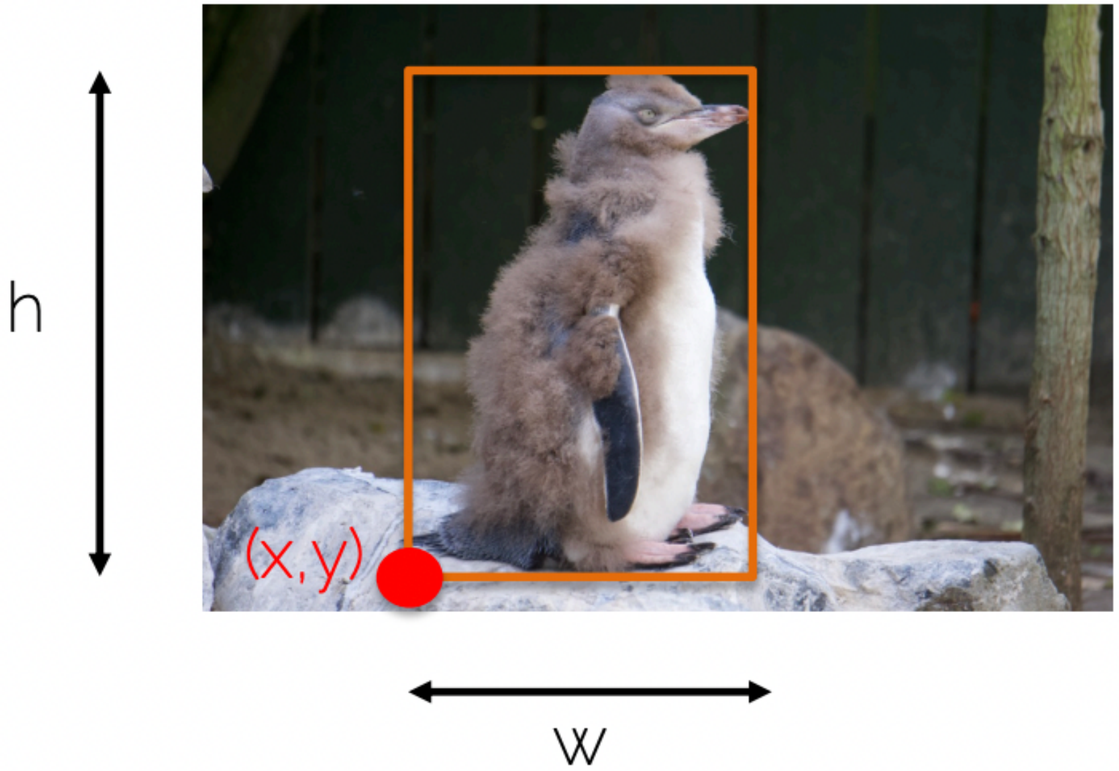
Object Detection with Deep Learning

S22.CS7.505 Computer Vision
International Institute of Information Technology Hyderabad

March 25, 2022 | 1400 - 1530

Naren Akash R J | Center for Visual Information Technology

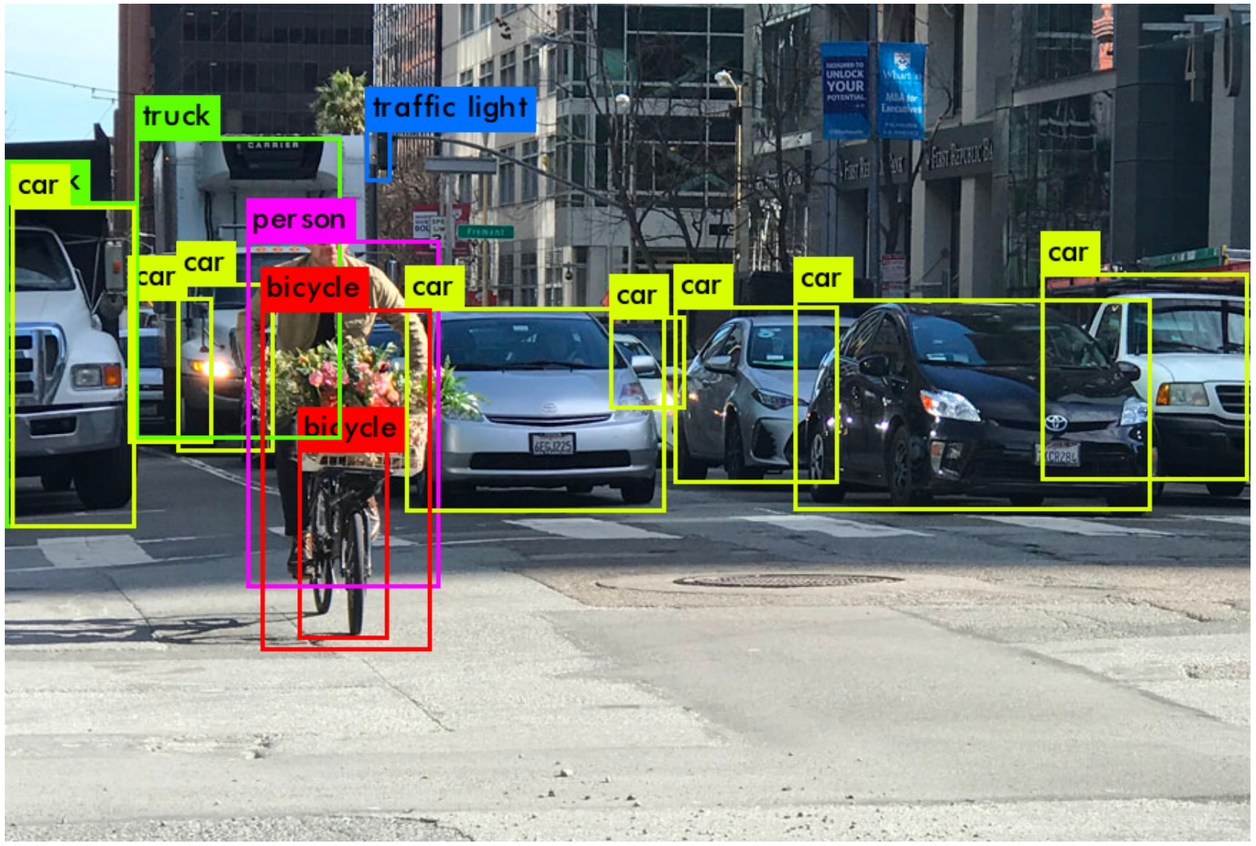
Task Definition



Object detection problem

Bounding box (x, y, w, h)

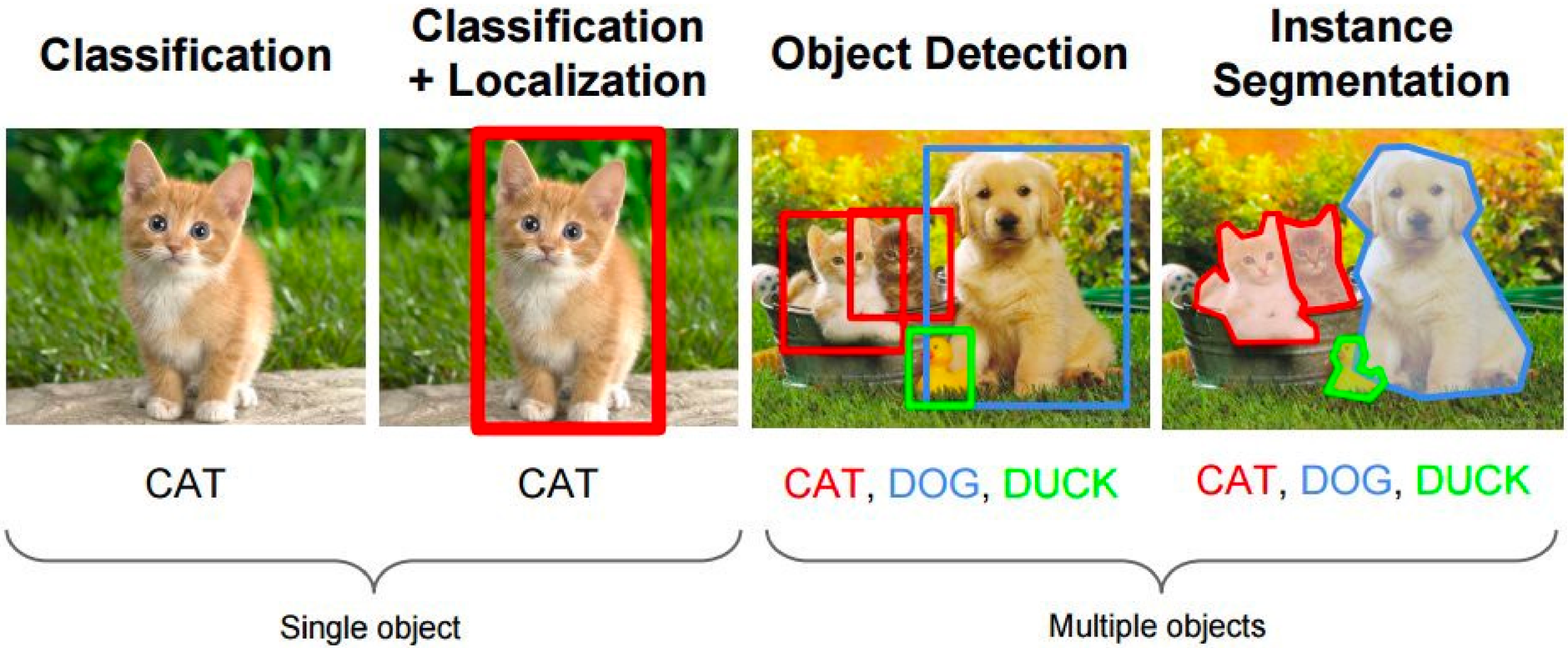
Task Definition



Object detection problem

Bounding box (x, y, w, h)
+ Class

Some Computer Vision Tasks



Classification + Localization

Classification: C classes
Input: Image
Output: Class label
Evaluation metric: Accuracy

CAT

Localization:
Input: Image
Output: Box in the image (x, y, w, h)
Evaluation metric: Intersection over Union

(x, y, w, h)

