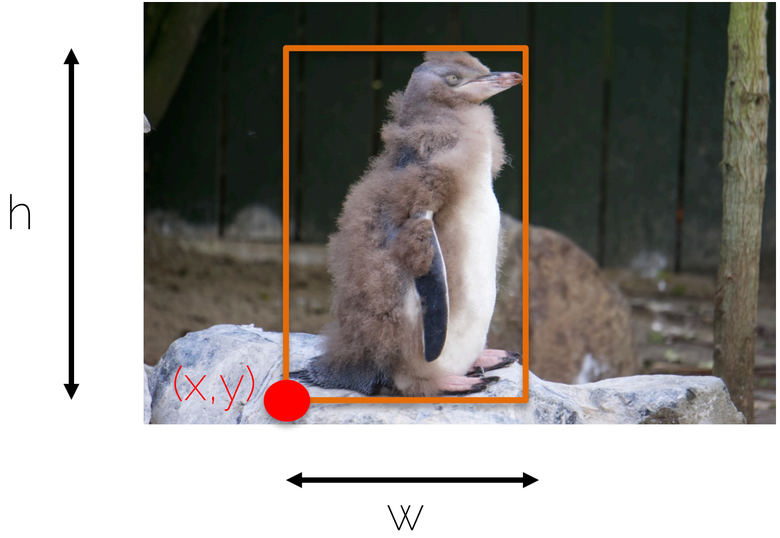
Object detection

Task definition

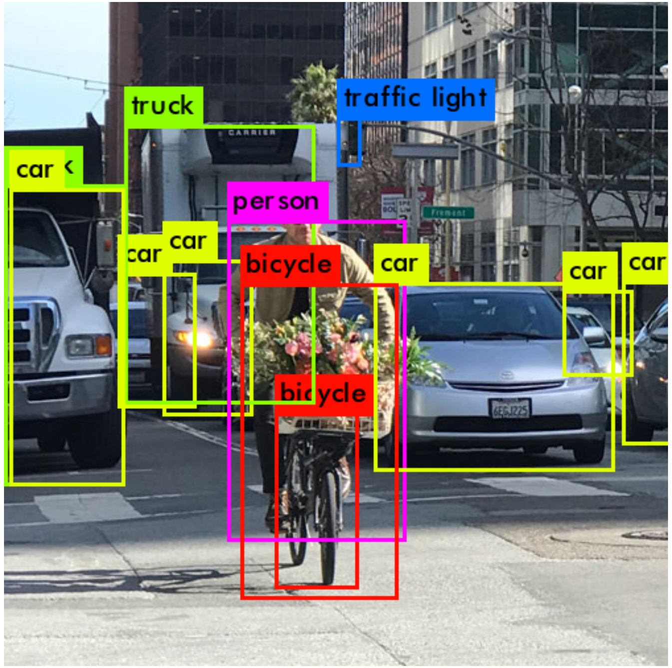
- Object detection problem



Bounding box. (x,y,w,h)

Task definition

- Object detection problem



Bounding box. (x,y,w,h)

+

class

A bit of history

Traditional object detection methods

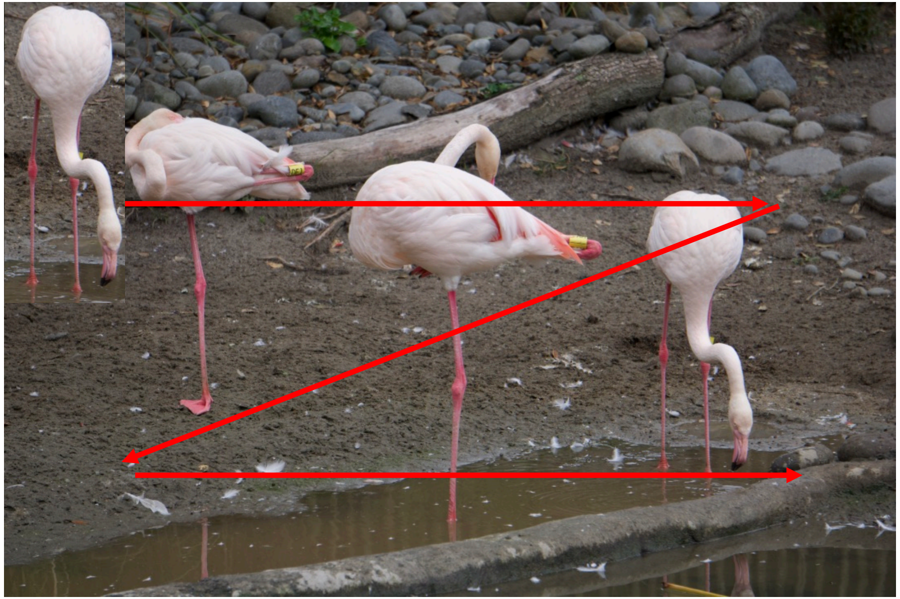
- 1. Template matching + sliding window

Image

Template

Traditional object detection methods

- 1. Template matching + sliding window



Image

Traditional object detection methods

- 1. Template matching + sliding window

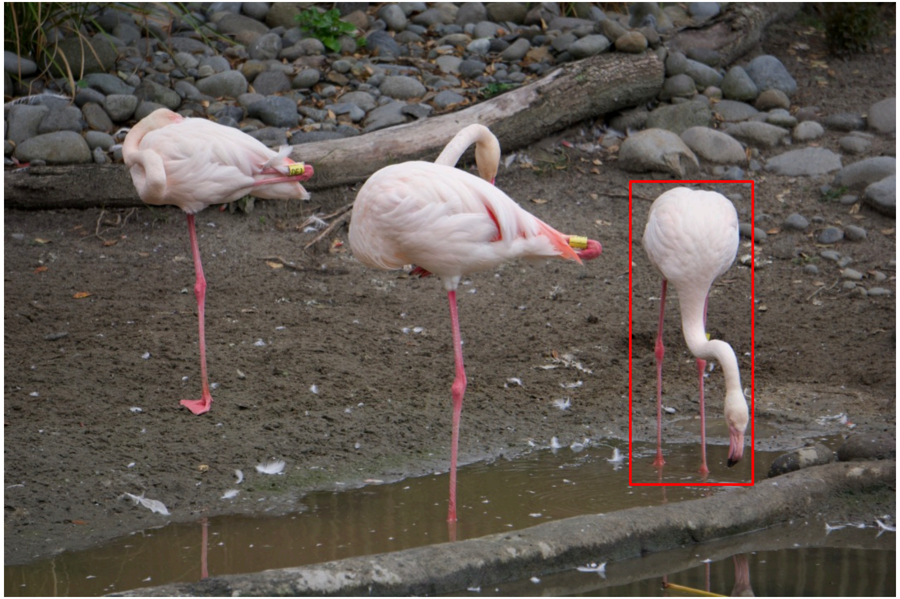
LOW correlation

Image

For every position you evaluate how much do the pixels in the image and template correlate

Traditional object detection methods

- 1. Template matching + sliding window



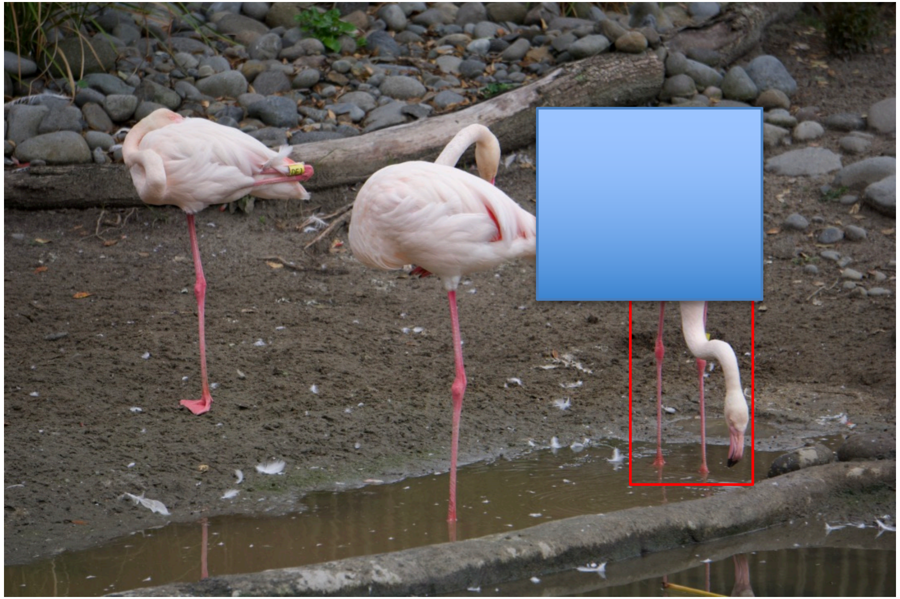
Image

HIGH correlation

For every position you evaluate how much do the pixels in the image and template correlate

Traditional object detection methods

- Problems of 1. Template matching + sliding window



Image

LOW correlation

For every position you evaluate how much do the pixels in the image and template correlate

Traditional object detection methods

- Problems of 1. Template matching + sliding window
  - Occlusions: we need to see the WHOLE object
  - This works to detect a given **instance** of an object but not a **class** of objects

Appearance and shape changes

Pose changes

Traditional object detection methods

- Problems of 1. Template matching + sliding window
  – Occlusions: we need to see the WHOLE object
  – This works to detect a given **instance** of an object but not a **class** of objects
  – Objects have an unknown position, scale and aspect ratio, the search space is searched inefficiently with sliding window

Traditional object detection methods

- 2. Feature extraction + classification

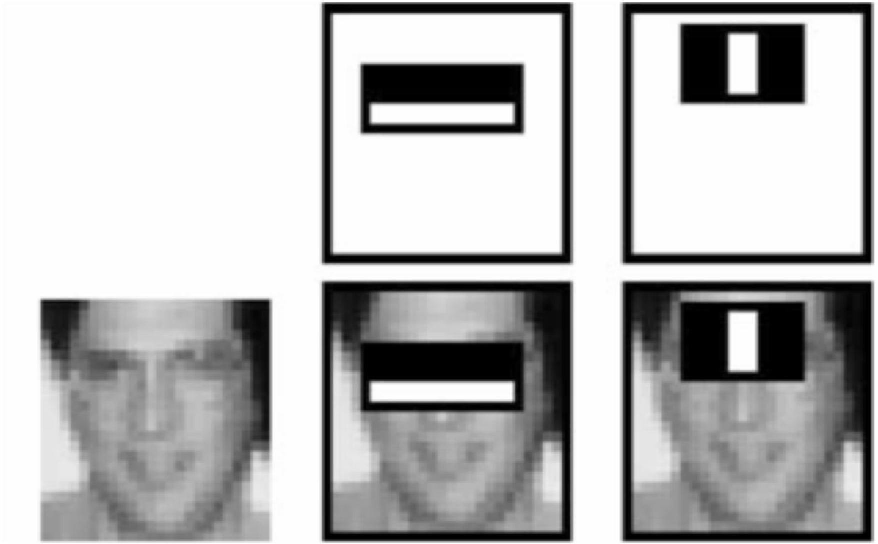
Viola-Jones detector

- 2. Feature extraction + classification
  - Learning multiple weak learners to build a strong classifier
  - That is, make many small decisions and combine them for a stronger final decision

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

Viola-Jones detector

- 2. Feature extraction + classification



Haar features

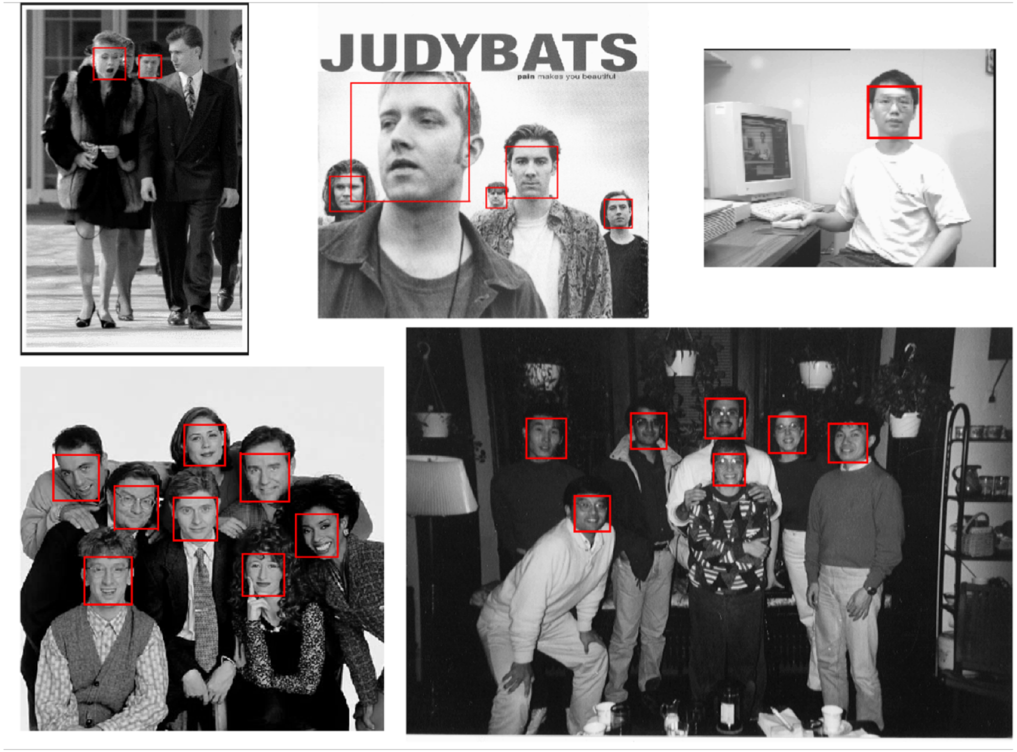
Viola and Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

Viola-Jones detector

- 2. Feature extraction + classification
  - Step 1: Select your Haar-like features
  - Step 2: Integral image for fast feature evaluation
    - I can evaluate which parts of the image have highest cross-correlation with my feature (template)
  - Step 3: AdaBoost for to find weak learner
    - I cannot possibly evaluate all features at test time for all image locations
    - Learn the best set of weak learners
    - Our final classifier is the linear combination of all weak learners

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

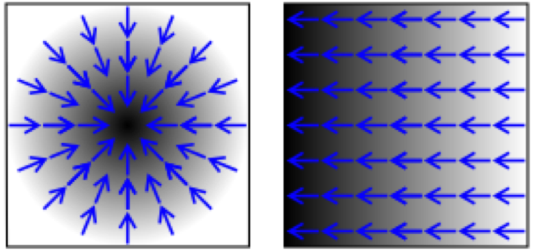
Viola-Jones detector



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

Histogram of Oriented Gradients

- 2. Feature extraction + classification



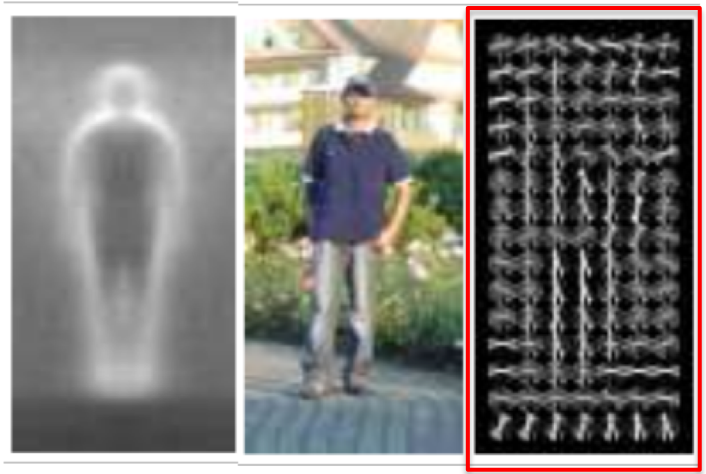
Gradient: blue arrows show the gradient, i.e., the direction of greatest change of the image.

