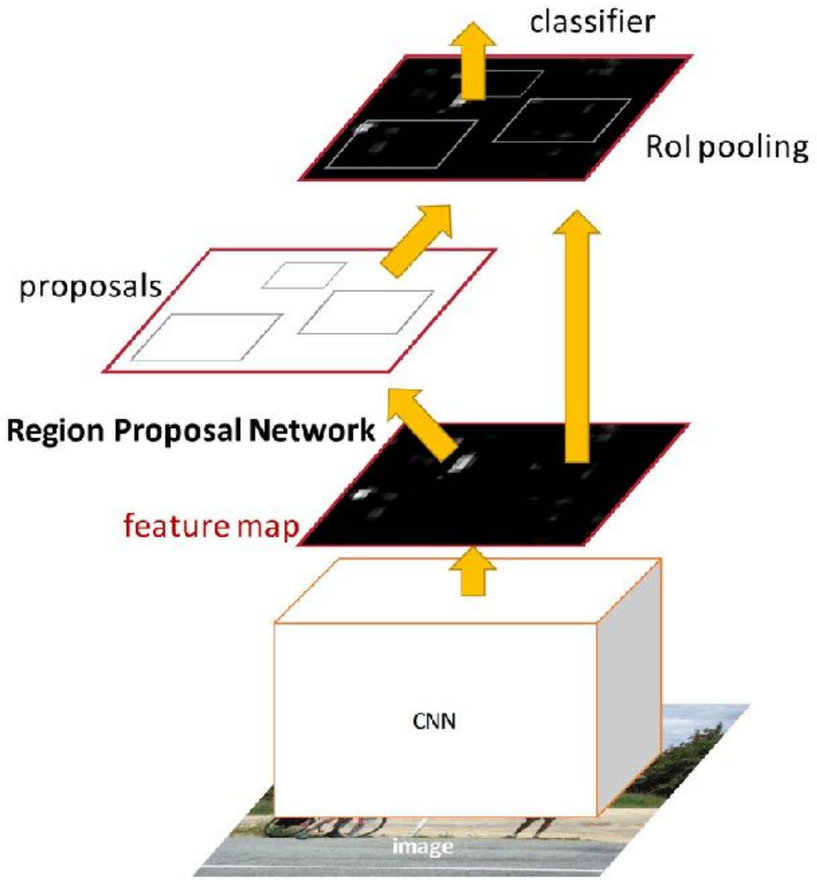
Making Fast R-CNN faster

- We still rely on a standalone object proposals

- Faster R-CNN: Integrated proposal generation with the rest of the pipeline
  - Region Proposal Network (RPN)