Classification + Localization: Do both

A bit of history

Traditional object detection methods

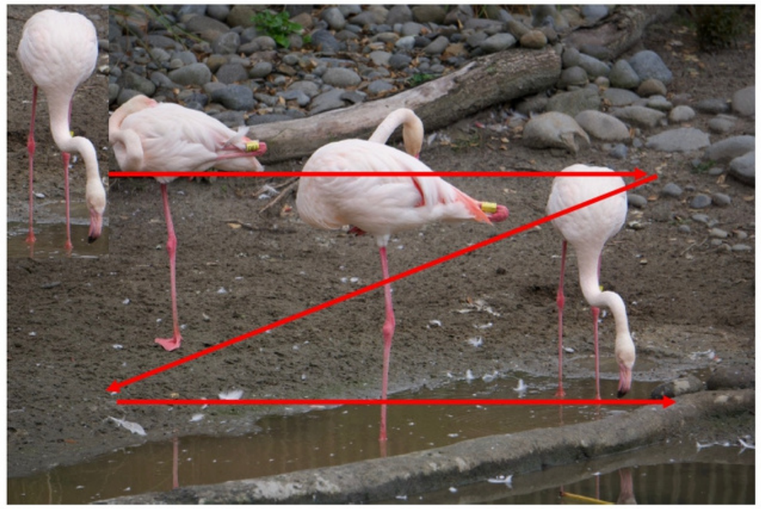
- 1. Template matching + sliding window

Image

Template

Traditional object detection methods

- 1. Template matching + sliding window



Image

Traditional object detection methods

- 1. Template matching + sliding window

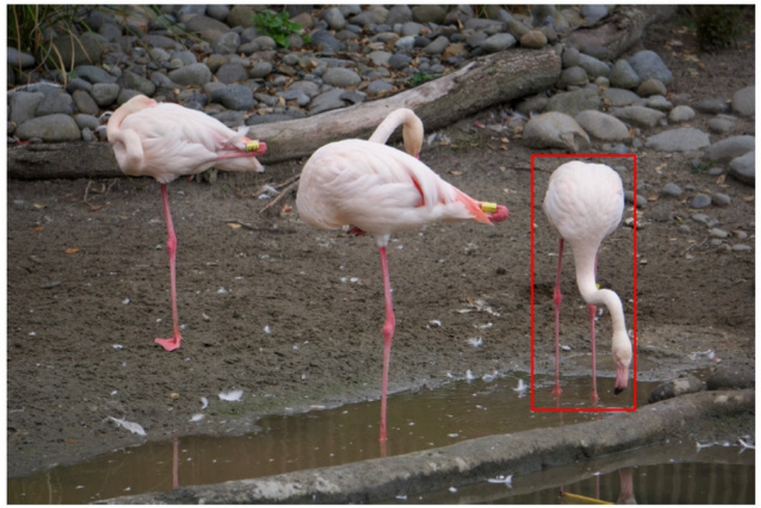
LOW correlation

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

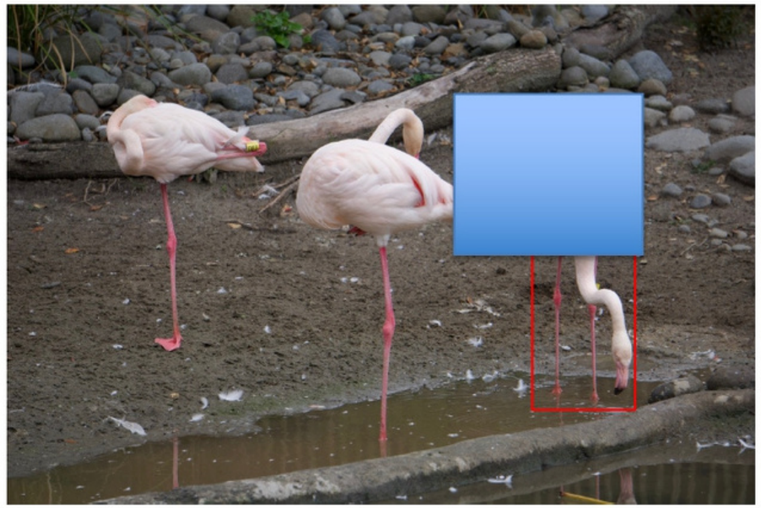
- Problems of 1. Template matching + sliding window
  - Occlusions: we need to see the WHOLE object
  - This works to detect a given **instance** of an object but not a **class** of objects

Appearance and shape changes

Pose changes

Traditional object detection methods

- Problems of 1. Template matching + sliding window



Image

LOW correlation

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

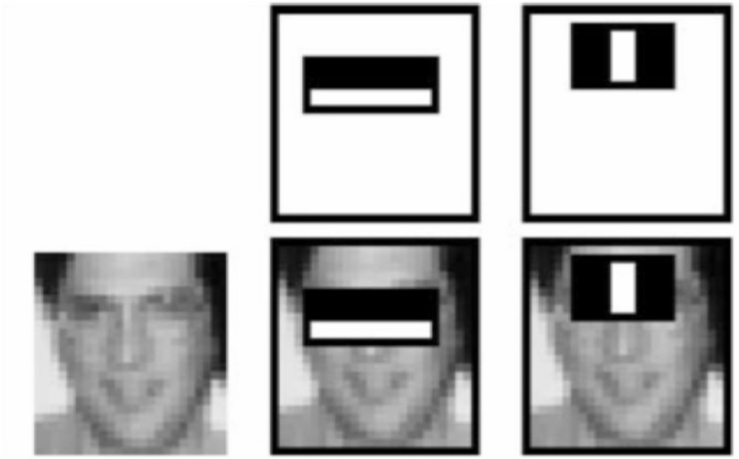
Traditional object detection methods

- 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
  - Objects have an unknown position, scale and aspect ratio, the search space is searched inefficiently with sliding window

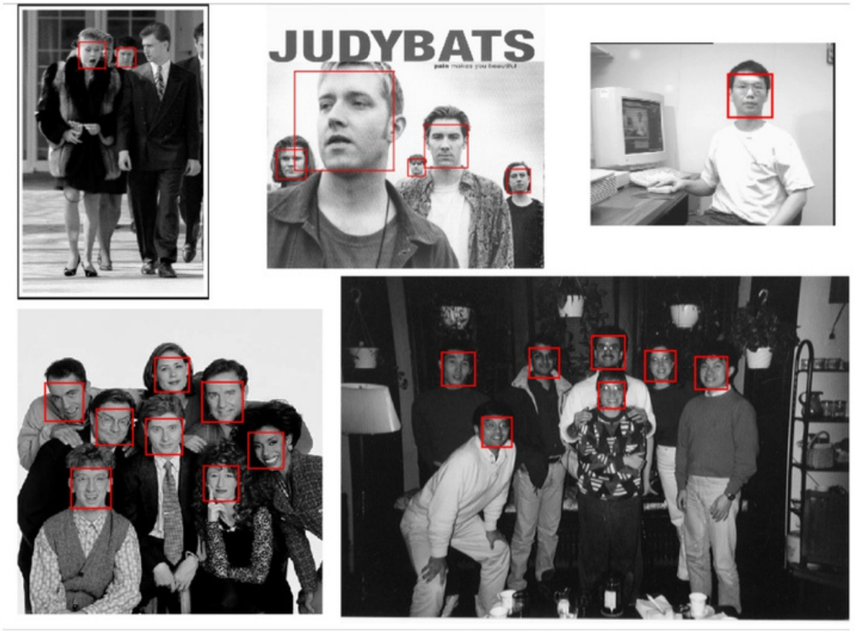
Viola-Jones Detector

- 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
  - 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-Jones Detector



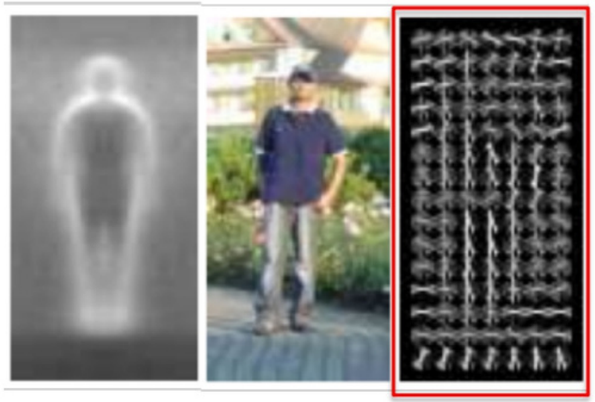
Haar features



P Viola and M Jones, "Rapid object detection using a boosted cascade of simple features", CVPR 2001

Histogram of Oriented Gradients

- 2. Feature extraction + classification

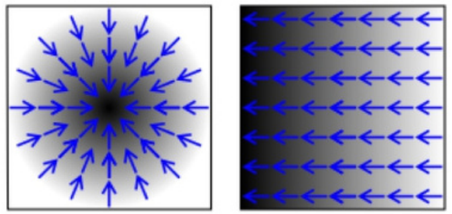


HOG descriptor → Histogram of oriented gradients.
Compute gradients in dense grids, compute gradients and create a histogram based on gradient direction.

N Dalal and B Triggs, "Histograms of oriented gradients for human detection", CVPR 2005

Histogram of Oriented Gradients

- 2. Feature extraction + classification



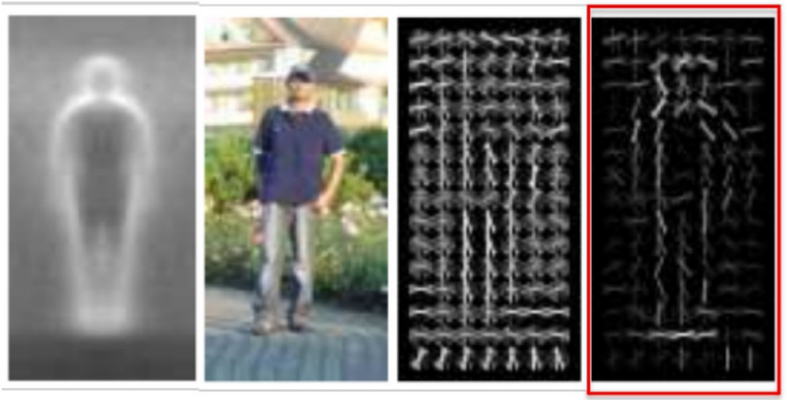
Gradient: blue arrows show the gradient, i.e., the direction of greatest change of the image.

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

N Dalal and B Triggs, "Histograms 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.

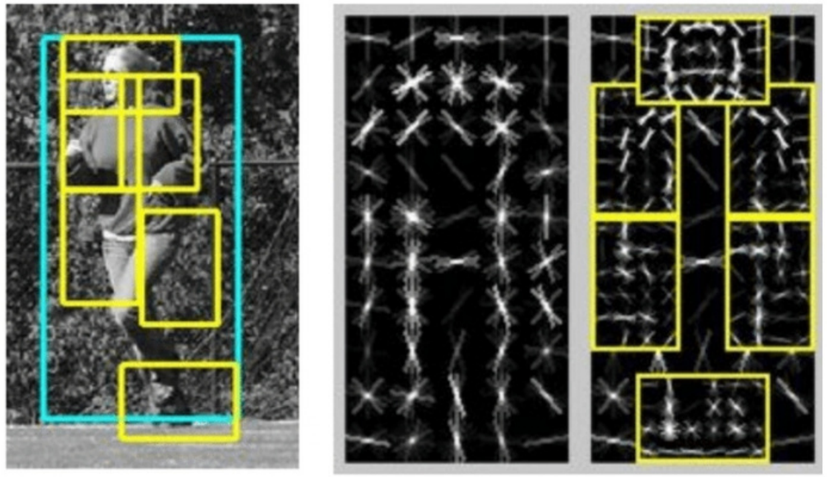
N Dalal and B Triggs, "Histograms of oriented gradients for human detection", CVPR 2005

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).