Average gradient image over training samples → gradients provide shape information. Let us create a descriptor that exploits that.

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

Histogram of Oriented Gradients

- 2. Feature extraction + classification



HOG descriptor → Histogram of oriented gradients.
Compute gradients in dense grids, compute gradients and create a histogram based on gradient direction.

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

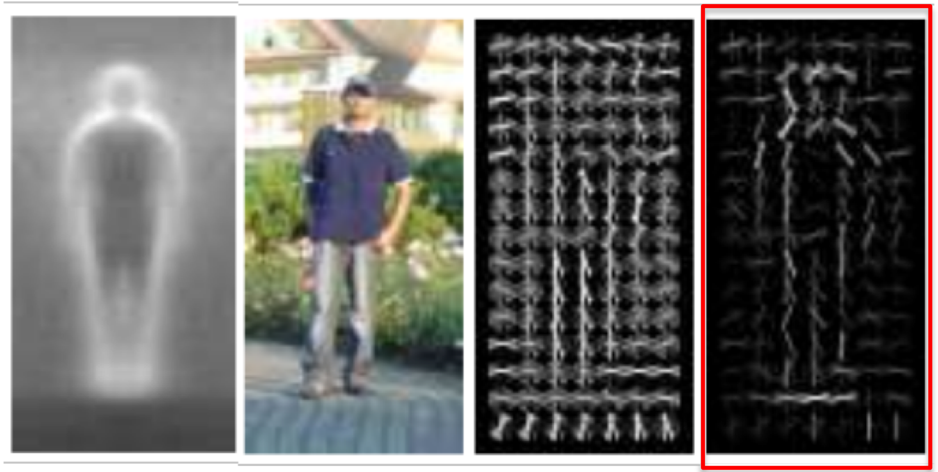
Histogram of Oriented Gradients

- 2. Feature extraction + classification
  - Step 1: Choose your training set of images that contain the object you want to detect.
  - Step 2: Choose a set of images that do NOT contain that object.
  - Step 3: Extract HOG features on both sets.
  - Step 4: Train an SVM classifier on the two sets to detect whether a feature vector represents the object of interest or not (0/1 classification).

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

Histogram of Oriented Gradients

- 2. Feature extraction + classification

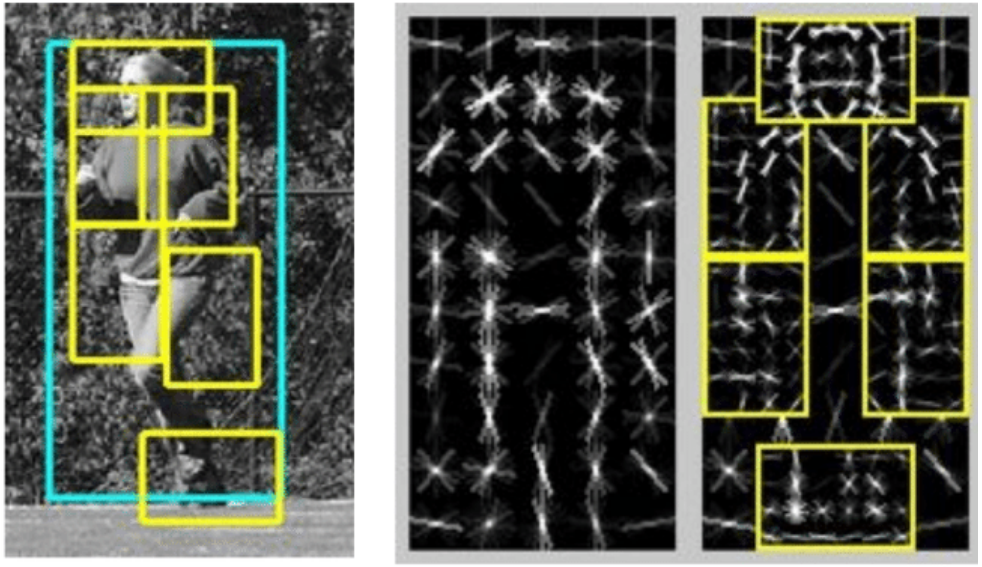


HOG features weighted by the positive SVM weights – the ones used for the pedestrian object classifier.

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

Deformable Part Model

- Also based on HOG features, but based on body part detection → more robust to different body poses

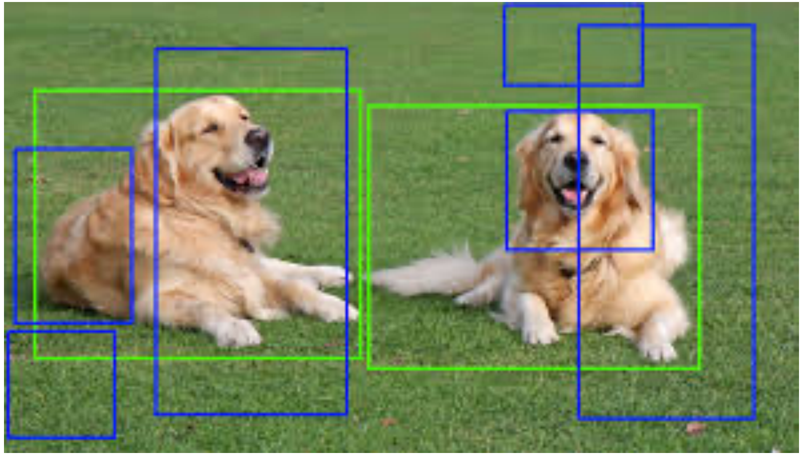


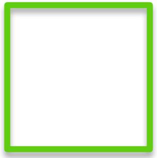
Felzenszwalb et al. A discriminatively trained, multiscale, deformable part model. CVPR 2008.

How to move towards general object detection?

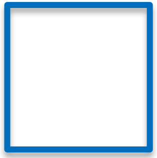
What defines an object?

- We need a generic, **class-agnostic** objectness measure: how likely it is for an image region to contain an object





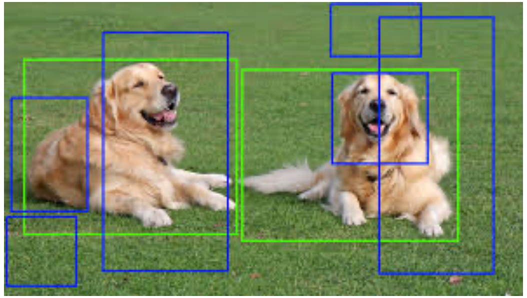
Very likely to be an object



Maybe it is an object

What defines an object?

- We need a generic, **class-agnostic** objectness measure: how likely it is for an image region to contain an object
- Using this measure yields a number of candidate **object proposals** or **regions of interest (RoI)** where to focus.



+ classifier

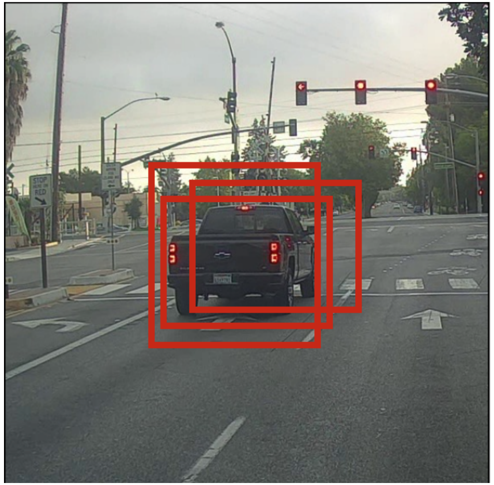
Object proposal methods

- **Selective search**: van de Sande et al. Segmentation as selective search for object recognition. ICCV 2011.

- **Edge boxes**: Zitnick and Dollar. Edge boxes: locating object proposals from edges. ECCV 2014.

Do we want all proposals?

- Many boxes trying to explain one object
- We need a method to keep only the “best” boxes



Non-Maximum Suppression (NMS)

- Many boxes trying to explain one object
- We need a method to keep only the “best” boxes

Non-Maximum Suppression (NMS)

Algorithm 1 Non-Max Suppression

procedure NMS(B,c)
  $B_{nms} \leftarrow \emptyset$
  for $b_i \in B$ **do**
    $discard \leftarrow$ False
    for $b_j \in B$ **do**
      if same$(b_i, b_j) > \boldsymbol{\lambda}_{\mathbf{nms}}$ **then**
        if score$(c, b_j) >$ score(c, b_i) **then**
          $discard \leftarrow$ True
    if not $discard$ **then**
      $B_{nms} \leftarrow B_{nms} \cup b_i$
  return B_{nms}

Start with anchor box i

For another box j

If they overlap

Discard box i if the score is lower than the score of j

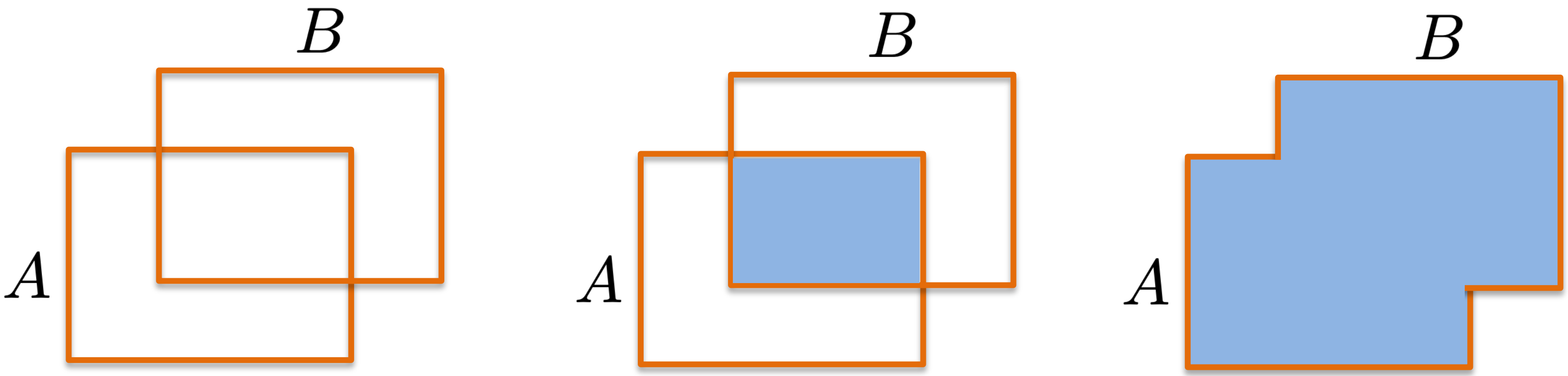
Overlap = to be defined

Score = depends on the task

Region overlap

- We measure region overlap with the **Intersection over Union (IoU)** or **Jaccard Index**:

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



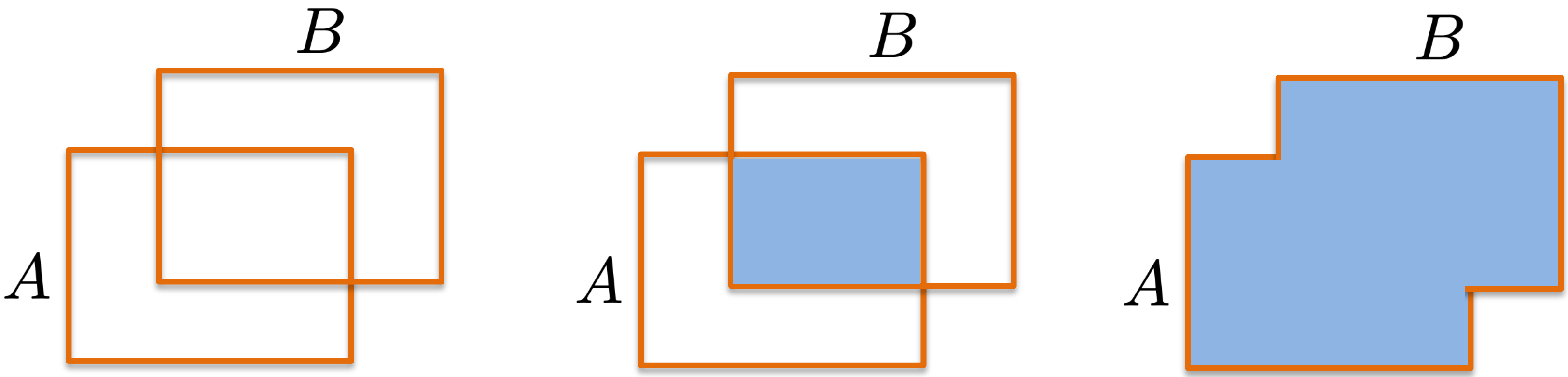
Intersection

Union

Region overlap

- We measure region overlap with the **Intersection over Union (IoU)** or **Jaccard Index**:

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



Intersection Union

Non-Maximum Suppression (NMS)

Algorithm 1 Non-Max Suppression

procedure NMS(B,c)
  $B_{nms} \leftarrow \emptyset$
  for $b_i \in B$ **do**
    $discard \leftarrow$ False
    for $b_j \in B$ **do**
      if same$(b_i, b_j) > \boxed{\lambda_{\mathbf{nms}}}$ **then**
        if score$(c, b_j) >$ score(c, b_i) **then**
          $discard \leftarrow$ True
    if not $discard$ **then**
      $B_{nms} \leftarrow B_{nms} \cup b_i$
  return B_{nms}