- Other than RPN, everything is like Fast R-CNN

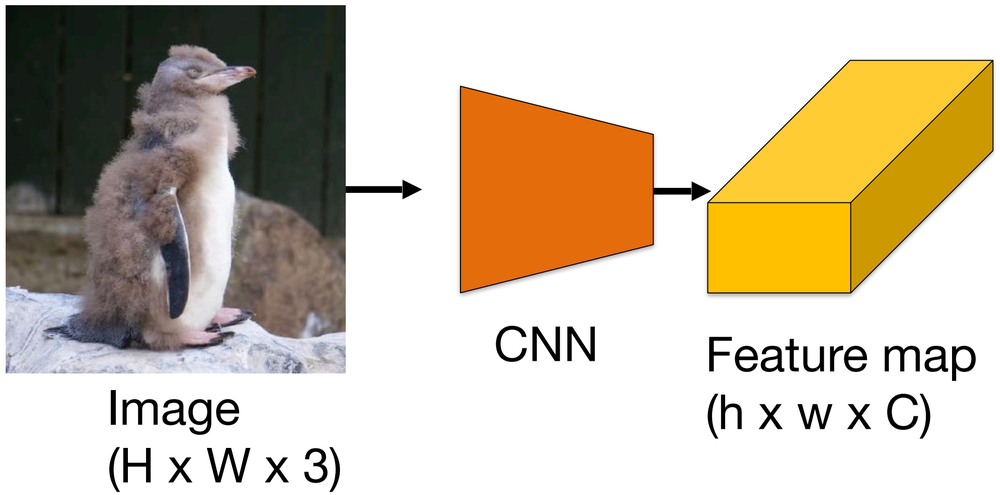
Faster R-CNN



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

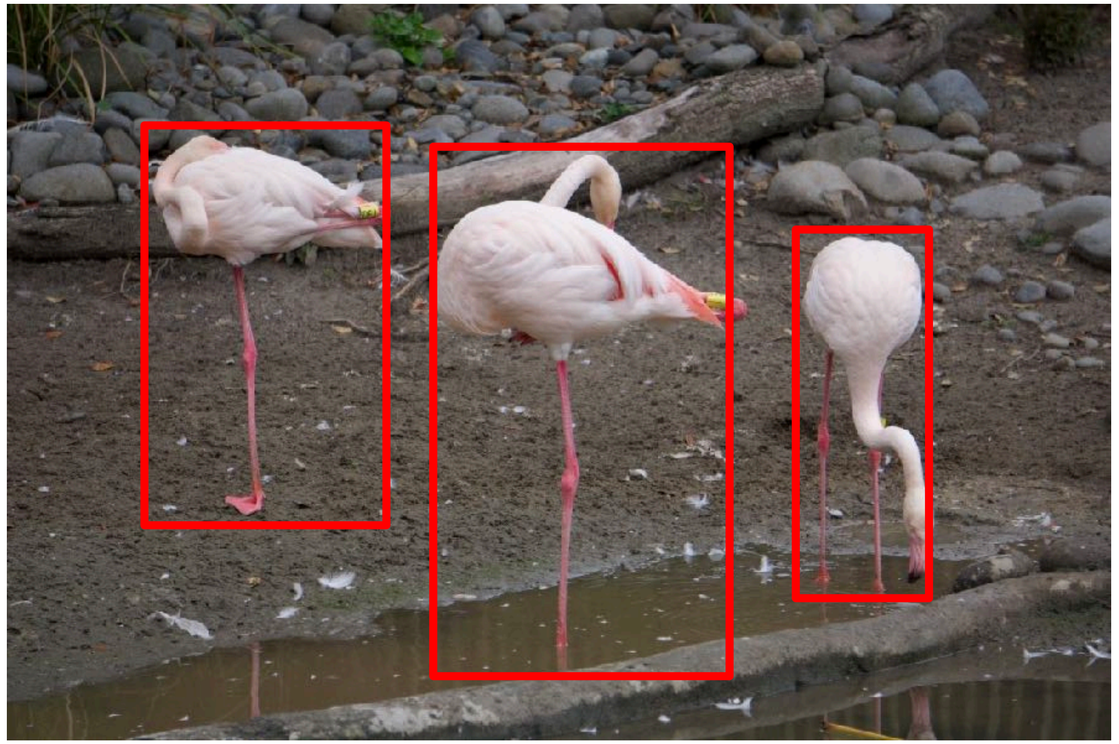
Slide credit: Ross Girschick

Region Proposal Network



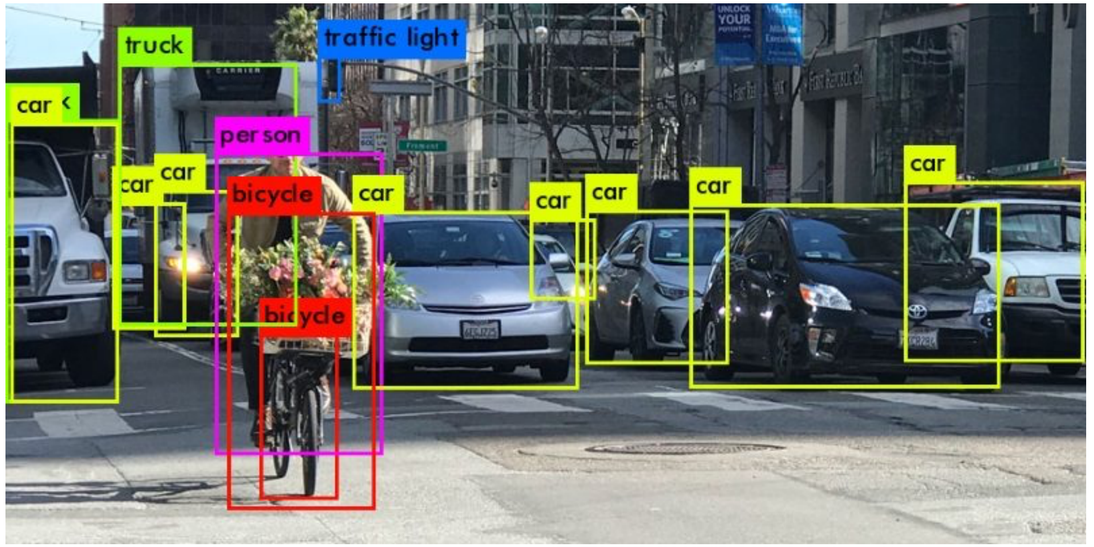
- How many proposals?
- How are they placed?