Deformable Part Model

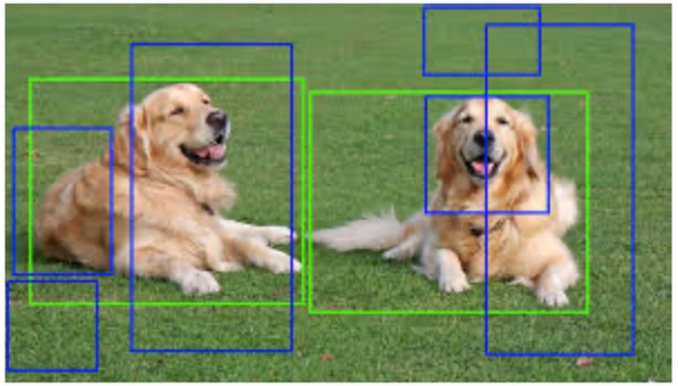
- Also based on HOG features, but based on body part detection → more robust to different body poses



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

What defines an object?

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



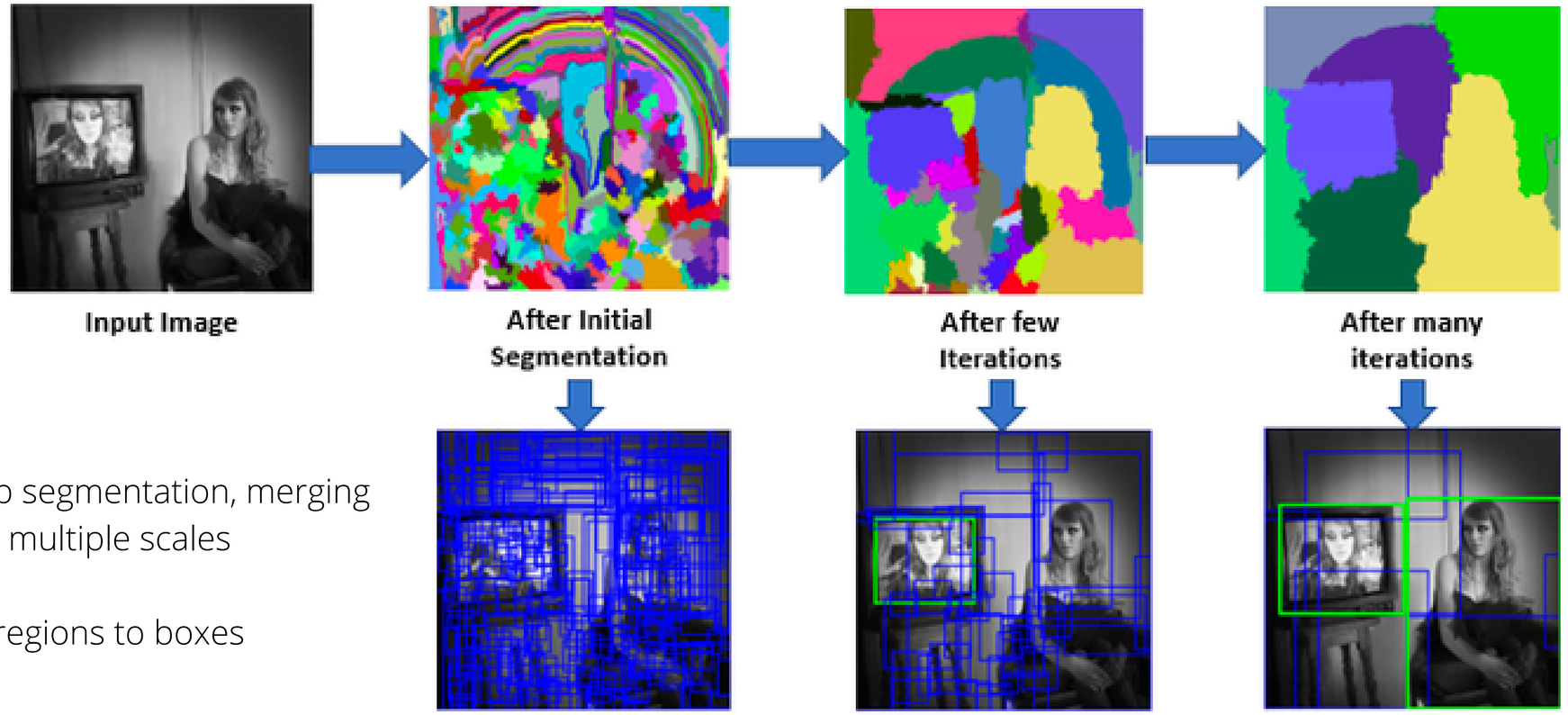
Very likely to be an object

Maybe it is an object

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.

Selective Search



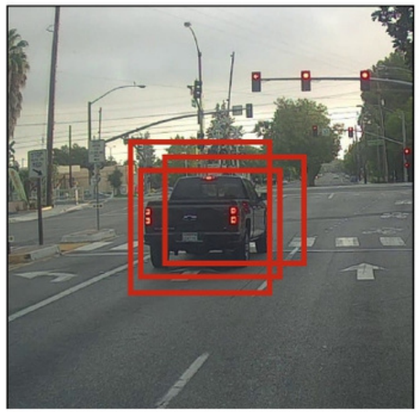
Bottom-up segmentation, merging regions at multiple scales