Start with anchor box i

For another box j

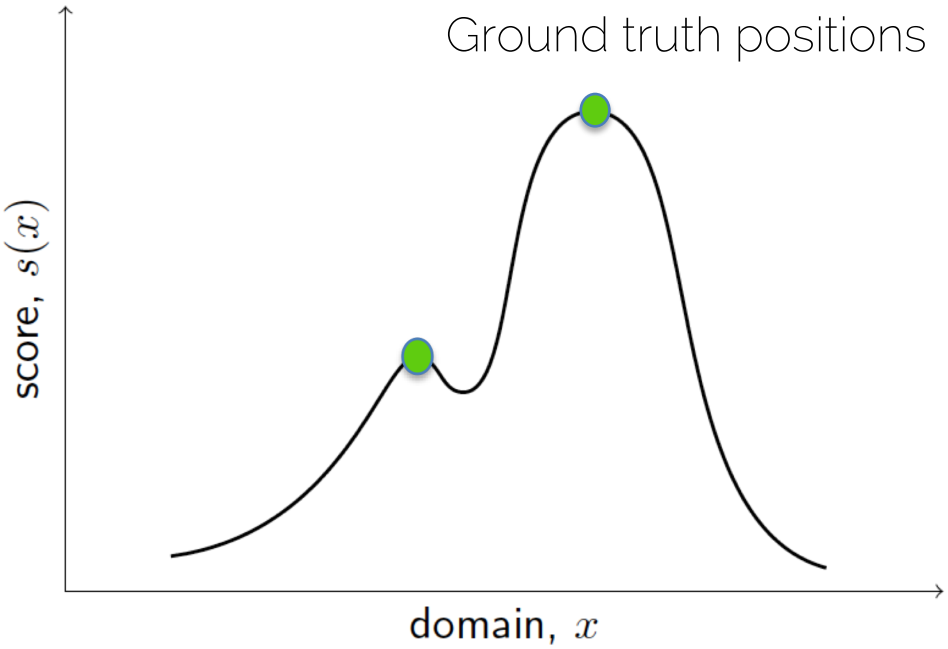
If they overlap

Discard box i if the score is lower than the score of j

Overlap = to be defined

Score = depends on the task

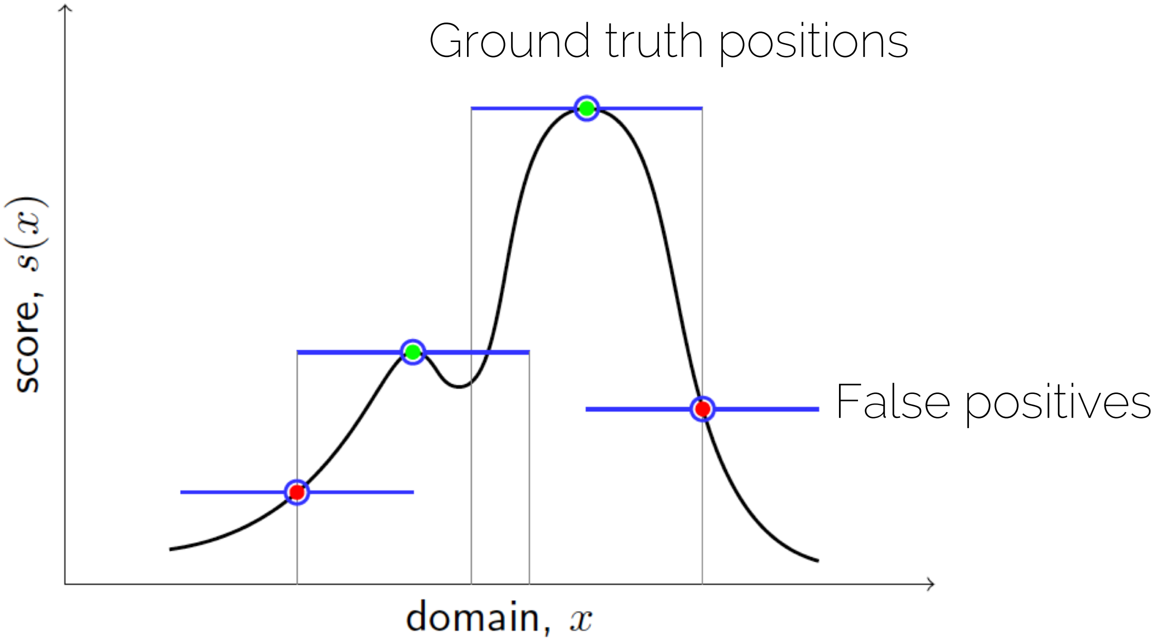
NMS: the problem



Hosang, Benenson and Schiele. A Convnet for Non-Maximum Suppression. 2015

NMS: the problem

- Choosing a narrow threshold

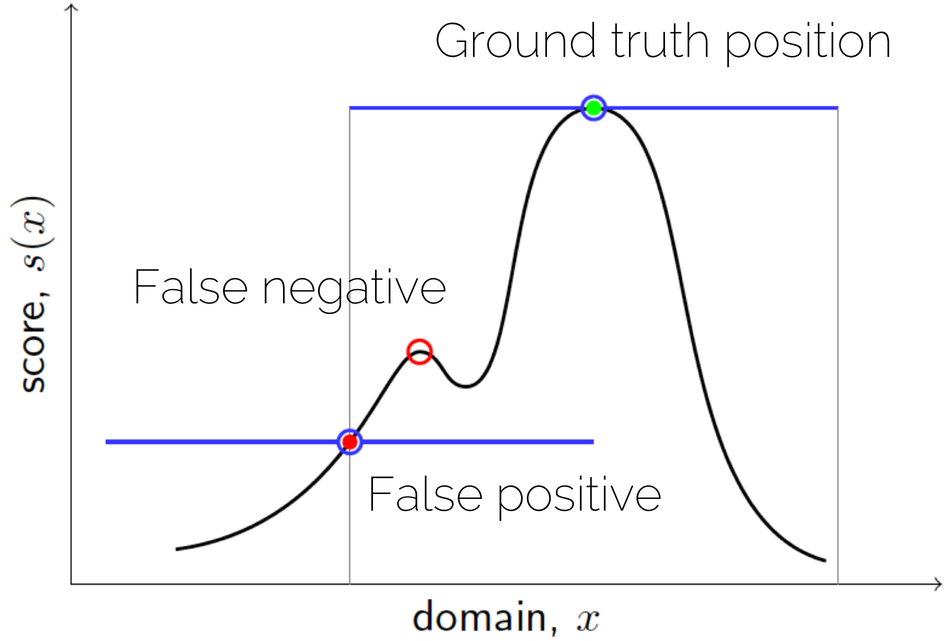


Low Precision

Hosang, Benenson and Schiele. A Convnet for Non-Maximum Suppression. 2015

NMS: the problem

- Choosing a wider threshold



Low Recall

Hosang, Benenson and Schiele. A Convnet for Non-Maximum Suppression. 2015

Non-Maximum Suppression (NMS)

- NMS will be used at test time. Most detection methods (even Deep Learning ones) use NMS!

Detection evaluation

Evaluation measures

- For each image and each class independently, rank the predicted detections by descending order of confidence (score).
- Assign each detection to the ground truth detection of **maximum overlap (IoU)** if the overlap is above a threshold (typically 0.5 or 0.7 IoU).
- Mark that detection as a true positive.
- One ground truth detection can be assigned to one predicted detection only.

Russakovsky et al. Imagenet Large Scale Visual Recognition Challenge. IJCV 2015.

Evaluation measures

- For each image and each class independently, rank the predicted detections by descending order of confidence (score).
- Assign each detection to the ground truth detection of **maximum overlap (IoU)** if the overlap is above a threshold (typically 0.5 or 0.7 IoU).
- Mark that detection as a true positive.
- One ground truth detection can be assigned to one predicted detection only.

Russakovsky et al. Imagenet Large Scale Visual Recognition Challenge. IJCV 2015.

Object detection datasets

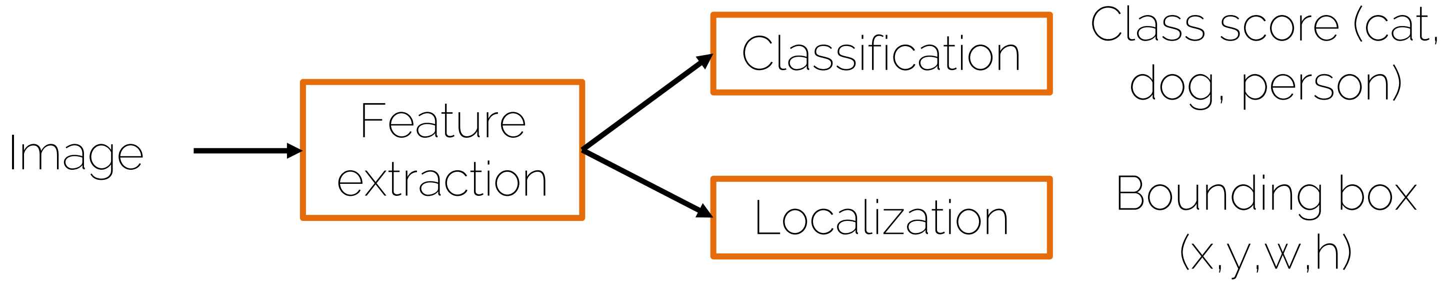
- PASCAL VOC 2007-12: 20 classes; images 5-11k train/val, 5-11k test (public for 2007)
- ImageNet ILSVRC 2010-17: 200 classes (subset or merged from classication task); images 400-450k train (partially annotated), 20k val, 40k test
- COCO 2015-: 80 classes; images 80k train, 40k val (115k/5k in 2017), 40k test, 120k unlabeled; smaller objects
- Open Images 2018-: 600 classes; images 1.74M train, 41k val, 125k test

Everingham et al. IJCV 2015. The PASCAL Visual Object Classes Challenge: a Retrospective.
Russakovsky et al. IJCV 2015. Imagenet Large Scale Visual Recognition Challenge.
Lin et al. ECCV 2014. Microsoft COCO: Common Objects in Context.
Kuznetsova et al. 2018. The Open Images Dataset V4: Unied image classication, object detection, and visual relationship detection at scale.

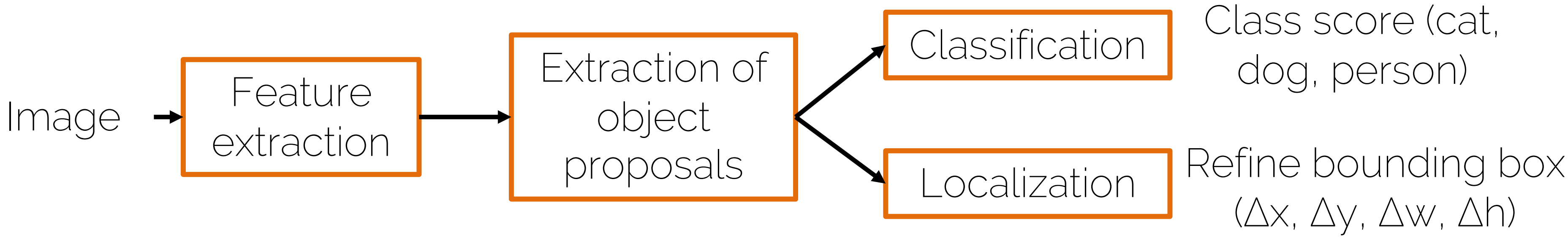
Learning-based detectors

Types of object detectors

- One-stage detectors



- Two-stage detectors



Types of object detectors

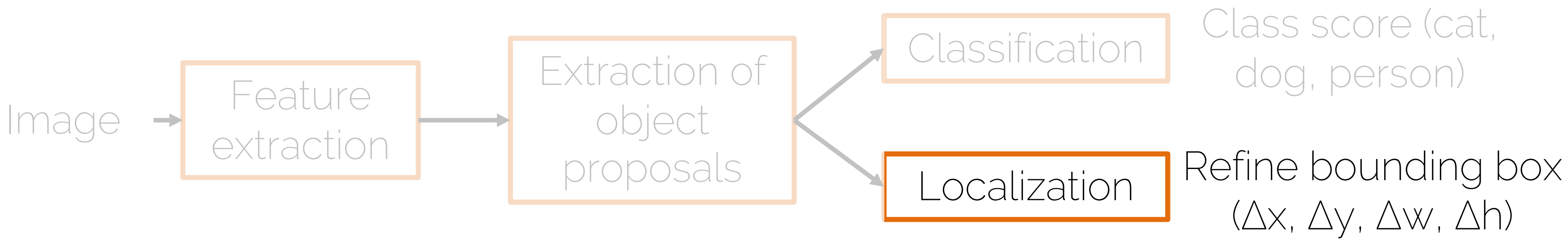
- One-stage detectors
  - YOLO, SSD, RetinaNet
  - CenterNet, CornerNet, ExtremeNet

- Two-stage detectors
  - R-CNN, Fast R-CNN, Faster R-CNN
  - SPP-Net, R-FCN, FPN

Two-stage detectors

Types of object detectors

- Two-stage detectors



Localization

- Bounding box regression

Image

Feature extraction (this time with a Neural Network)

Output:
Box coordinates (x,y,w,h)

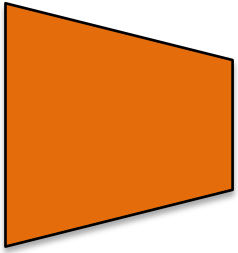
L2 loss function

Ground truth: Box coordinates

Localization

- Bounding box regression

Image



Convolutional
Neural Network

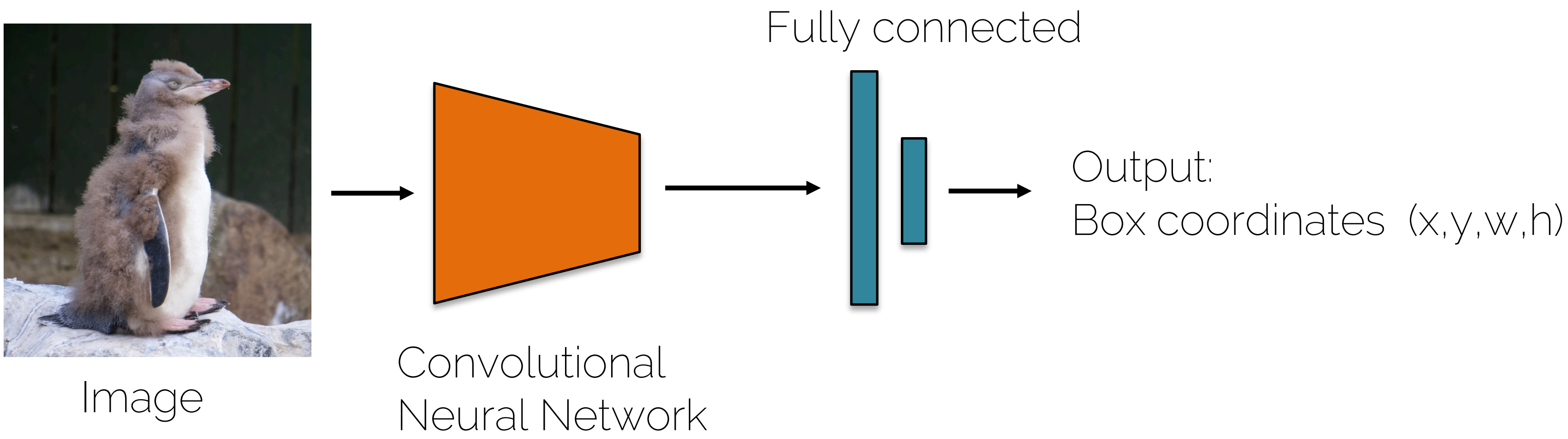
Output:
Box coordinates (x,y,w,h)