Multi-object detection



3 objects means having an output of 12 numbers (3 x 4)

Multi-object detection



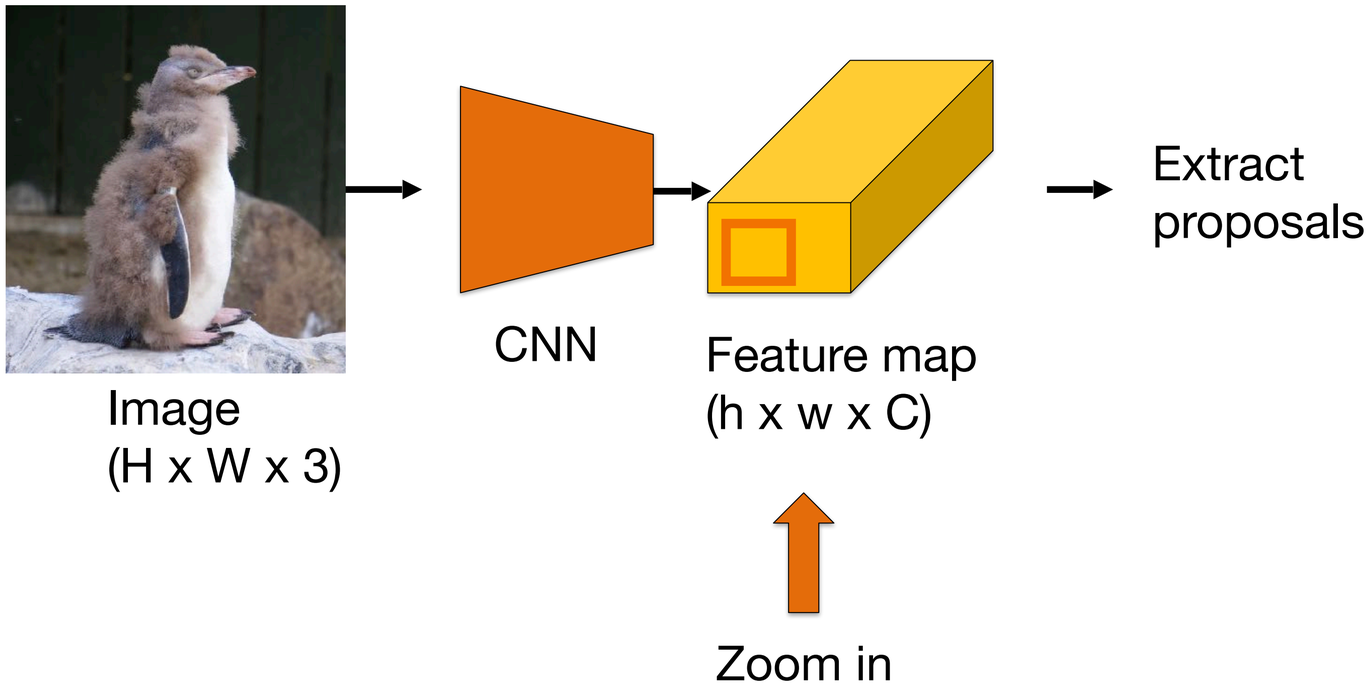
14 objects means having an output of 56 numbers (14 x 4)

Multi-object detection

- Dealing with variable-sized output is challenging
  - There are a couple of workarounds:
    - RNN: (Romera-Paredes and Torr, 2016; Ren et al., 2017; Araslanov et al., 2019).
    - Predict the # of objects: (e.g., Rezatofighi, 2018)

- RPN: Place multiple proposals uniformly densely
  - Learn to predict confidence for each proposal