Converts regions to boxes

Uijlings et al, "Selective Search for Object Recognition", IJCV 2013

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

```
1: procedure NMS(B,c)
2:     B_nms ← ∅
3:     for b_i ∈ B do
4:         discard ← False
5:         for b_j ∈ B do
6:             if same(b_i, b_j) > λ_nms then
7:                 if score(c, b_j) > score(c, b_i) then
8:                     discard ← True
9:         if not discard then
10:            B_nms ← B_nms ∪ b_i
11:    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

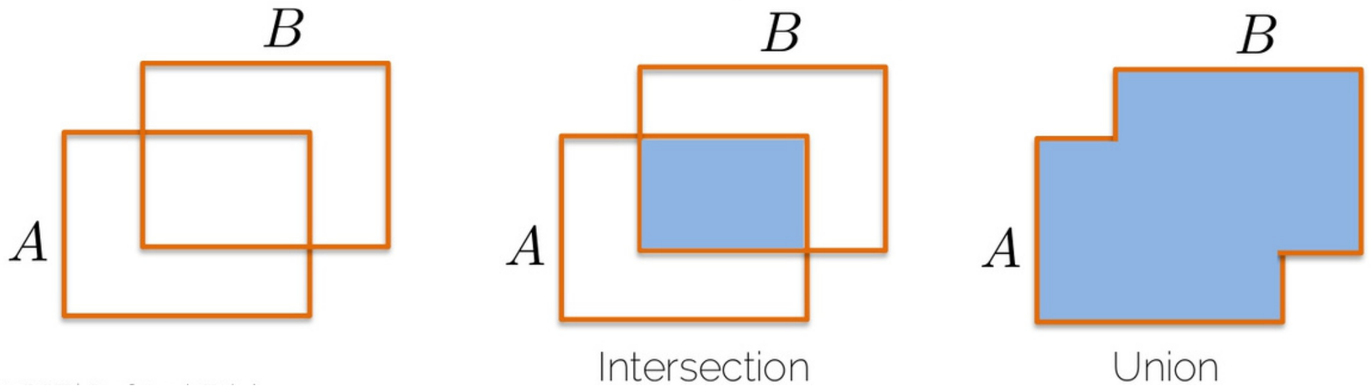
Overlap = to be defined

Score = depends on the task

Region Overlap

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

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



Non-Maximum Suppression (NMS)

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

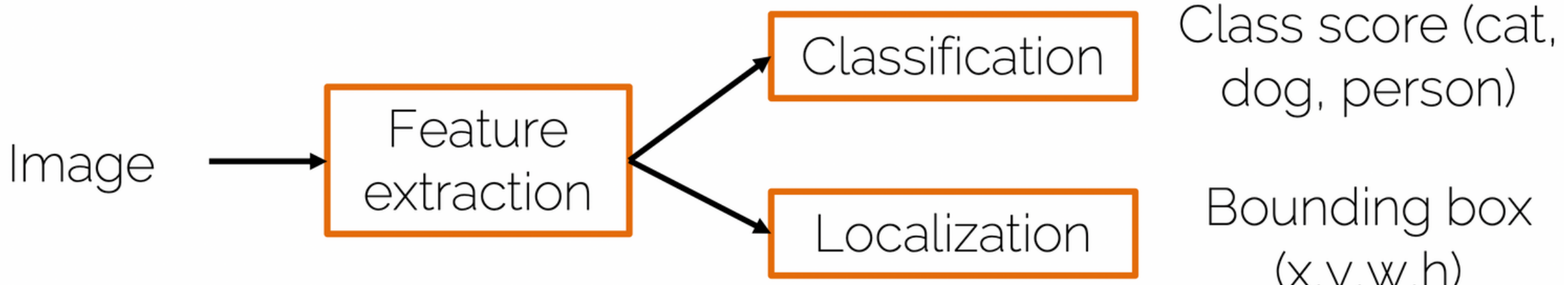
Object detection: Evaluation

- Popular metric: “mean average precision” (mAP)
- Precision: # Correct detections / # Detections
- Combine all detections from all test images to draw a P-R curve for each class;
  AP is the area under the curve
- Compute the average precision (AP) separately for each class.
  Its mean over classes gives mAP