L2 loss function

Ground truth: Box
coordinates

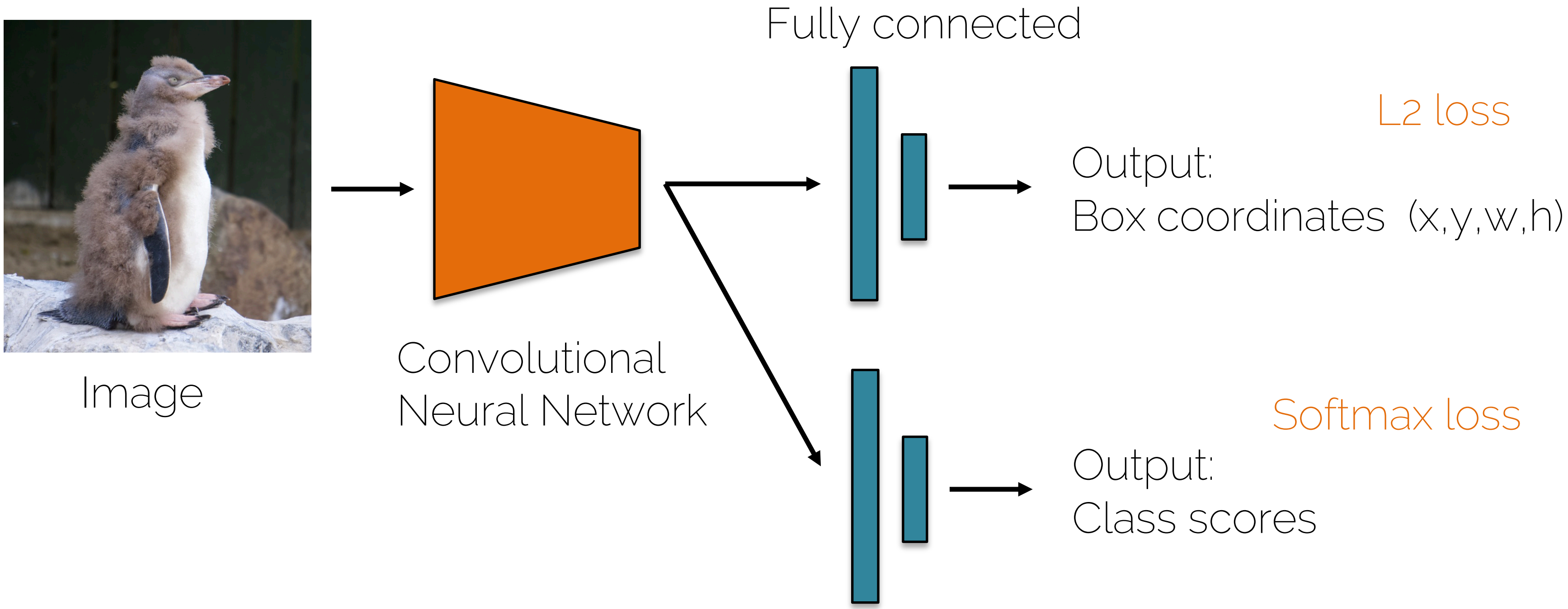
Localization and classification

- Bounding box regression



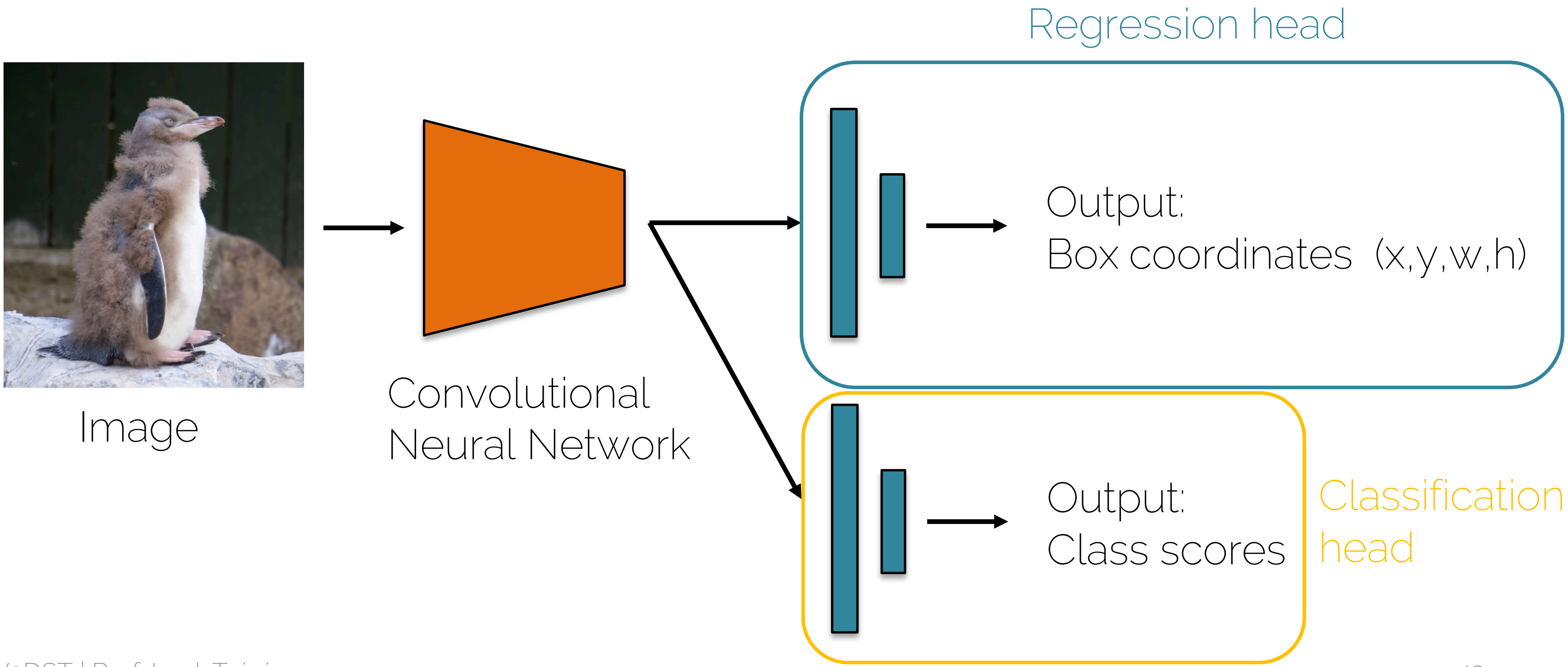
Localization and classification

- Bounding box regression



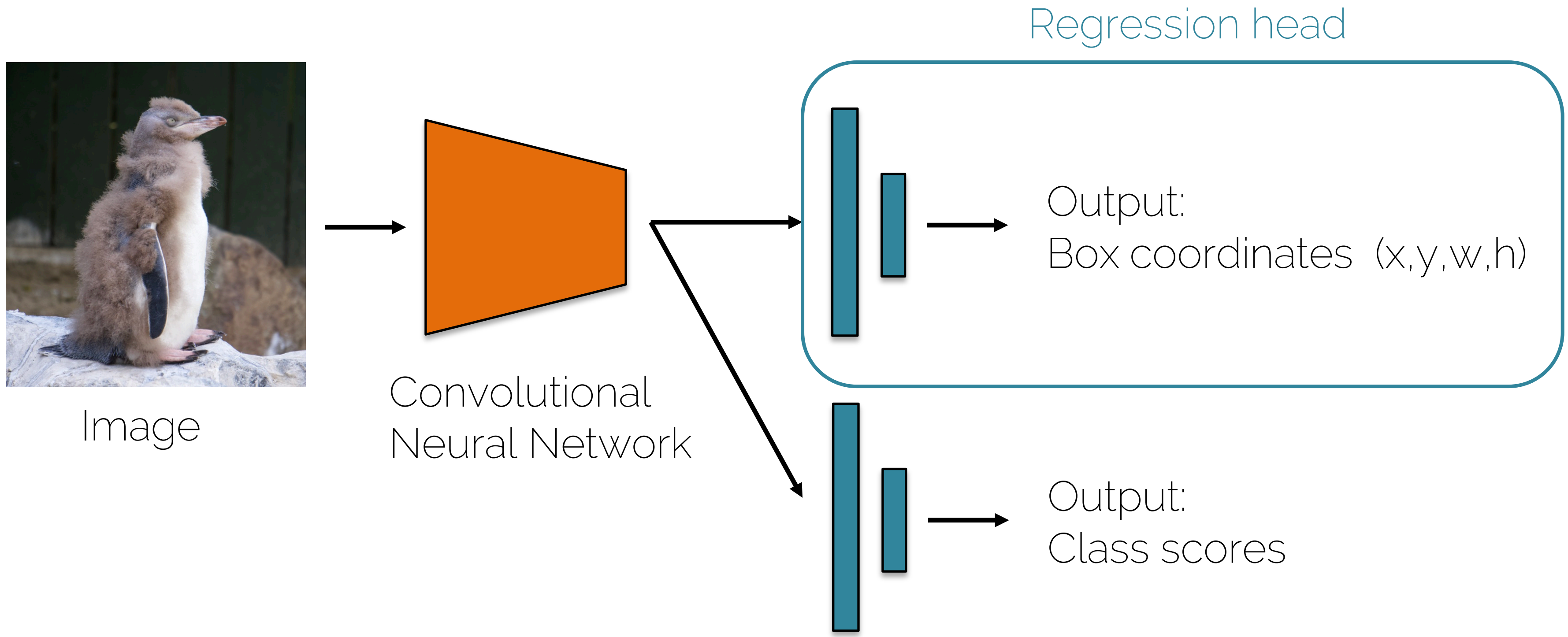
Localization and classification

- Bounding box regression



Localization and classification

- Bounding box regression



Localization and classification

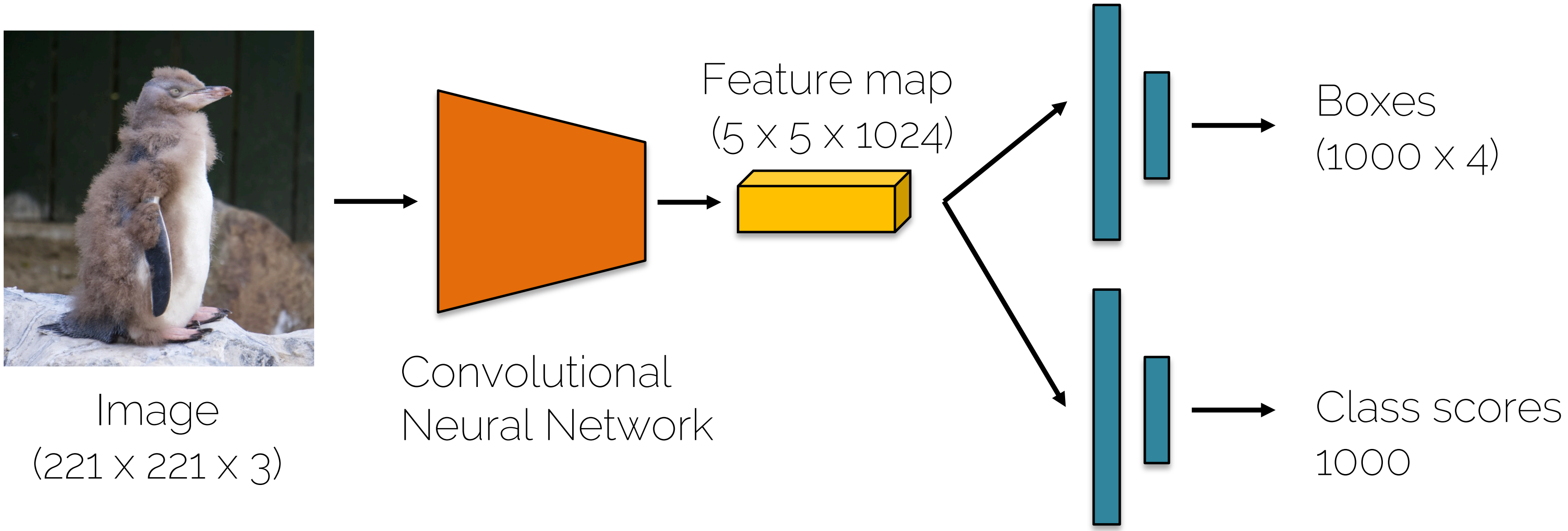
- It was typical to train the classification head first, freeze the layers
- Then train the regression head

- At test time, we use both!