Region Proposal Network

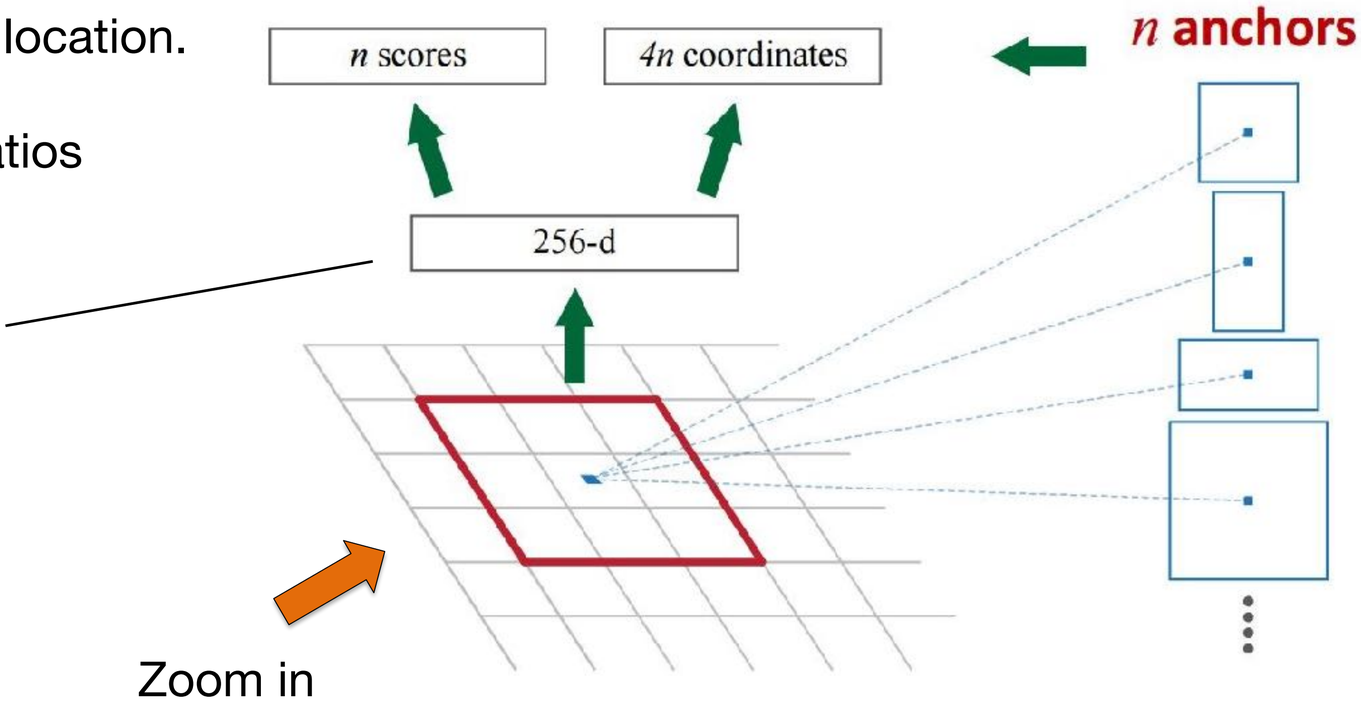


Region Proposal Network

We fix the number of proposals
by using a set of $n = 9$ anchors per location.