- A detection is a true positive if it has IoU with a ground-truth box greater than some threshold (usually 0.5) (mAP@0.5)
- mAP is a number from 0 to 100; high is good

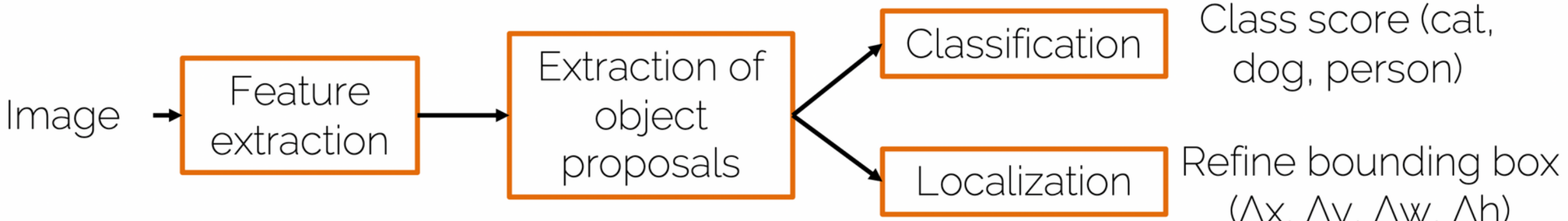
Learning-based Detectors

Types of Object Detectors

- One-stage detectors



- Two-stage detectors



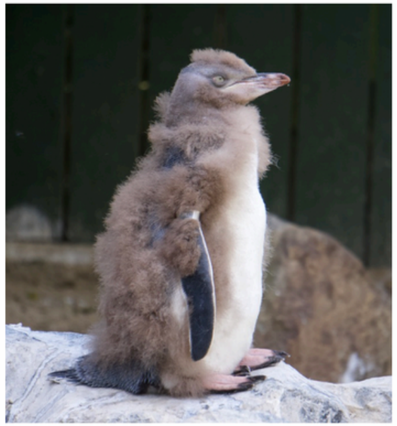
Slides adapted from Prof. Dr. Laura Leal-Taixé and Dr. Aljosa Osep, Technical University of Munich

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

Localization

- Bounding box regression



Image

Feature extraction
(this time with a
Neural Network)

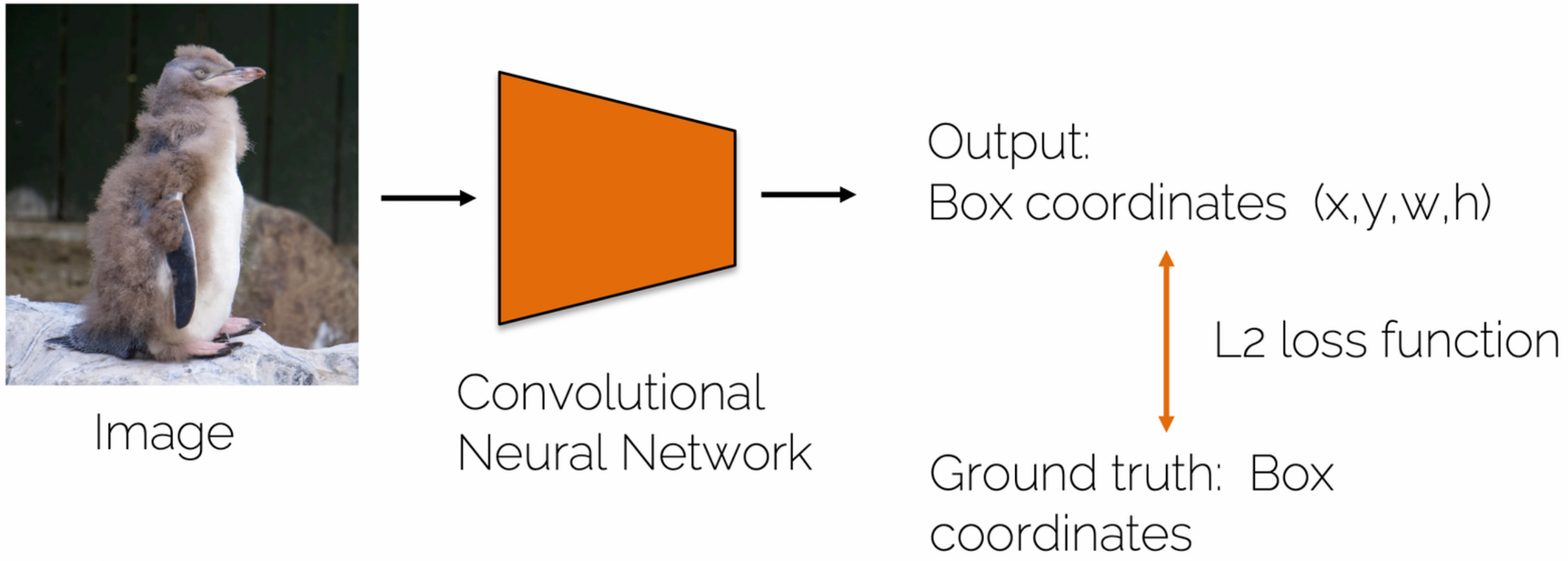
Output:
Box coordinates (x,y,w,h)

L2 loss function

Ground truth: Box
coordinates

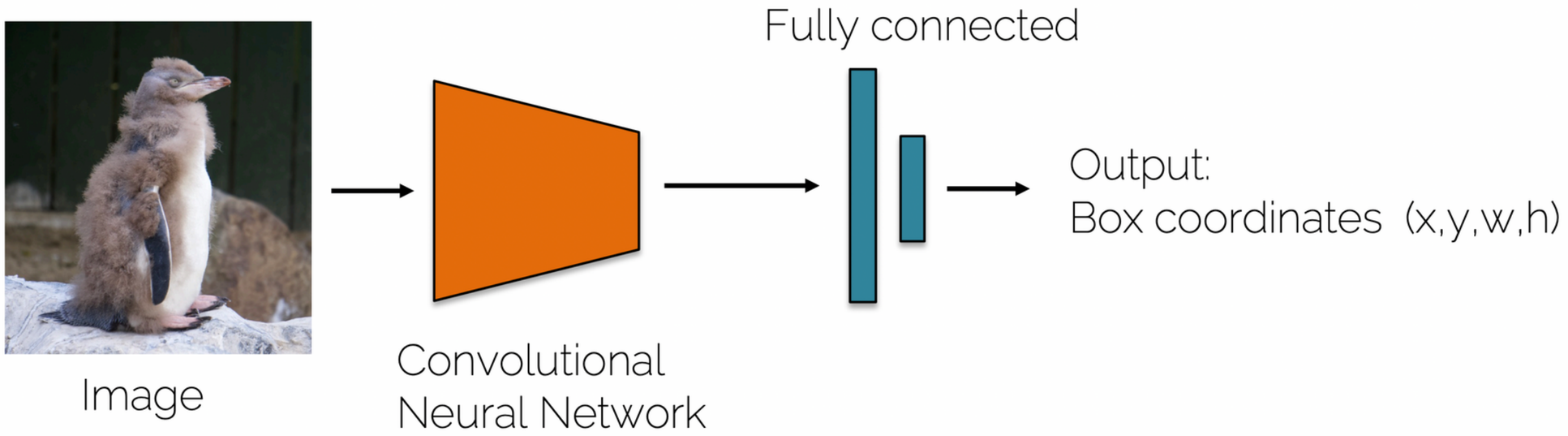
Localization

- Bounding box regression



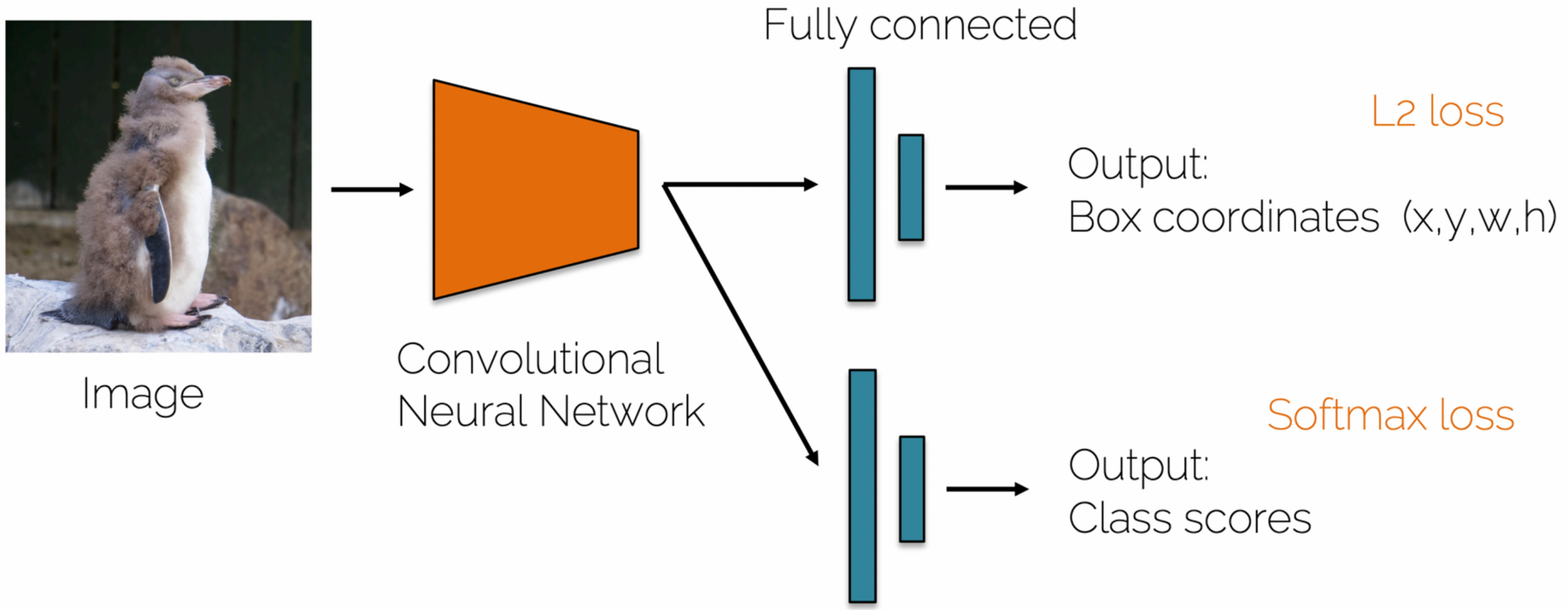
Localization and Classification

- Bounding box regression



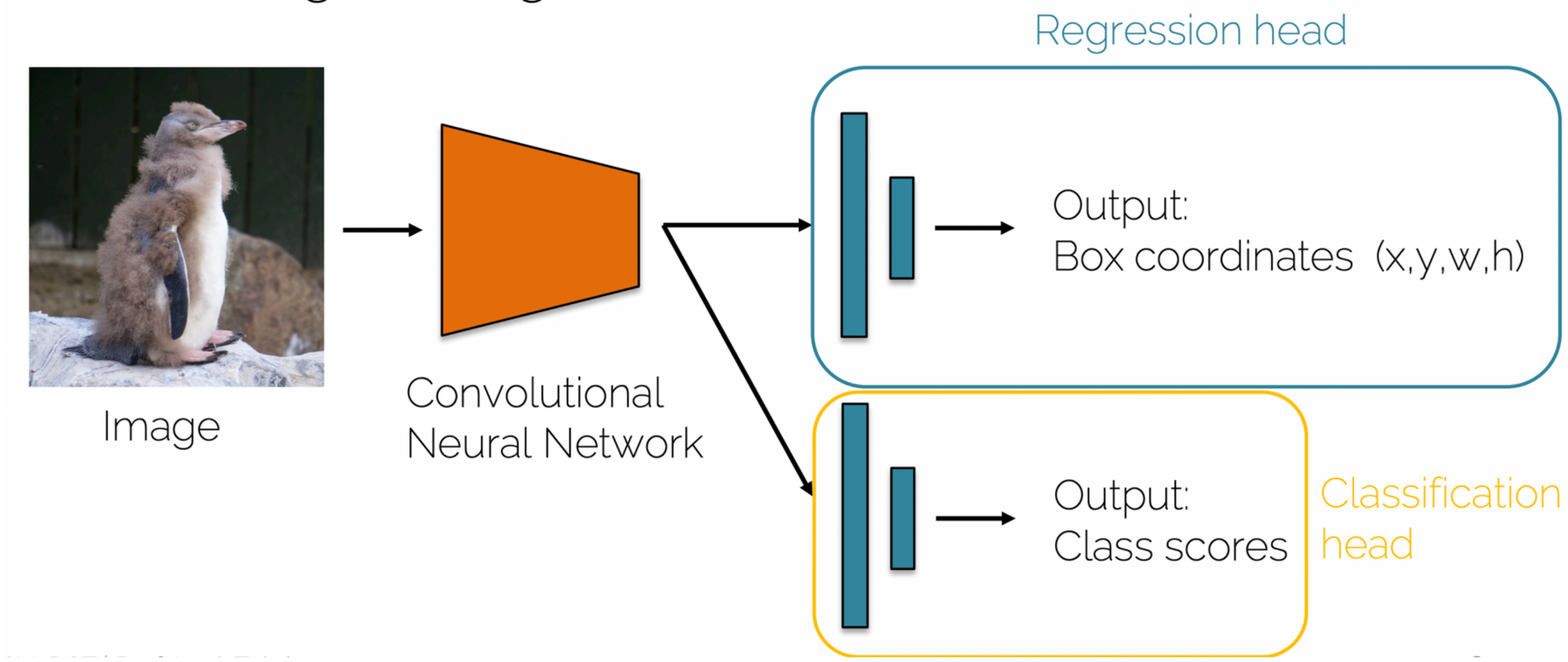
Localization and Classification

- Bounding box regression



Localization and Classification

- Bounding box regression



Localization and Classification

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

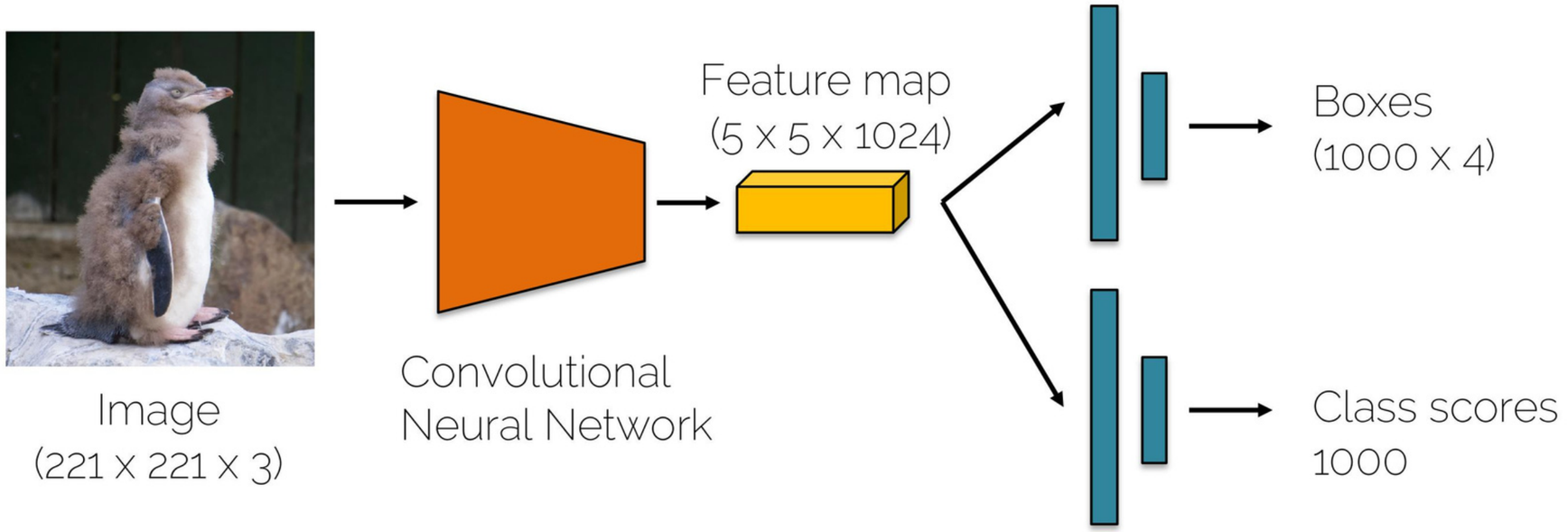
- Then train the regression head