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

Overfeat

- Sliding window + box regression + classification



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

Overfeat

- Sliding window + box regression + classification

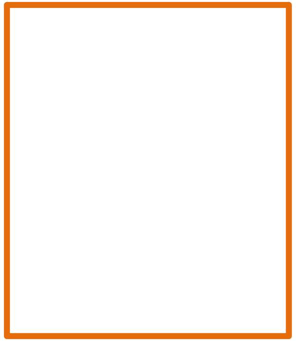


Image (468 x 356 x 3)

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

Overfeat

- Sliding window + box regression + classification

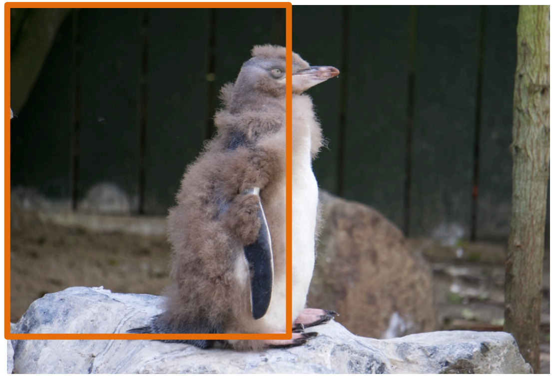


Image (468 x 356 x 3)

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

Overfeat

- Sliding window + box regression + classification

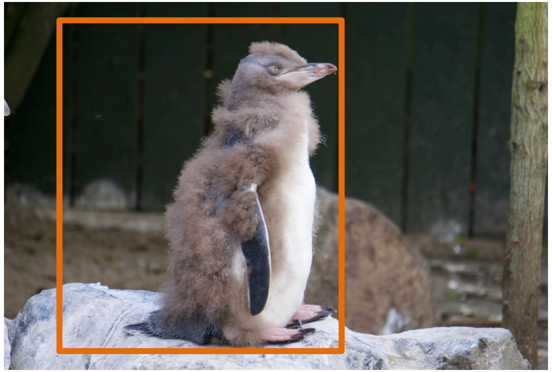


Image (468 x 356 x 3)

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

Overfeat

- Sliding window + box regression + classification

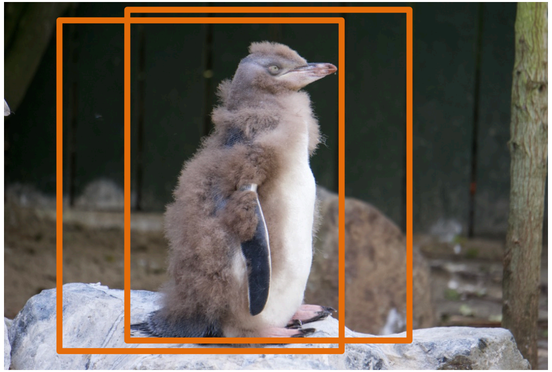


Image (468 x 356 x 3)

Sermanet et al, "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 (in Overfeat they have a greedy method)

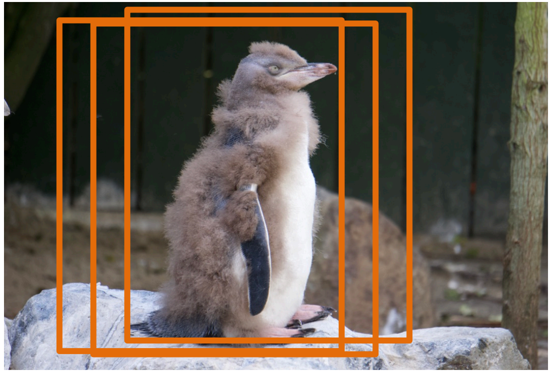


Image (468 x 356 x 3)

Sermanet et al, "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 (in Overfeat they have a greedy method)

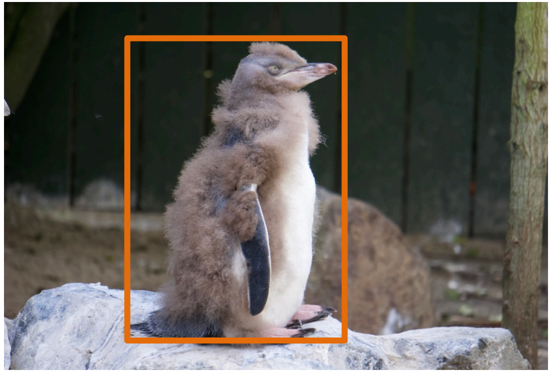


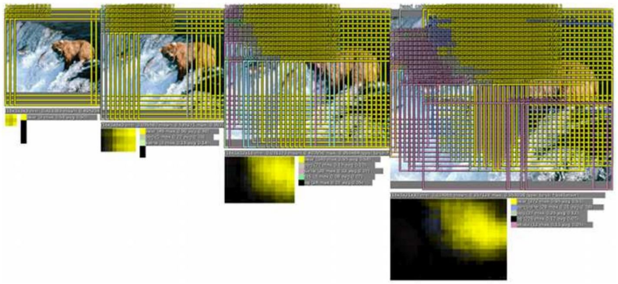
Image (468 x 356 x 3)

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

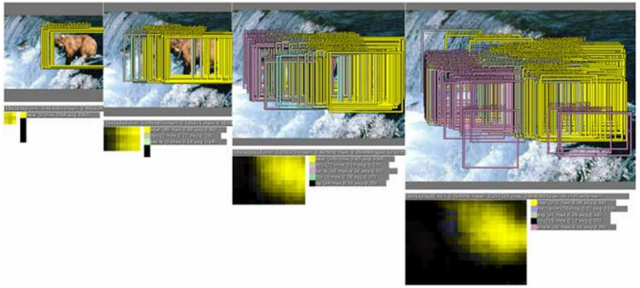
Overfeat

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

Window positions + score maps



Box regression outputs

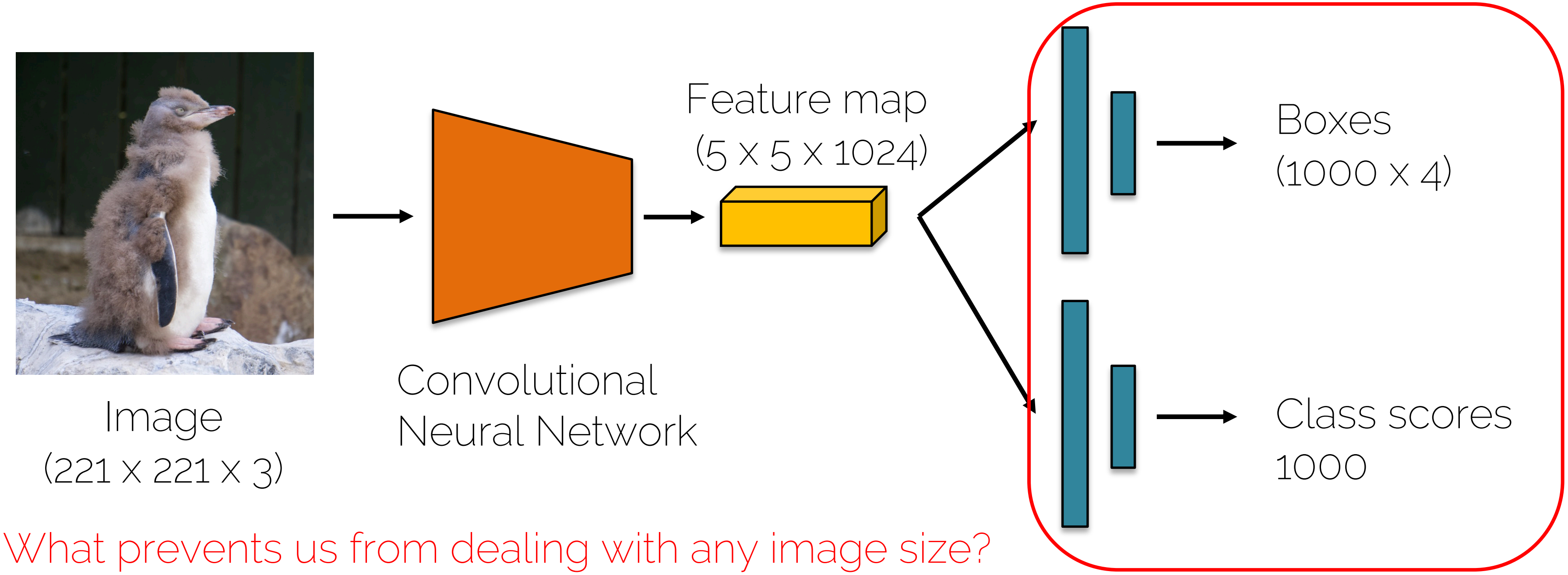


Final Predictions

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

Overfeat

- Sliding window + box regression + classification

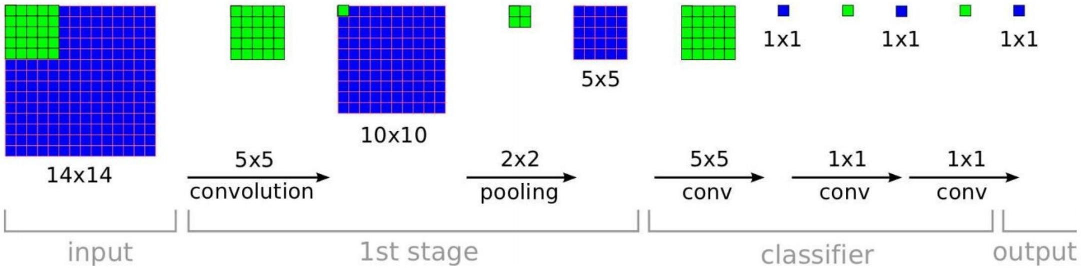


What prevents us from dealing with any image size?

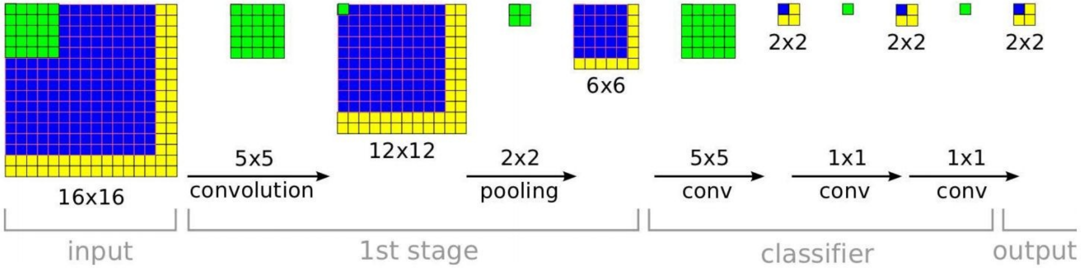
Sermanet et al, "Integrated Recognition, Localization and Detection using Convolutional Networks", ICLR 2014

Overfeat

Training time: Small image, 1 x 1 classifier output



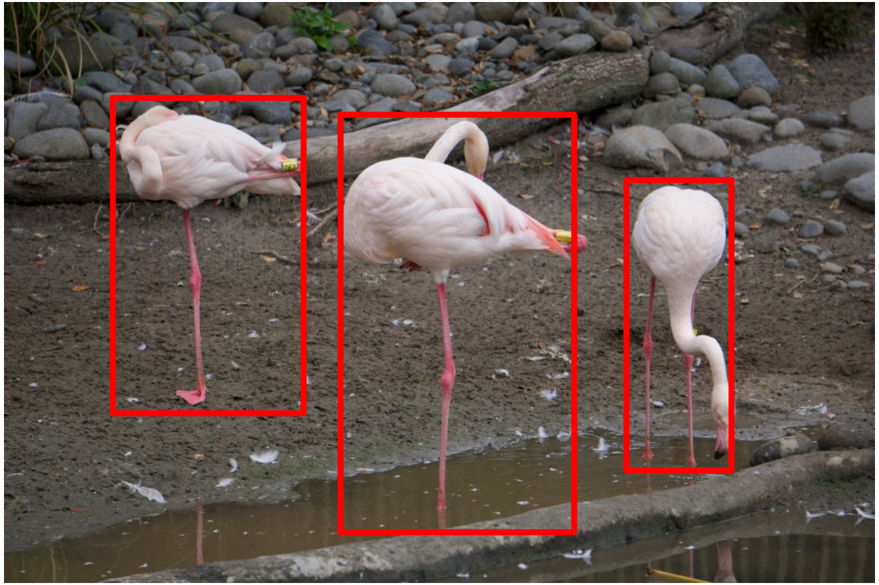
Test time: Larger image, 2 x 2 classifier output, only extra compute at yellow regions



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

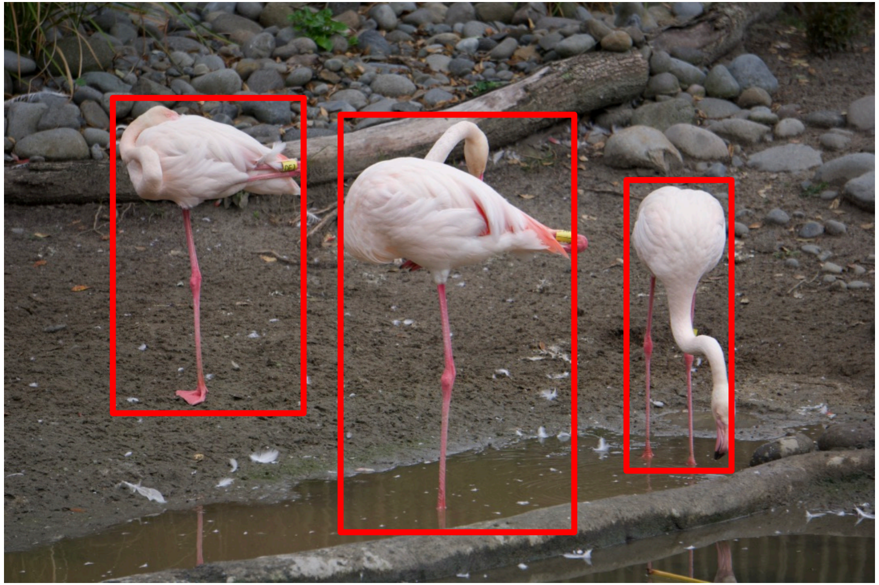
What about multiple objects?

- Localization: Regression
- How about detection?



What about multiple objects?

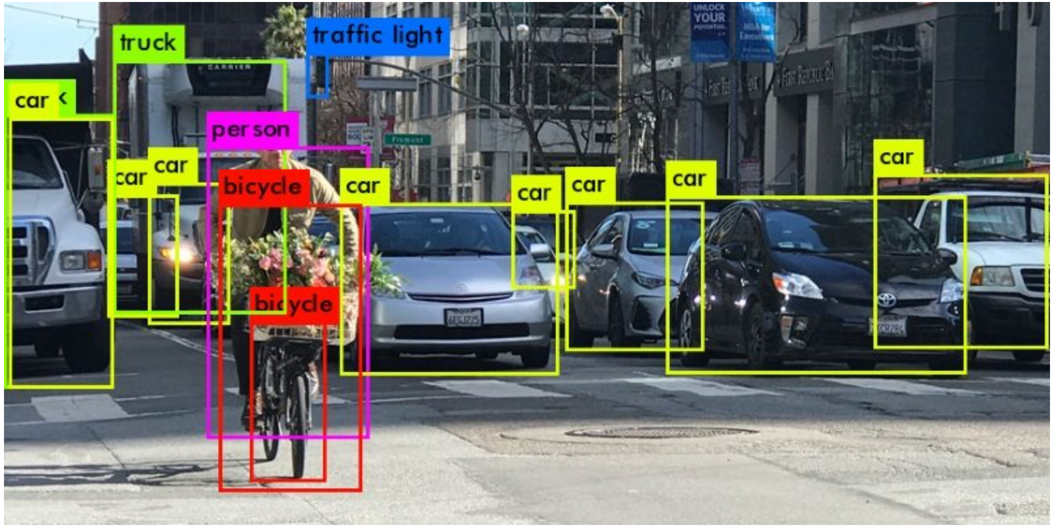
- Localization: Regression
- How about detection?