9 anchors = 3 scales x 3 aspect ratios

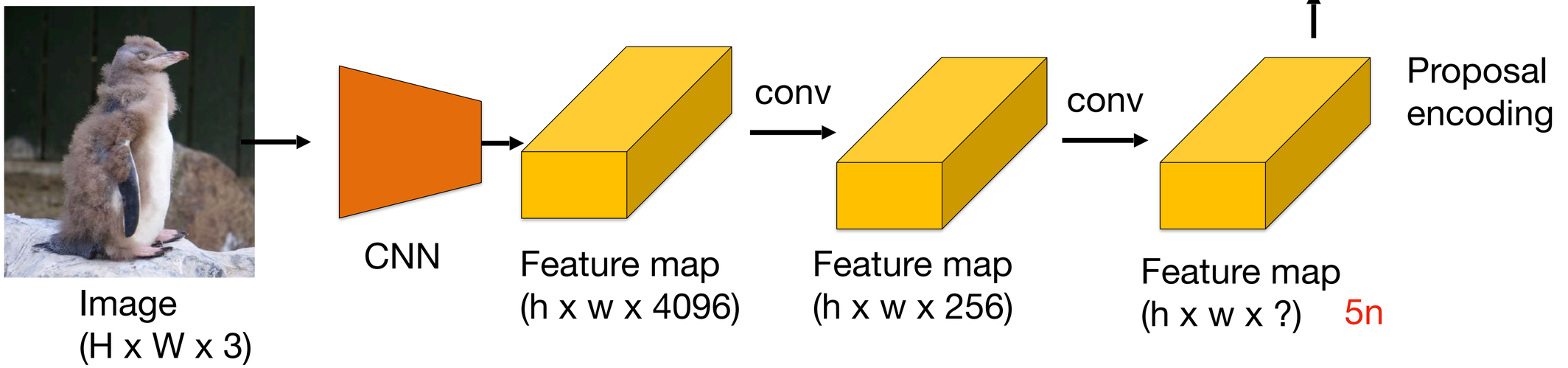
Every location is characterised by
a 256-d descriptor



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

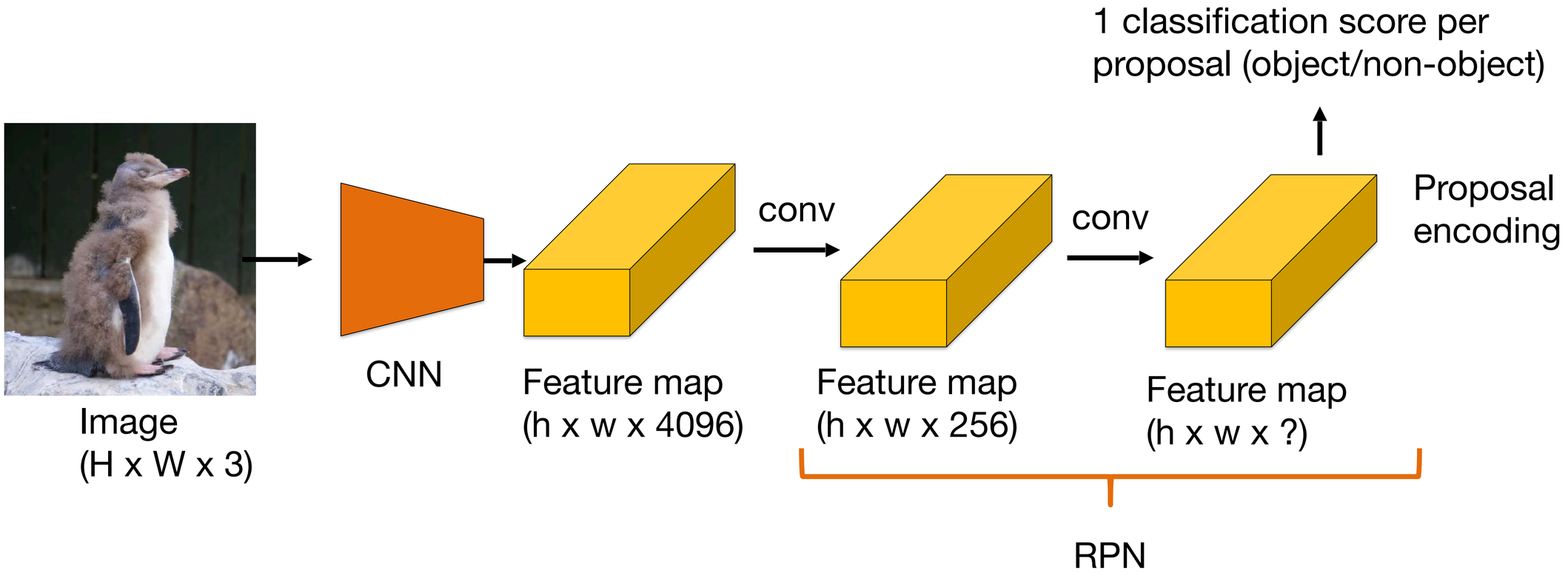
Slide credit: Ross Girschick

Region Proposal Network



QUIZ: Suppose we have n anchors

Region Proposal Network



Per feature map location, we get a set of anchor correction and classification into object/non-object