- 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

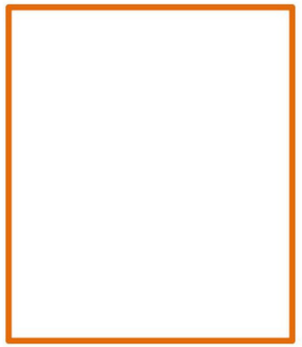


Image (468 x 356 x 3)

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

Overfeat

- Sliding window + box regression + classification

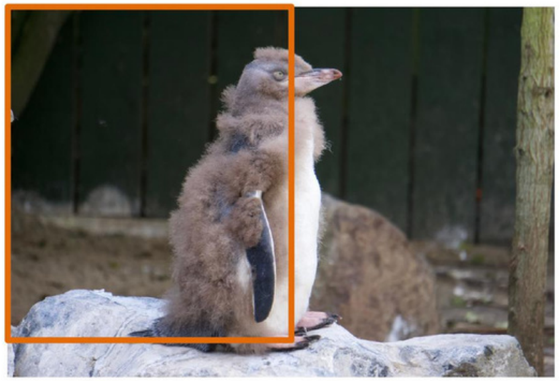


Image (468 x 356 x 3)

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

Overfeat

- Sliding window + box regression + classification

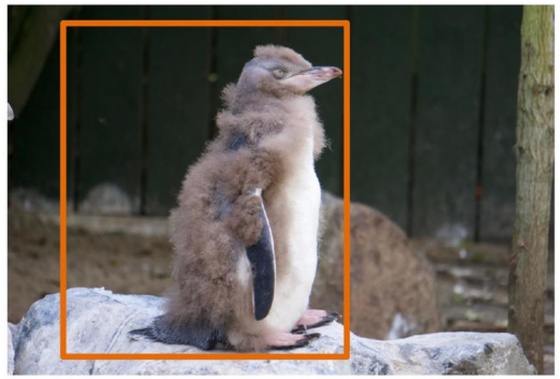


Image (468 x 356 x 3)

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

Overfeat

- Sliding window + box regression + classification

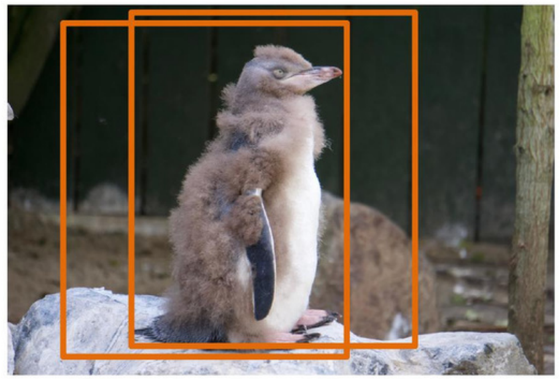


Image (468 x 356 x 3)

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)

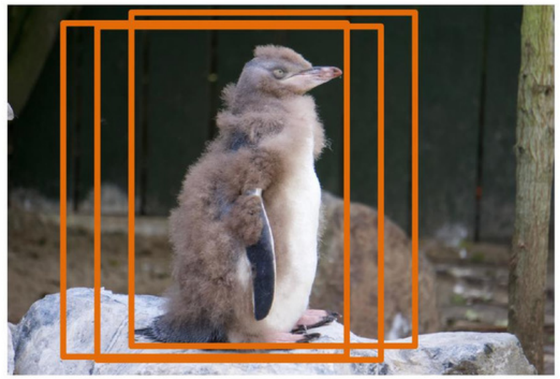


Image (468 x 356 x 3)

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)

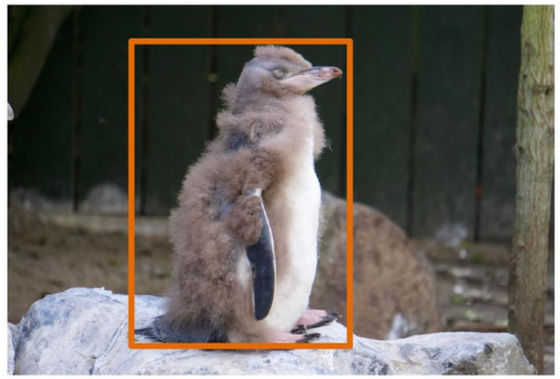


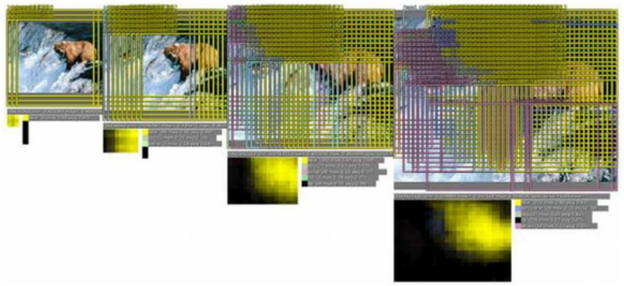
Image (468 x 356 x 3)

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

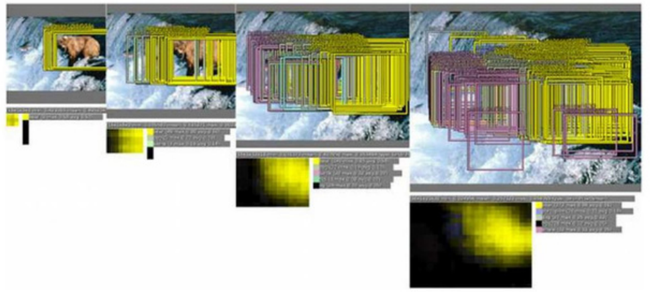
Overfeat

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

Window positions + score maps



Box regression outputs

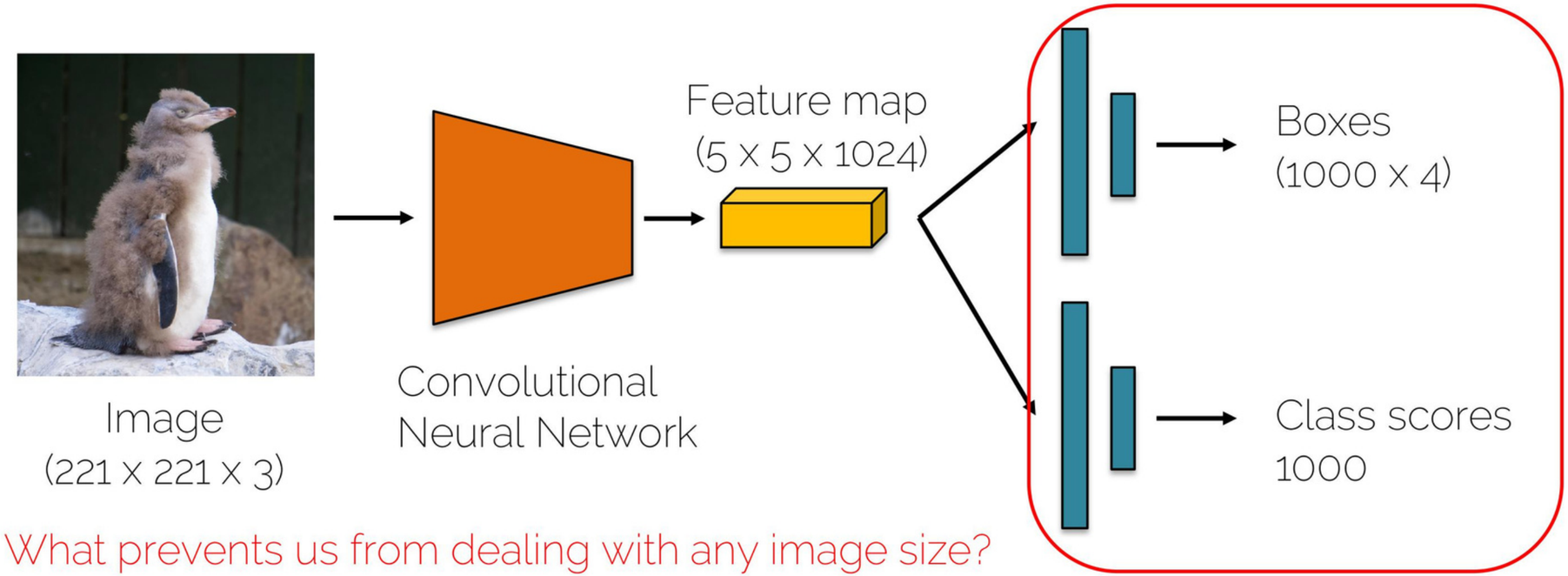


Final Predictions