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

What about multiple objects?

- Localization: Regression
- How about detection?



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

What about multiple objects?

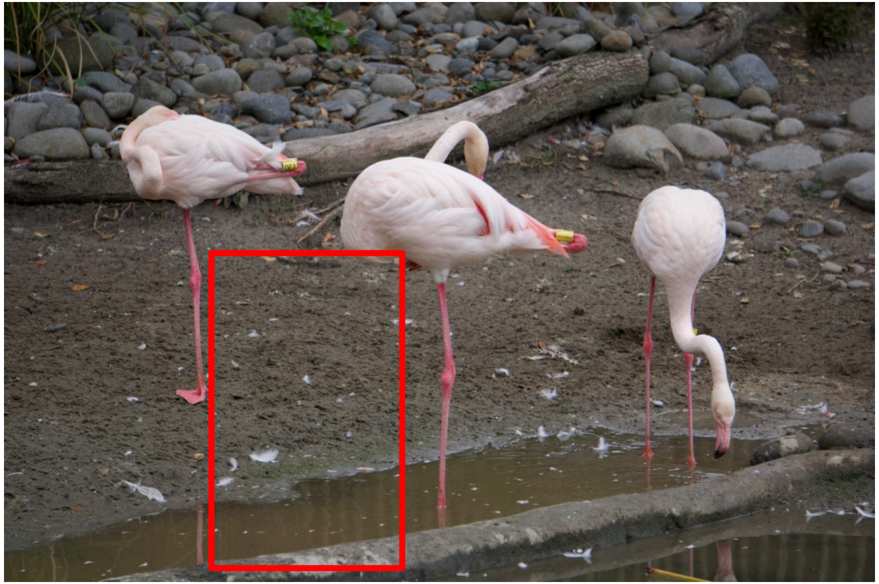
- Localization: Regression
- How about detection?

- Having a variable sized output is not optimal for Neural Networks

- There are a couple of workarounds:
  - RNN: Romera-Paredes and Torr. Recurrent Instance Segmentation. ECCV 2016.
  - Set prediction: Rezatofighi, Kaskman, Motlagh, Shi, Cremers, Leal-Taixé, Reid. Deep Perm-Set Net: Learn to predict sets with unknown permutation and cardinality using deep neural networks. Arxiv: 1805.00613

Detection as classification?

- Localization: Regression
- How about detection? Regression

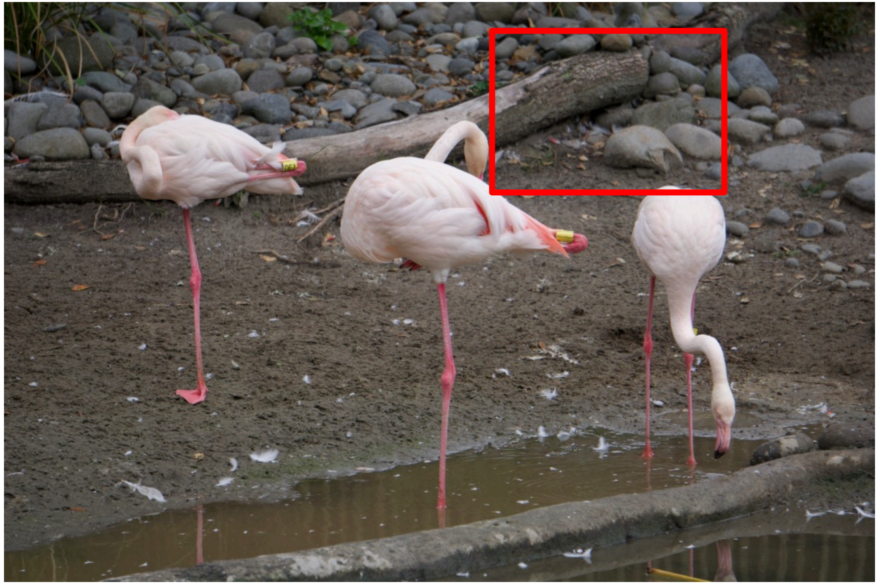


Is this a Flamingo?

NO

Detection as classification?

- Localization: Regression
- How about detection? Regression

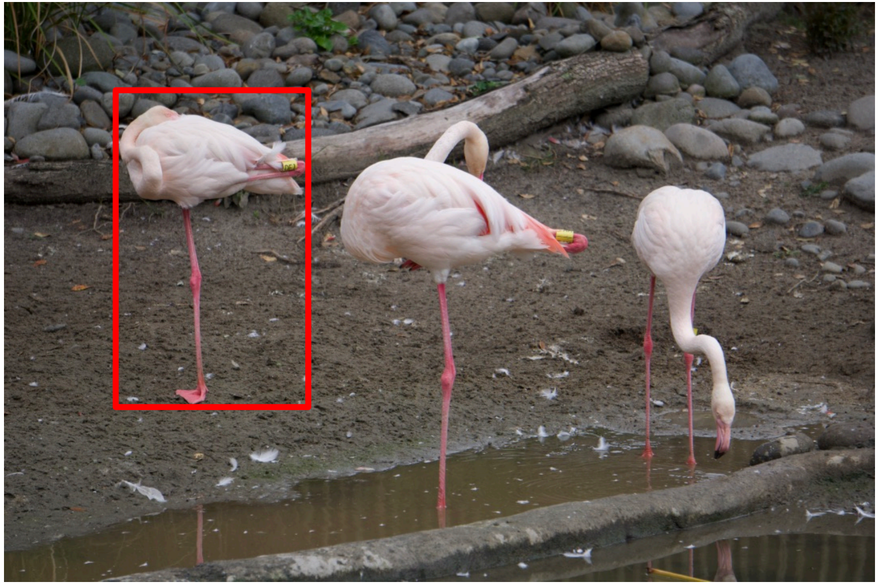


Is this a Flamingo?

NO

Detection as classification?

- Localization: Regression
- How about detection? Regression



Is this a Flamingo?

YES!

Detection as classification?

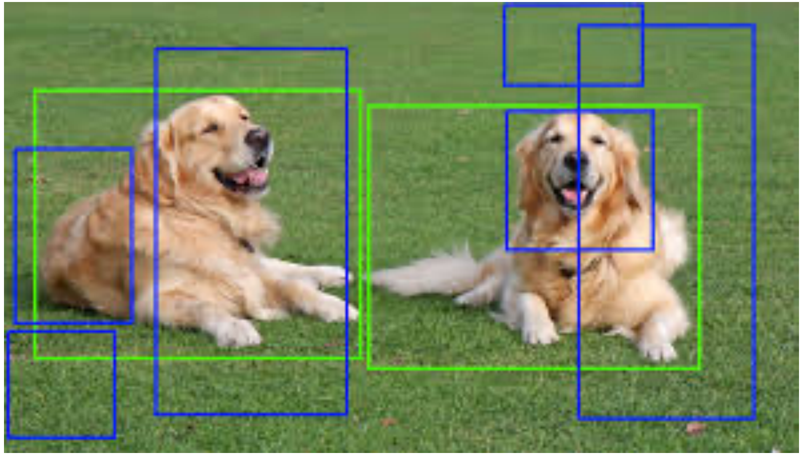
- Localization: Regression
- How about detection? Classification

- Problem:
  - Expensive to try all possible positions, scales and aspect ratios
  - How about trying only on a subset of boxes with most potential?

Region Proposals

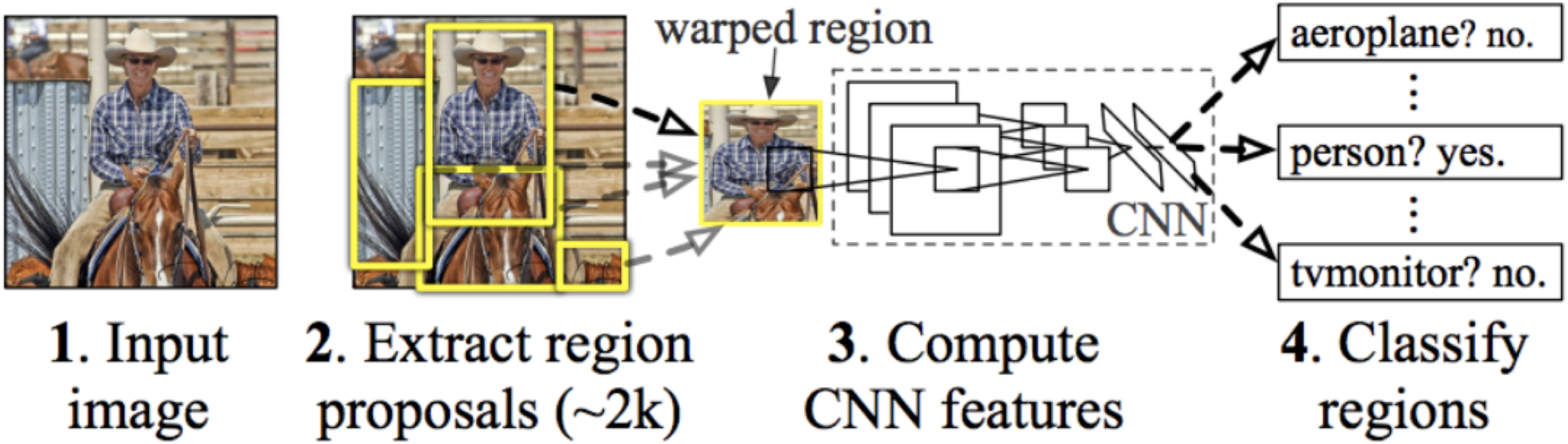
- We have already seen a method that gives us "interesting" regions in an image that potentially contain an object

- Step 1: Obtain region proposals
- Step 2: Classify them.



The R-CNN family

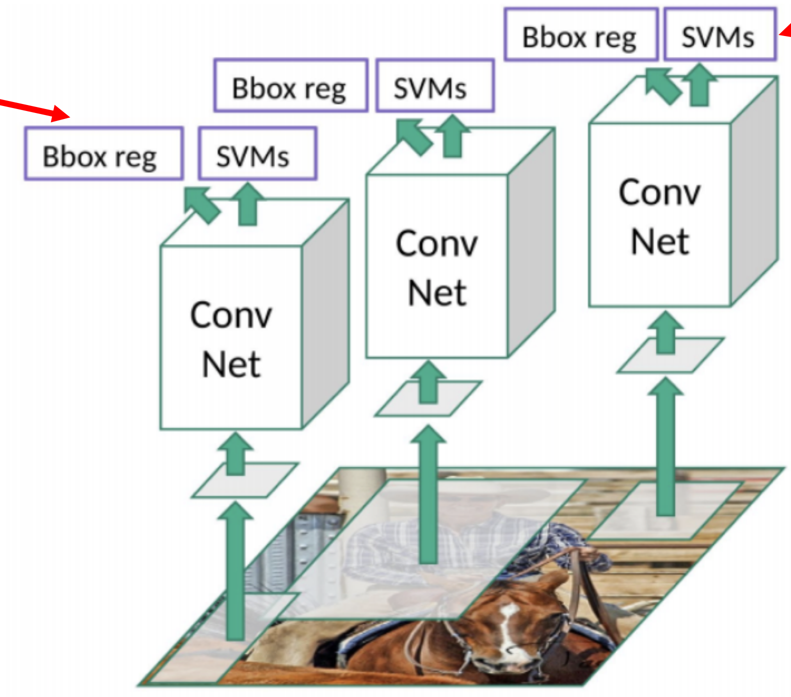
R-CNN



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

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

R-CNN

- Training scheme:
  - 1. Pre-train the CNN on ImageNet
  - 2. Finetune the CNN on the number of classes the detector is aiming to classify (softmax loss)
  - 3. Train a linear Support Vector Machine classifier to classify image regions. One SVM per class! (hinge loss)
  - 4. Train the bounding box regressor (L2 loss)

R-CNN

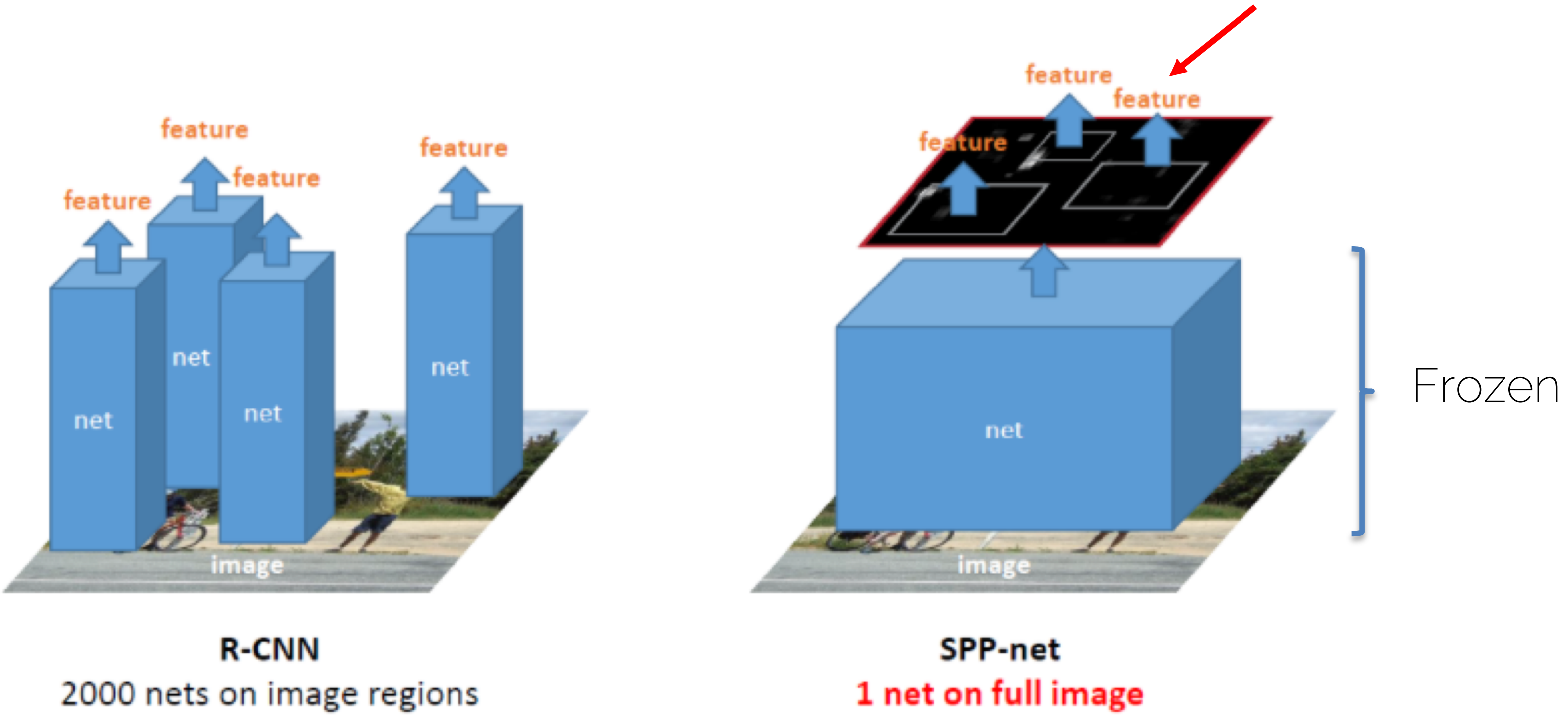
- PROS:
  - The pipeline of proposals, feature extraction and SVM classification is well-known and tested. Only features are changed (CNN instead of HOG).
  - CNN summarizes each proposal into a 4096 vector (much more compact representation compared to HOG)
  - Leverage transfer learning: the CNN can 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 → not exploiting learning to its full potential.