RPN: Training

- Classification ground truth: We compute $p*$ which indicates how much an anchor overlaps with the ground truth bounding boxes

$$p^* = 1 \quad if \quad \text{IoU} > 0.7$$

$$p^* = 0 \quad if \quad \text{IoU} < 0.3$$

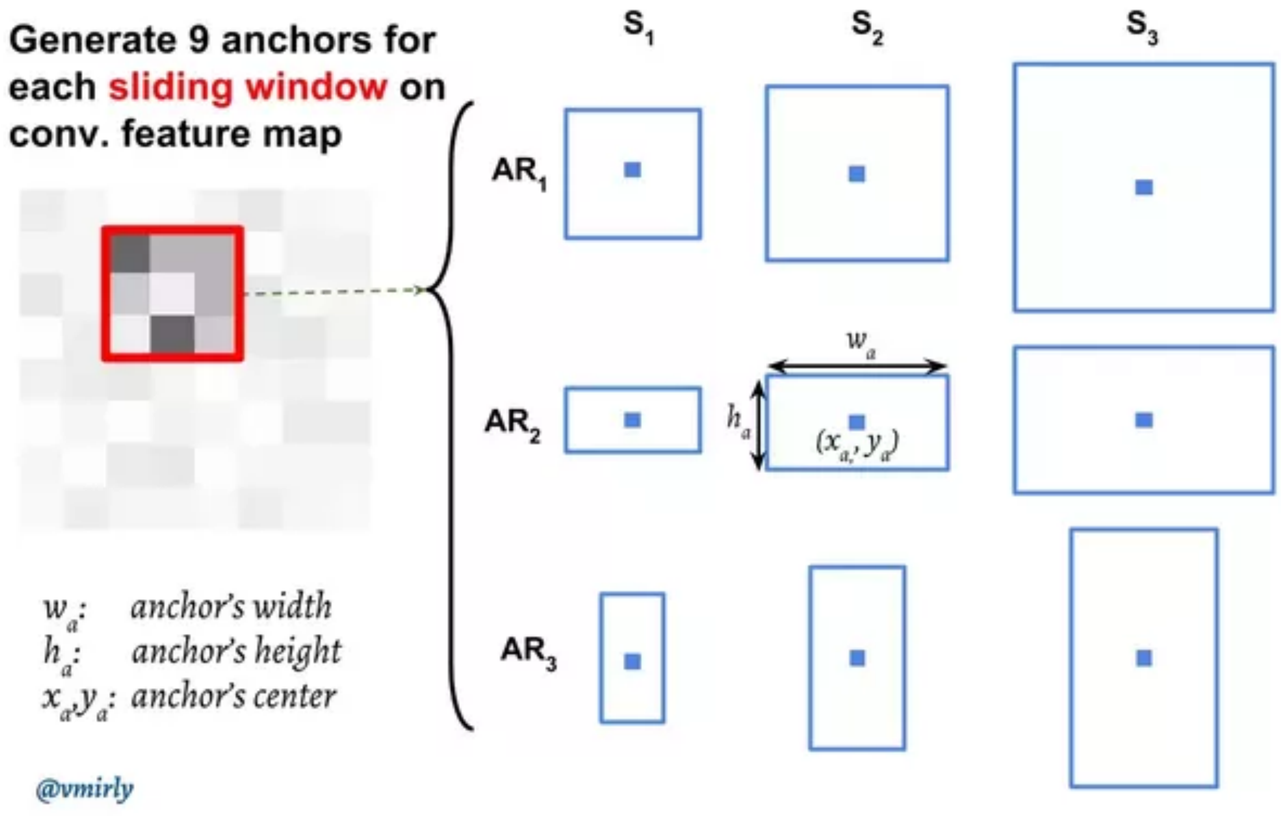
- 1 indicates the anchor represent an object (foreground) and 0 indicates background object. The rest do not contribute to the training.