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

Overfeat

- Sliding window + box regression + classification

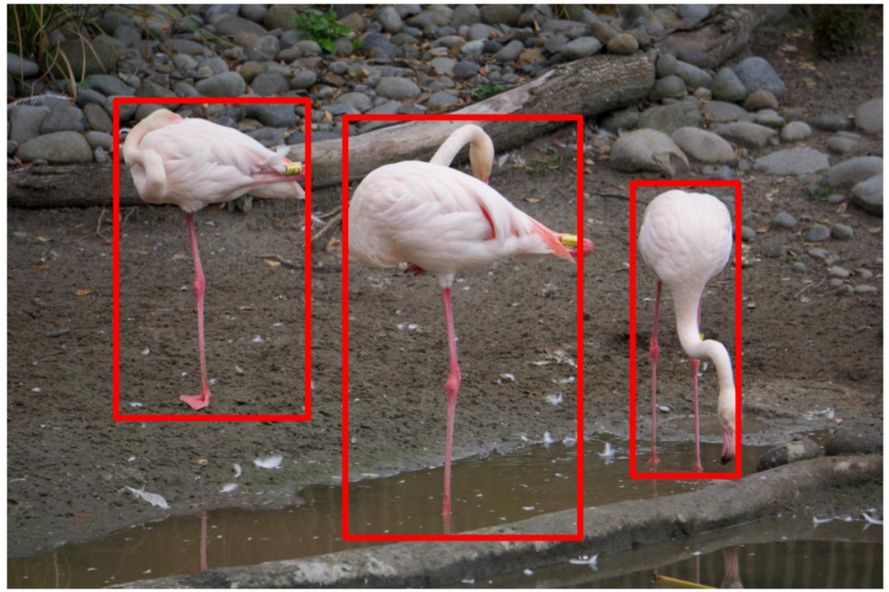


What prevents us from dealing with any image size?

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

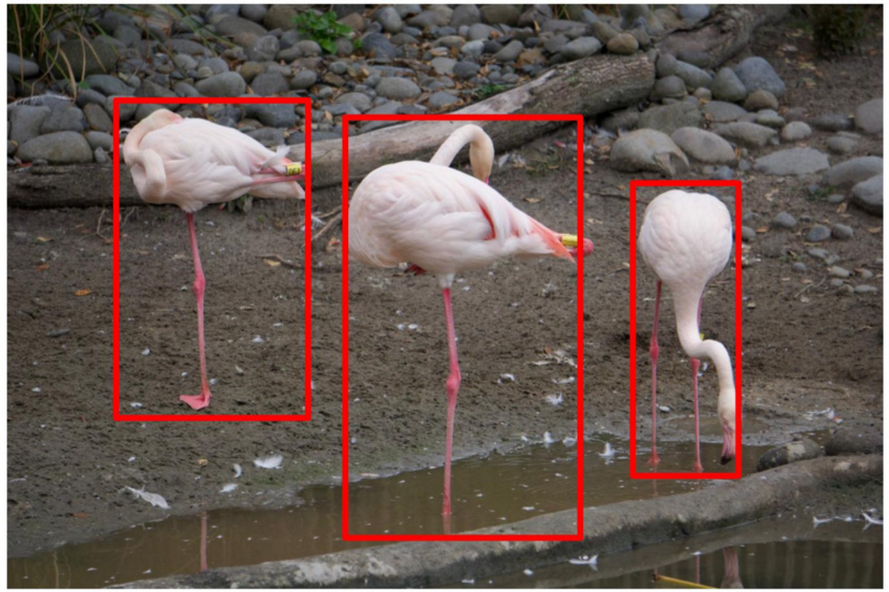
What about multiple objects?

- Localization: ✅ Regression
- How about detection?



What about multiple objects?

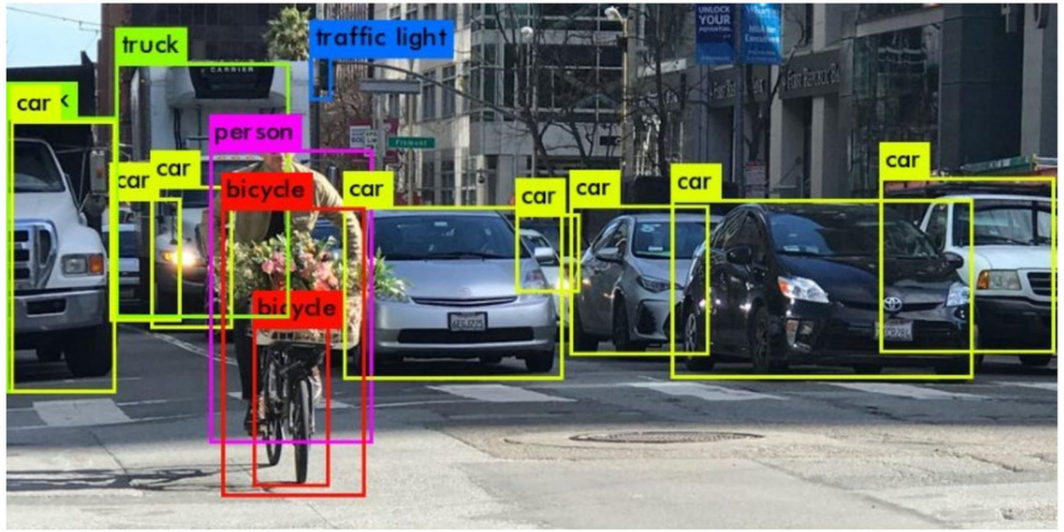
- Localization: ✅ Regression
- How about detection?



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?

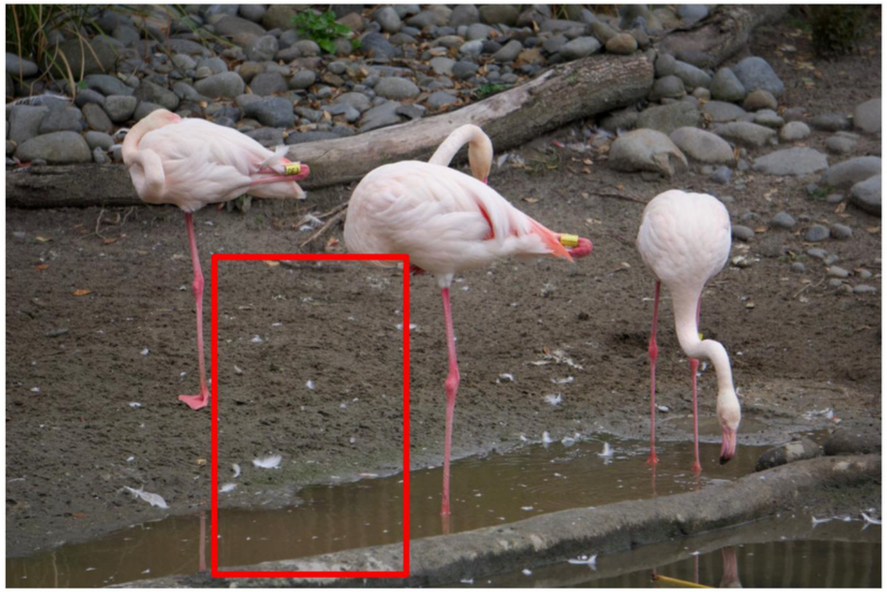
- Localization: ✅ Regression
- How about detection?

- 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

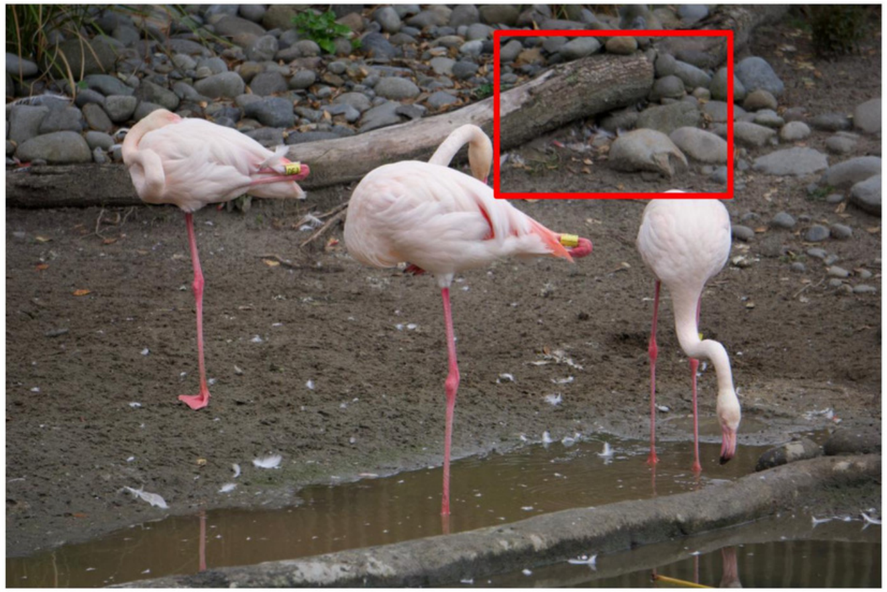


Is this a Flamingo?

NO

Detection as classification?

- Localization: ✅ Regression
- How about detection? ❌ Regression

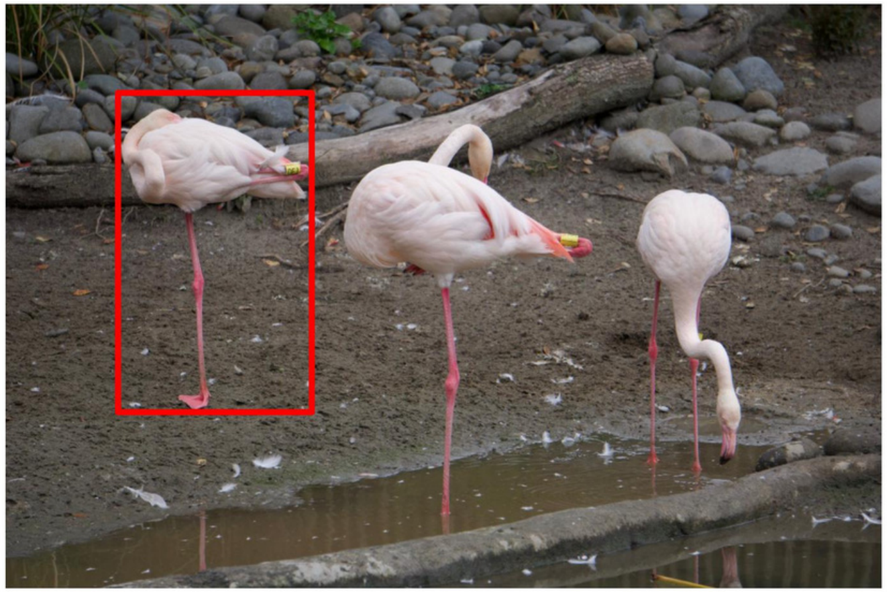


Is this a Flamingo?

NO

Detection as classification?

- Localization: ✅ Regression
- How about detection? ❌ Regression



Is this a Flamingo?

YES!

Detection as classification?

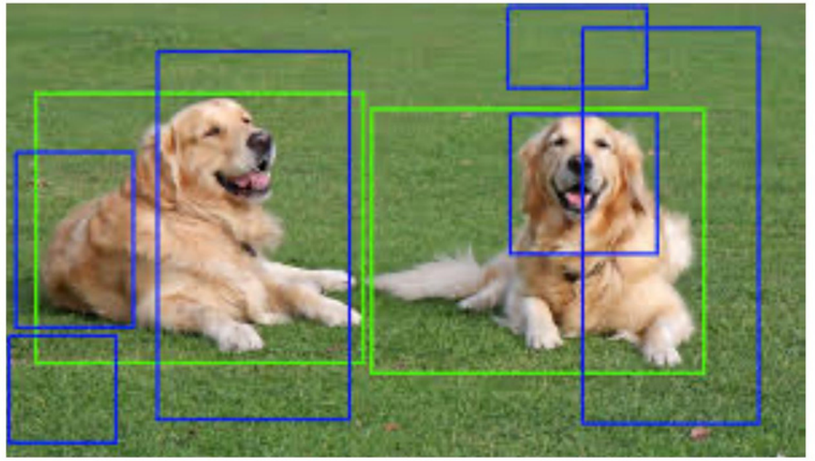
- Localization: ✅ Regression
- How about detection? ✅ Classification