SPP-Net

How do we "pool" these features into a common size



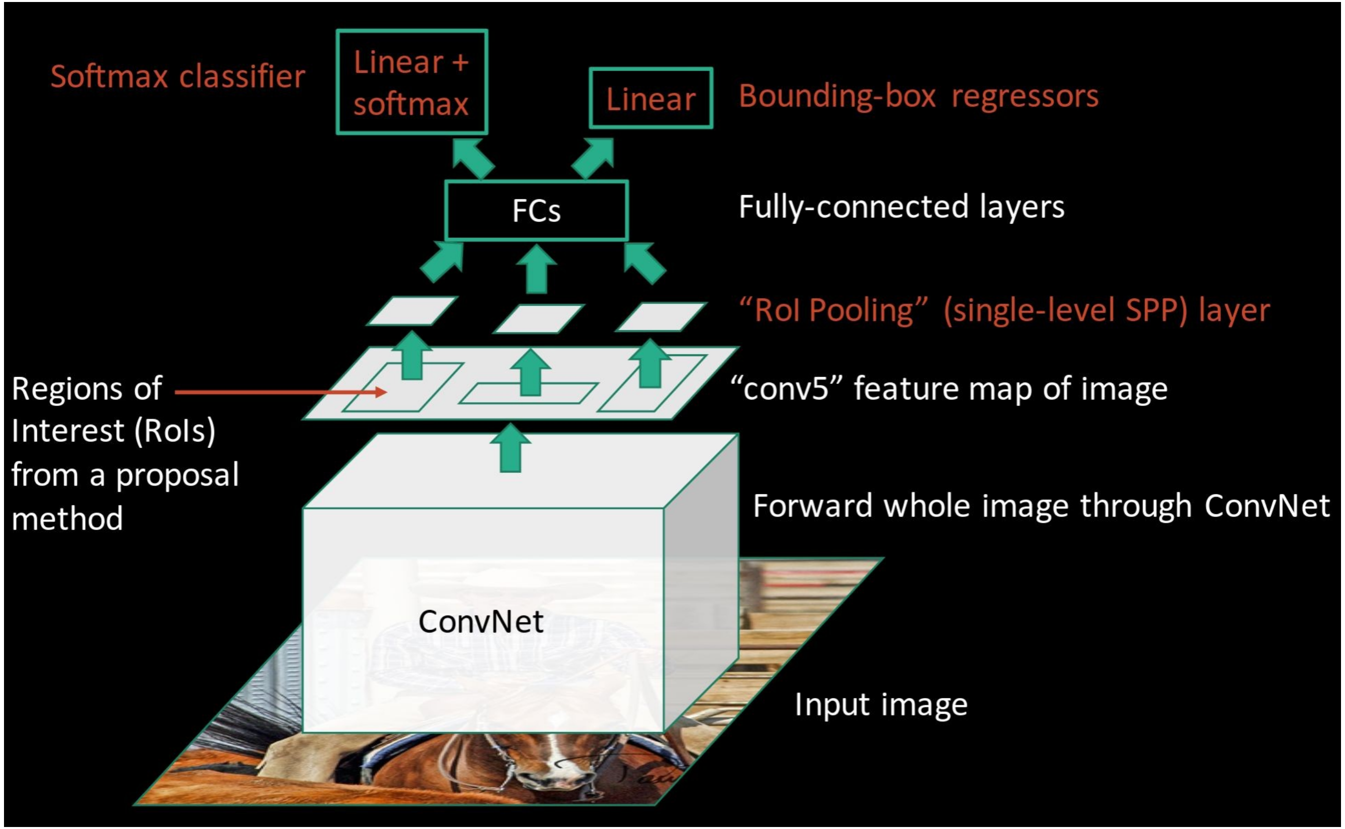
He et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. ECCV 2014.

SPP-Net

- It solved the R-CNN problem of being slow at test time
- It still has some problems inherited from R-CNN:
  - Training is still slow (a bit faster than R-CNN)
  - Training scheme is still complex
  - Still no end-to-end training

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

Fast R-CNN

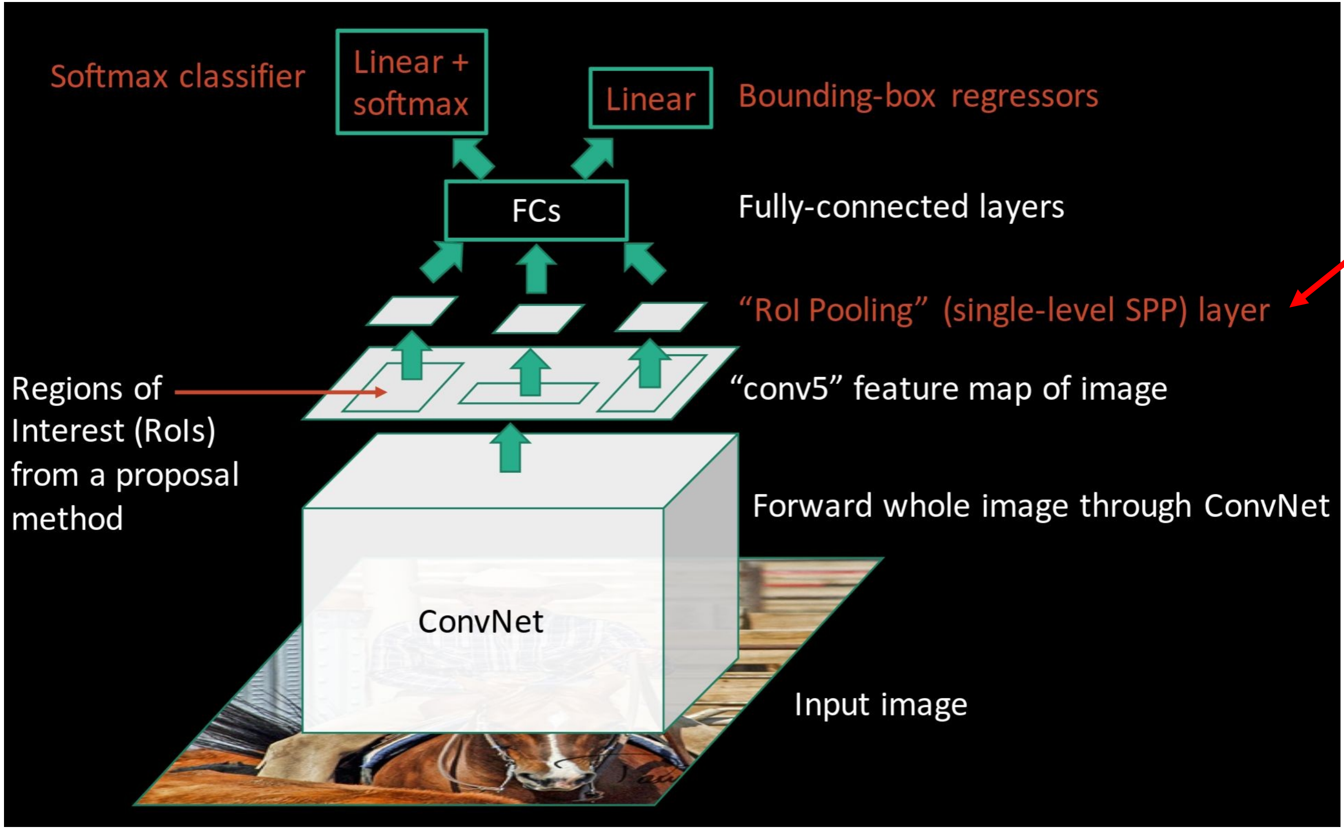


Shared computation at test time (like SPP)