RPN: Training

- For an image, we randomly sample 256 anchors to form a mini-batch (balanced objects vs. non-objects)
- We learn anchor activate with the binary cross-entropy loss
- Those anchors that contain an object are used to compute the regression loss

RPN: Training

- Each anchor is described by the center position, width and height



RPN: Training

- Each anchor is described by the center position, width and height
- What the network actually predicts are

$$t_x = (x - x_a)/w_a, \qquad t_y = (y - y_a)/h_a,$$

Normalized horizontal shift — Normalized y

$$t_w = log(w/w_a), \qquad t_h = log(h/h_a),$$

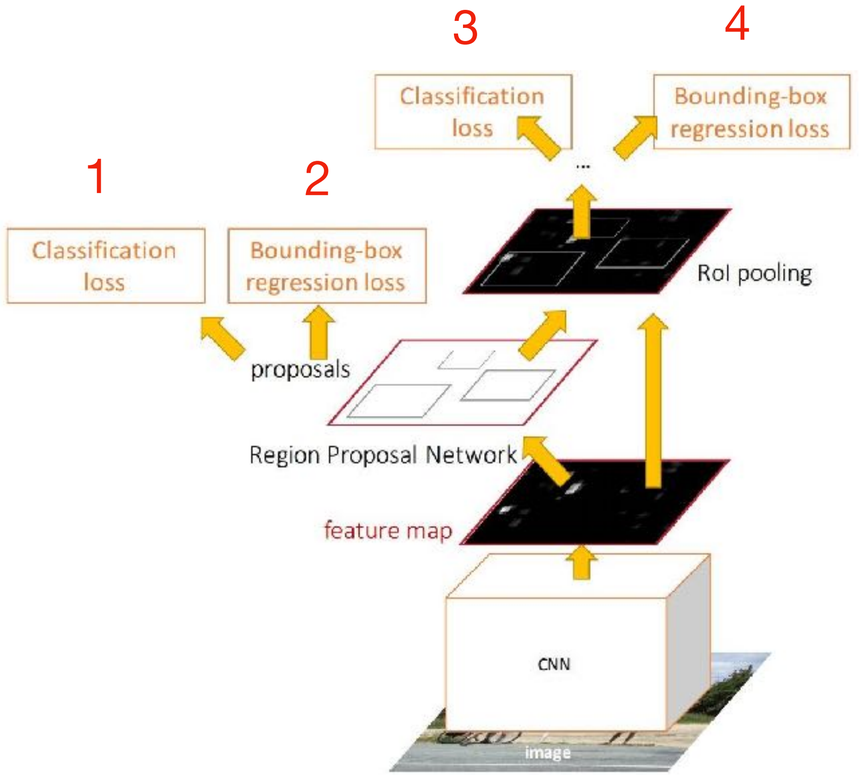
Normalized width — Normalized height

- Smooth L1 loss on regression targets

Faster R-CNN: Training

- First implementation, training of RPN separate from the rest.
- Now we can train jointly!

- Four losses:
  1. RPN classification (object/non-object)
  2. RPN regression (anchor → proposal)
  3. Fast R-CNN classification (type of object)
  4. Fast R-CNN regression (proposal → box)



Faster R-CNN: Training

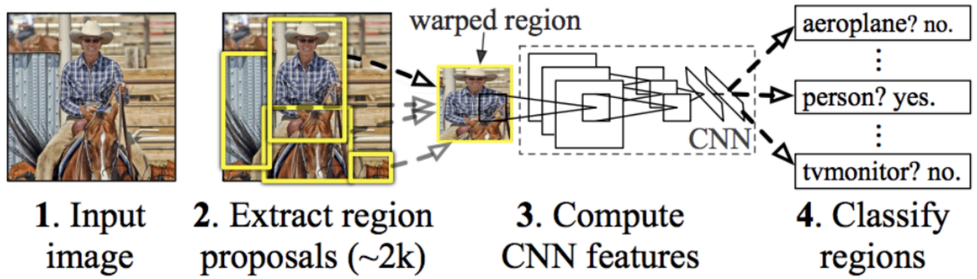
- 10x faster at test time w.r.t. Fast R-CNN
- Trained end-to-end including feature extraction, region proposals, classifier and regressor
- RPN is fully convolutional
- More accurate, since proposals are learned