Girschick, "Fast R-CNN", ICCV 2015

Slide credit: Ross Girschick

Fast R-CNN



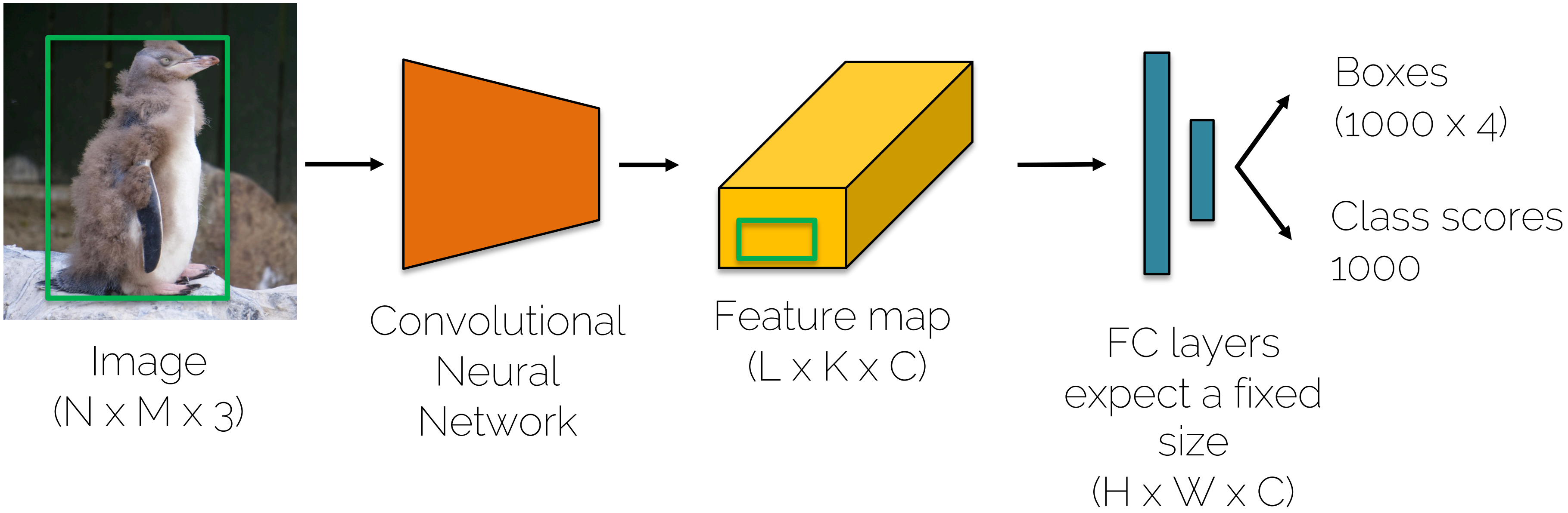
Region of Interest Pooling

Girschick, "Fast R-CNN", ICCV 2015

Slide credit: Ross Girschick

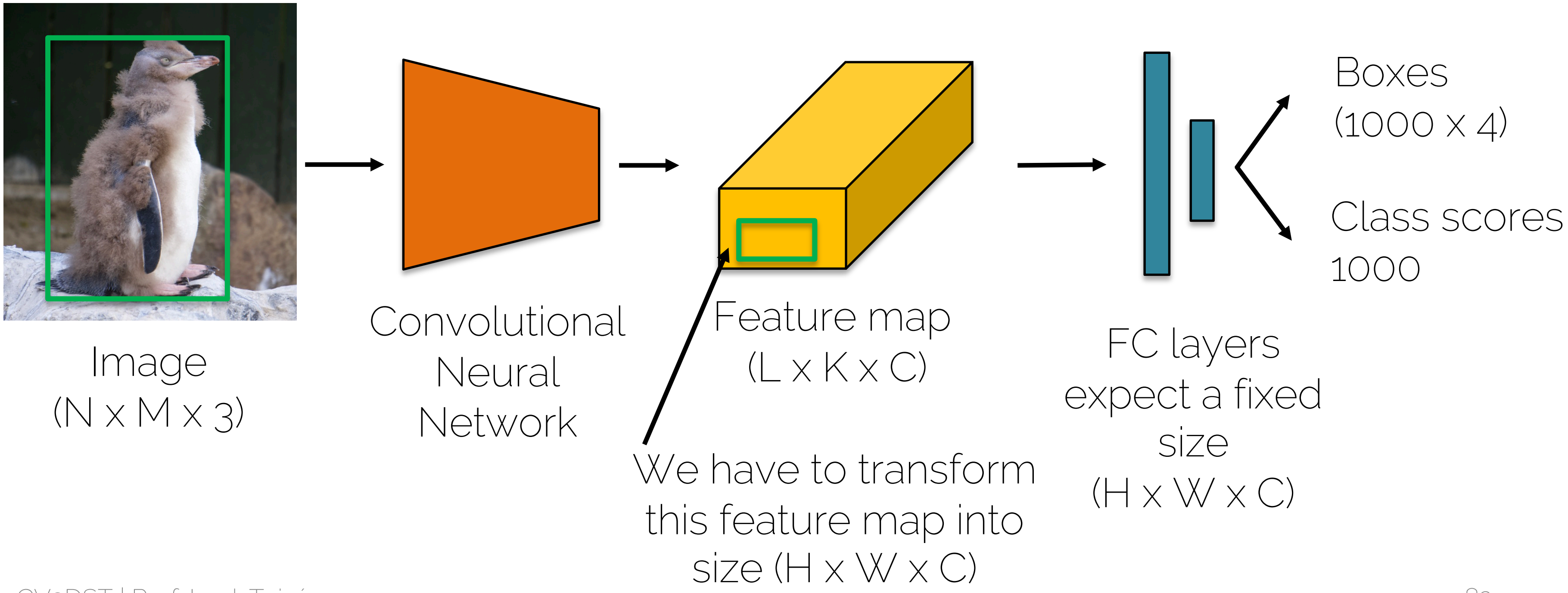
Fast R-CNN: RoI Pooling

- Region of Interest Pooling



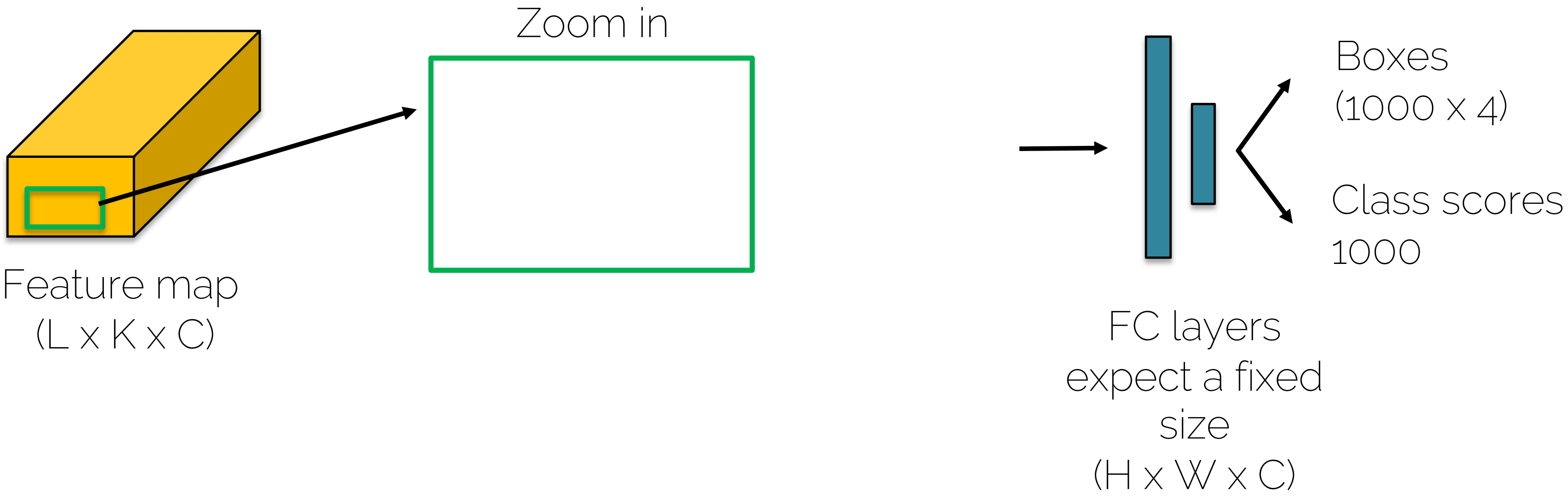
Fast R-CNN: RoI Pooling

- Region of Interest Pooling



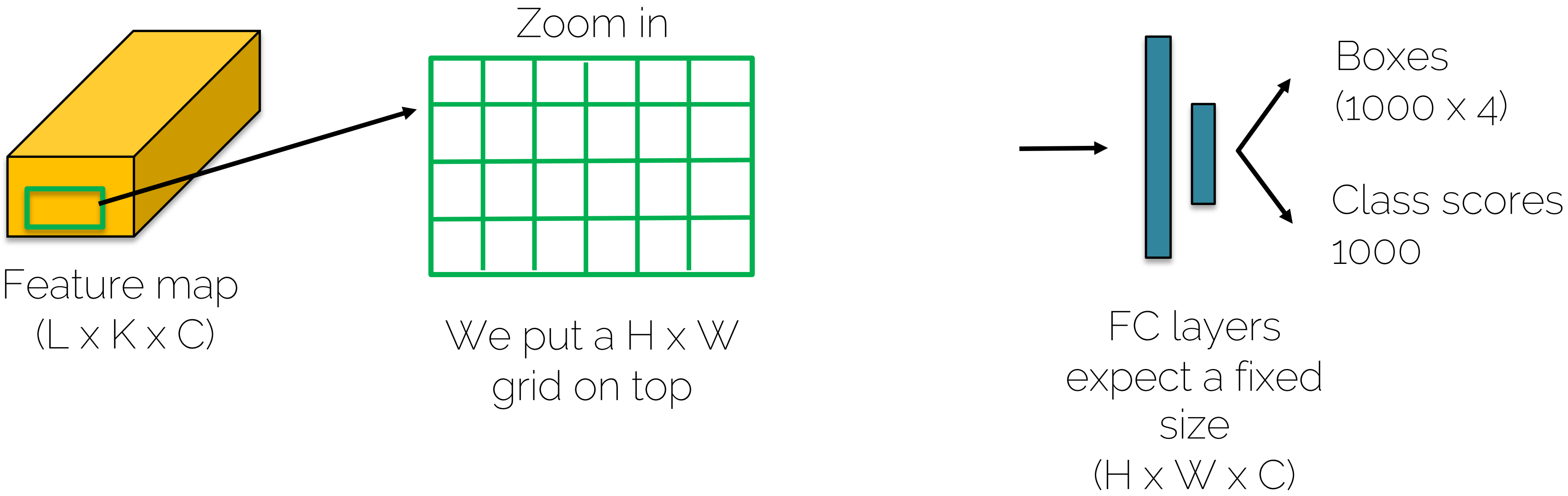
Fast R-CNN: RoI Pooling

- Region of Interest Pooling



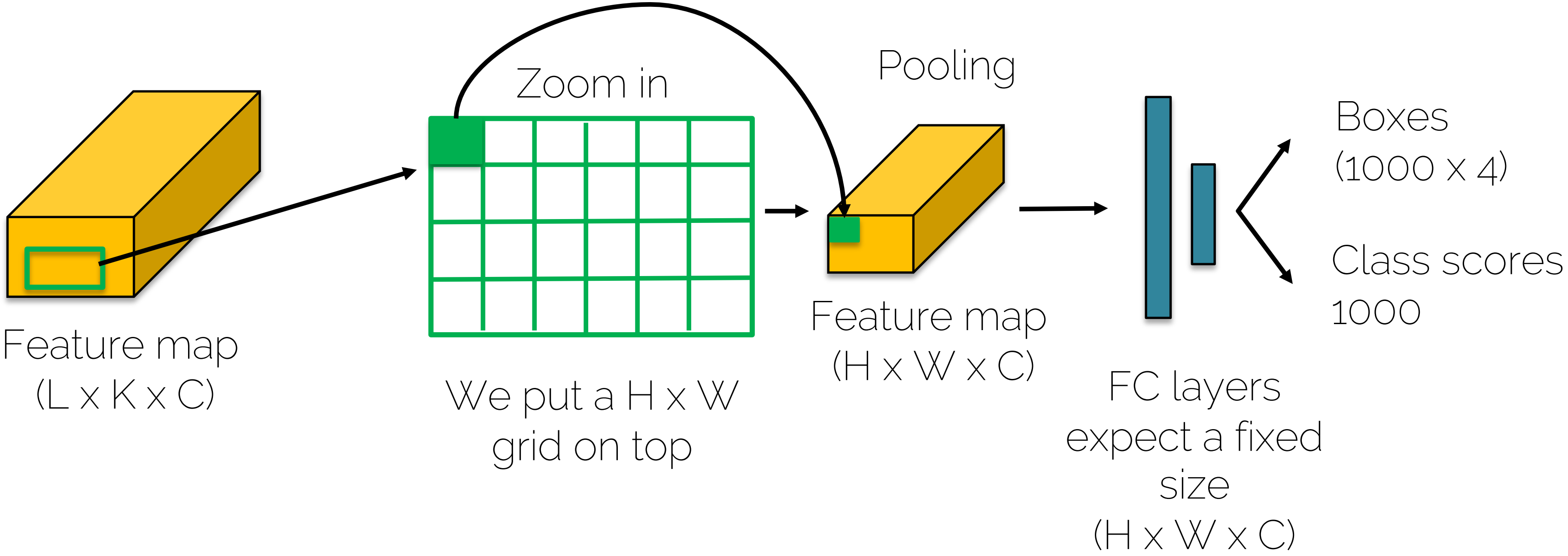
Fast R-CNN: RoI Pooling

- Region of Interest Pooling



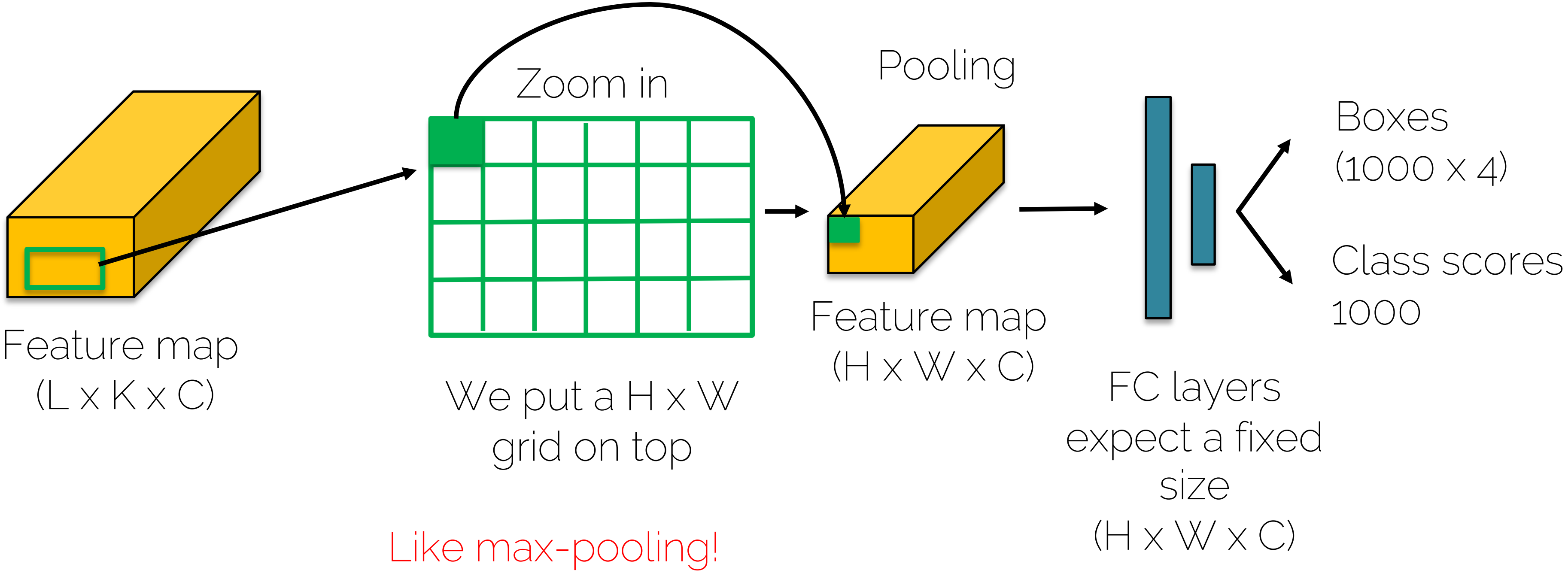
Fast R-CNN: RoI Pooling

- Region of Interest Pooling



Fast R-CNN: RoI Pooling

- RoI Pooling: how do you do backpropagation?



Fast R-CNN Results

- VGG-16 CNN on Pascal VOC 2007 dataset

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

Fast R-CNN Results

- VGG-16 CNN on Pascal VOC 2007 dataset

		R-CNN	Fast R-CNN
Faster!	Training Time:	84 hours	9.5 hours
	(Speedup)	1x	8.8x
FASTER!	Test time per image	47 seconds	0.32 seconds
	(Speedup)	1x	146x

Fast R-CNN Results

- VGG-16 CNN on Pascal VOC 2007 dataset

		R-CNN	Fast R-CNN
Faster!	Training Time:	84 hours	9.5 hours
	(Speedup)	1x	8.8x
FASTER!	Test time per image	47 seconds	0.32 seconds
	(Speedup)	1x	146x
Better!	mAP (VOC 2007)	66.0	66.9

Fast R-CNN Results

The test times do not include proposal generation!

- VGG-16 CNN on Pascal VOC 2007 dataset

		R-CNN	Fast R-CNN
Faster!	Training Time:	84 hours	9.5 hours
	(Speedup)	1x	8.8x
FASTER!	Test time per image	47 seconds	0.32 seconds
	(Speedup)	1x	146x
Better!	mAP (VOC 2007)	66.0	66.9

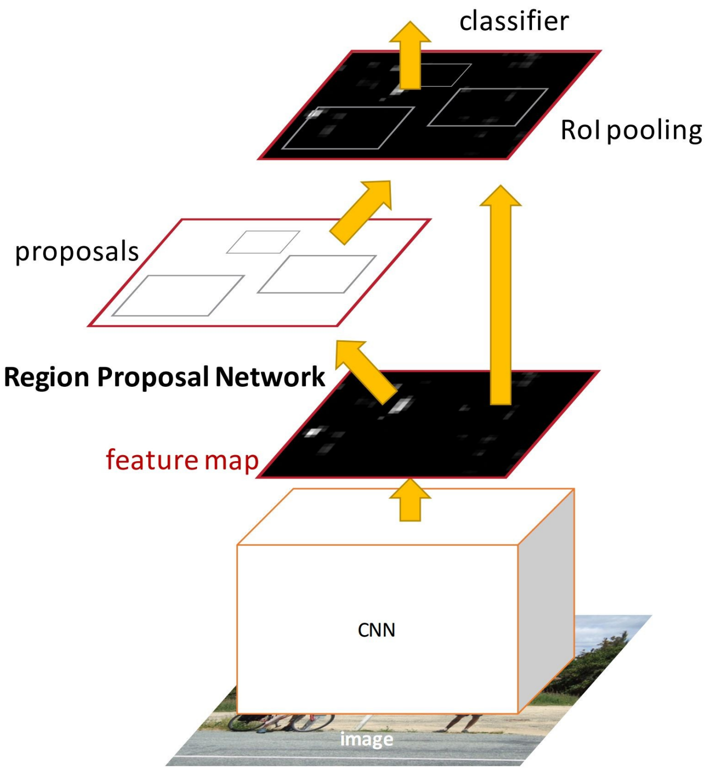
Fast R-CNN Results

With proposals included

- VGG-16 CNN on Pascal VOC 2007 dataset

		R-CNN	Fast R-CNN
Faster!	Training Time:	84 hours	9.5 hours
	(Speedup)	1x	8.8x
FASTER!	Test time per image	50 seconds	2 seconds
	(Speedup)	1x	25x
Better!	mAP (VOC 2007)	66.0	66.9

Faster R-CNN:



- Solution: Have the proposal generation integrated with the rest of the pipeline
- Region Proposal Network (RPN) trained to produce region proposals directly.
- After RPN, everything is like Fast R-CNN

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

Slide credit: Ross Girschick

Next lectures

- How does a Region Proposal Network work?
- One-stage detectors

- Next lecture is on November 29th!

- Details of the exercise will follow soon.