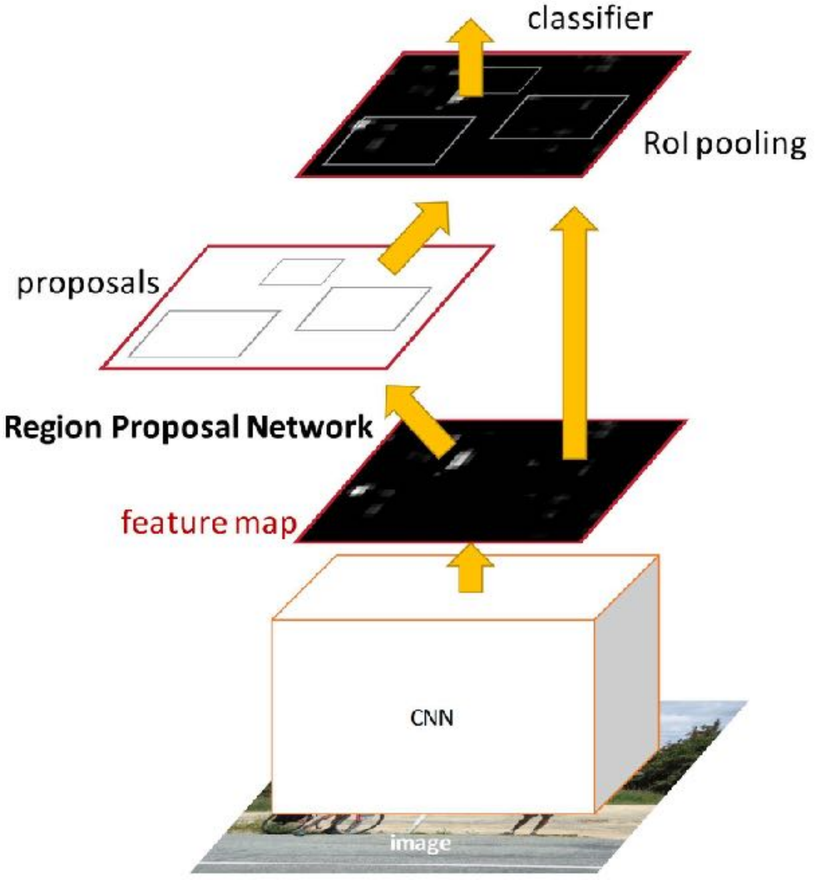
Faster R-CNN: Training

VGG-16 CNN on Pascal VOC 2007 dataset

	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image (w/ proposals, s)	50	2	0.2
Speed-up	1x	8.8x	250x
mAP	66.0	66.9	66.9

From R-CNN to Faster R-CNN





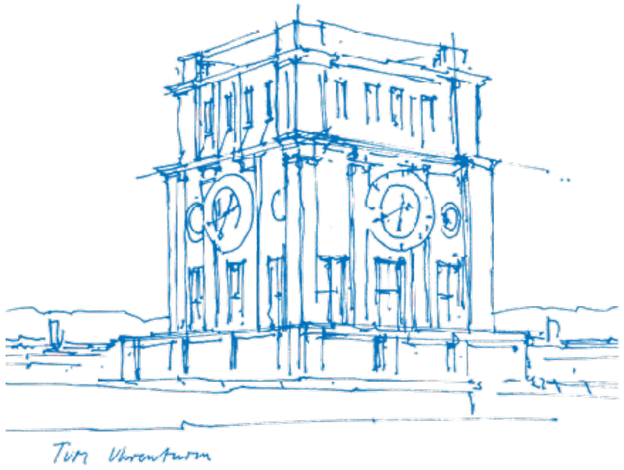
- More efficient and more accurate.
- Incremental but powerful improvements.
- Ent-to-end training: larger benefits from more data

Computer Vision III:

Two-stage object detectors

Nikita Araslanov
08.11.2022

Content credit:
Prof. Laura Leal-Taixé
https://dvl.in.tum.de



Related works

- Shrivastava, Gupta, Girshick. “Training region-based object detectors with online hard example mining”. CVPR 2016.
- Dai, Li, He and Sun. “R-FCN: Object detection via region-based fully convolutional networks”. 2016.
- Dai, Qi, Xiong, Li, Zhang, Hu and Wei. “Deformable convolutional networks”. ICCV 2017.
- Lin, Dollar, Girshick, He, Hariharan and Belongie. “Feature Pyramid Networks for object detection”. CVPR 2017.