- 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

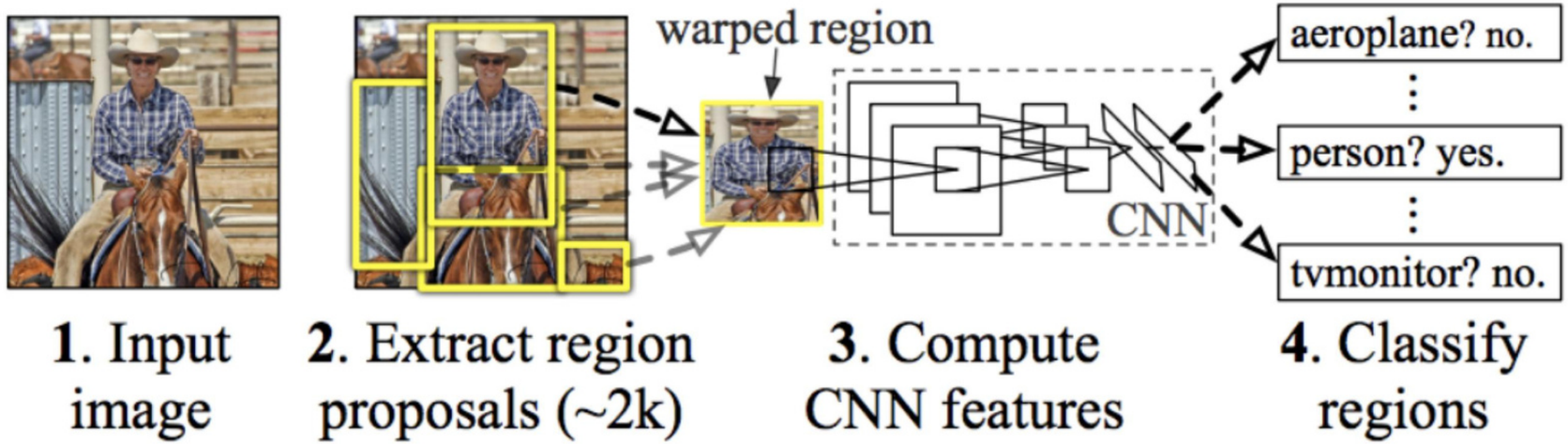
- We have already seen a method that gives us “interesting” regions in an image that potentially contain an object

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



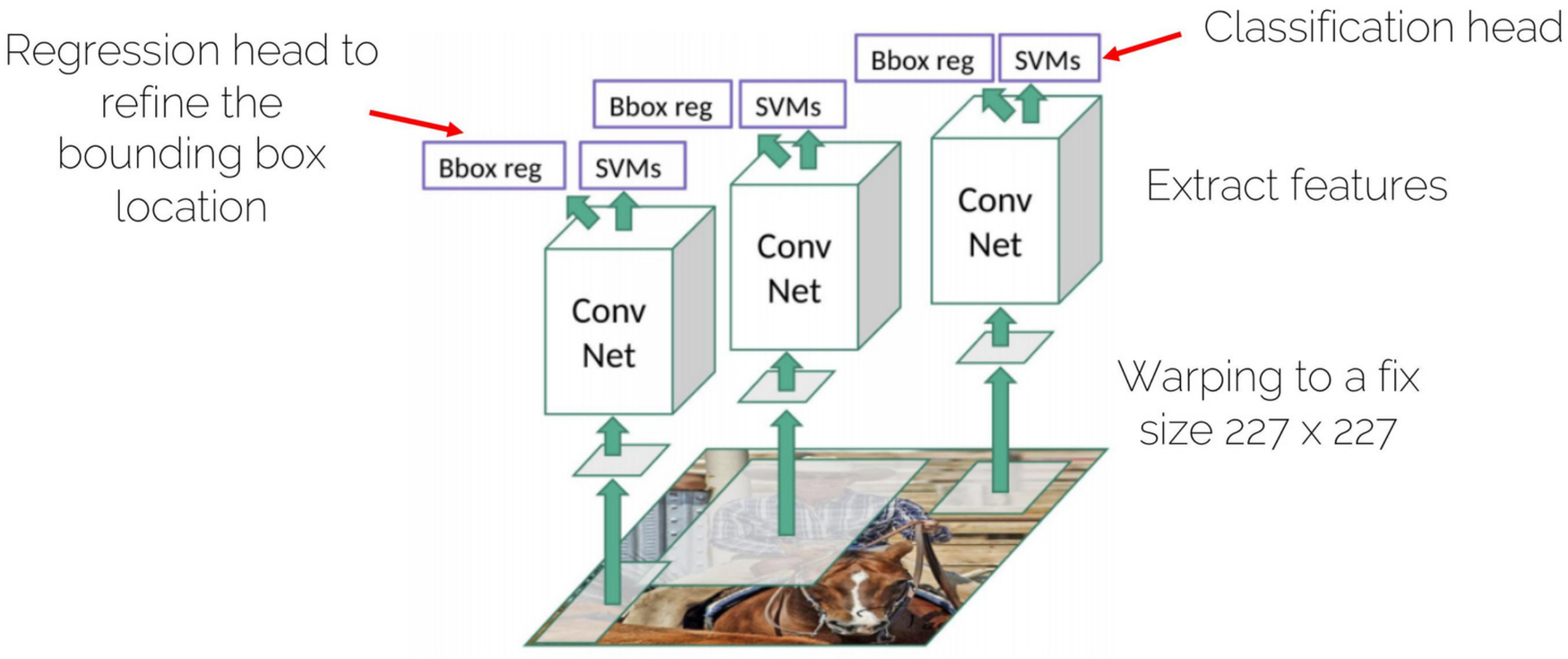
The R-CNN Family

R-CNN



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

R-CNN



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)

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

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.

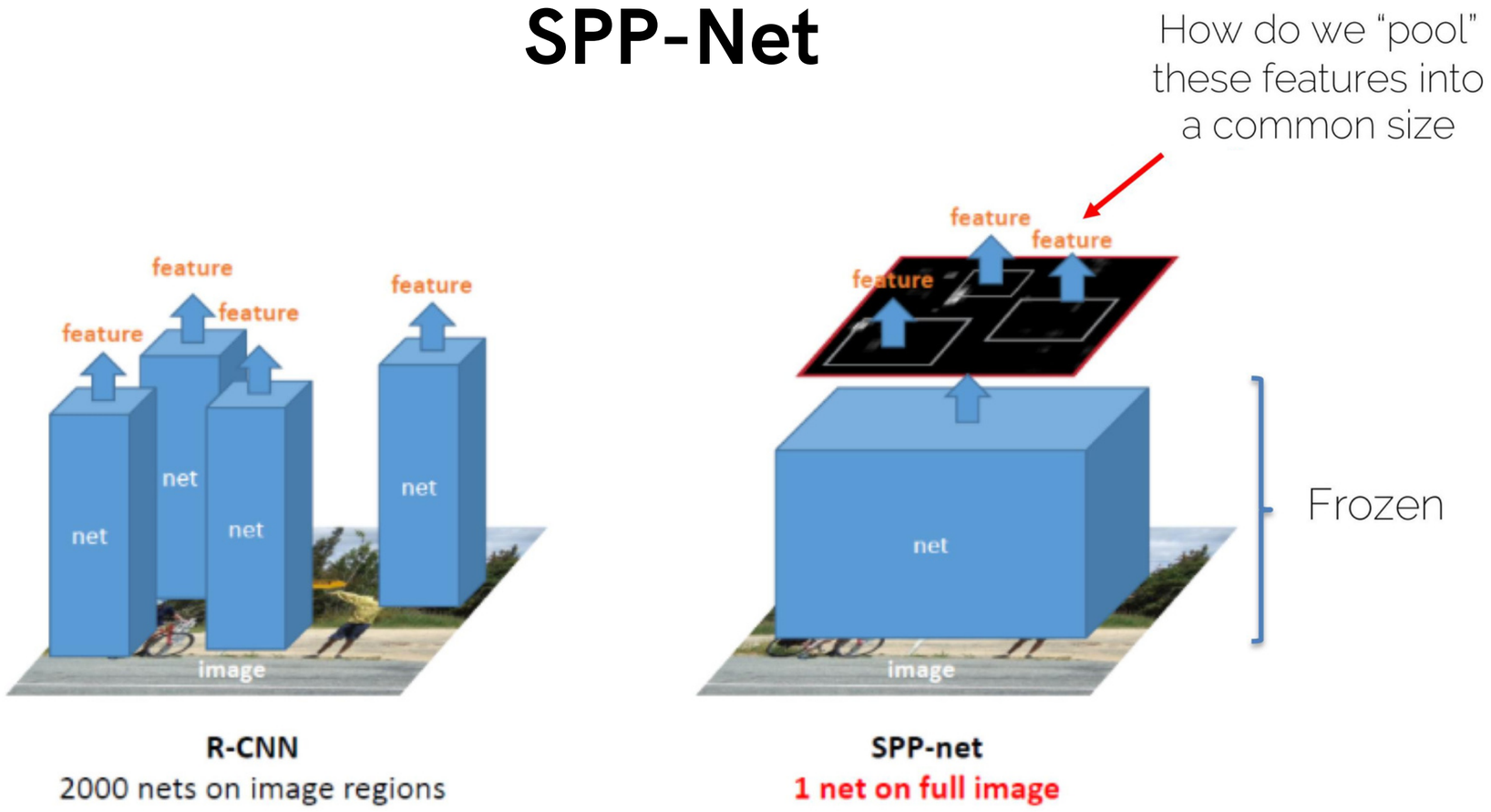
Girschick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014

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.

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

SPP-Net



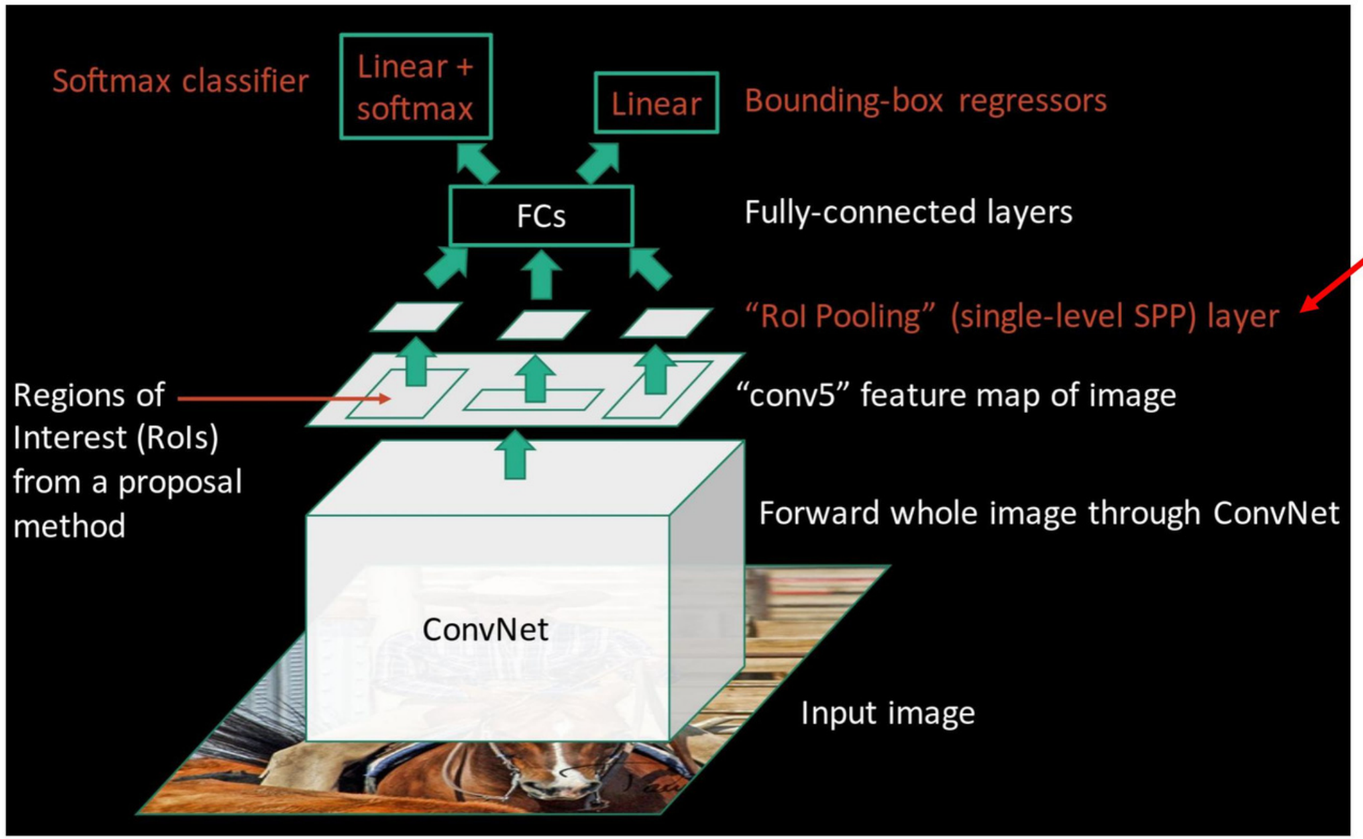
He et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. ECCV 2014.

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.

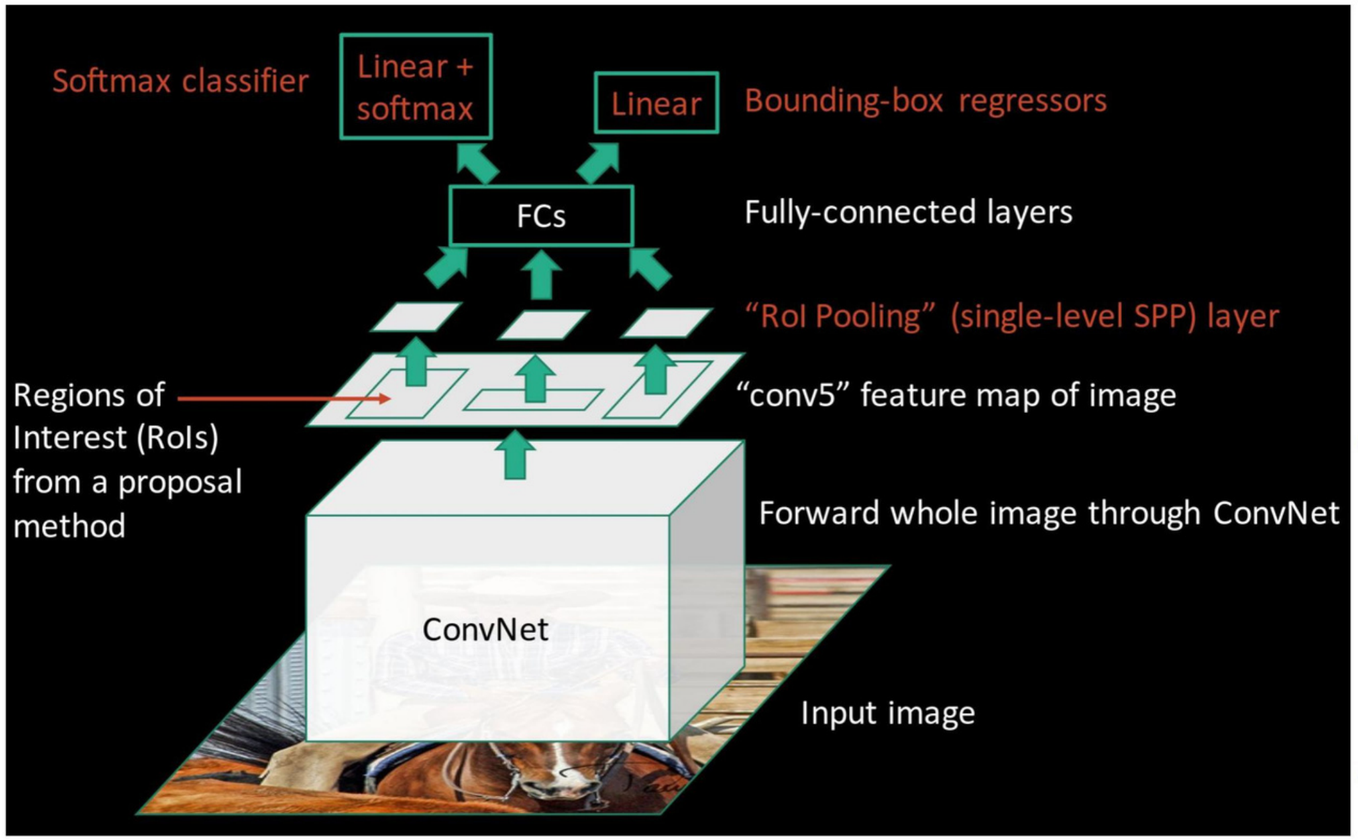
Fast R-CNN



Region of Interest Pooling

Girschick, "Fast R-CNN", ICCV 2015

Fast R-CNN

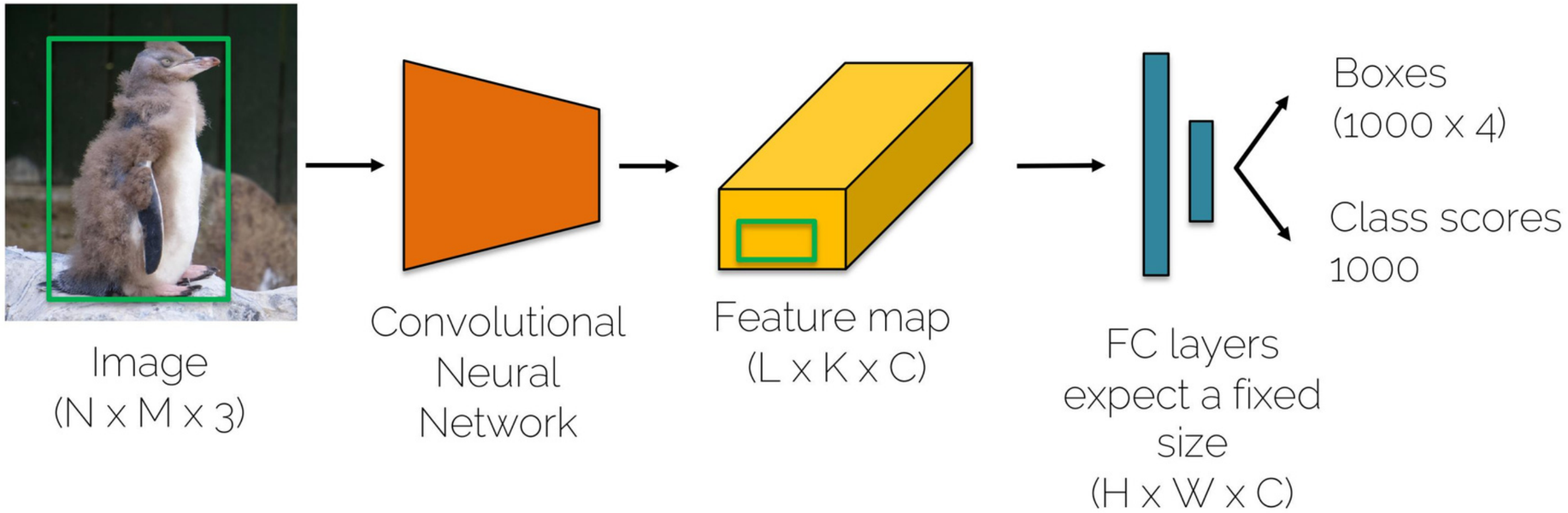


Shared computation at test time (like SPP)

Girschick, "Fast R-CNN", ICCV 2015

Fast R-CNN: RoI Pooling

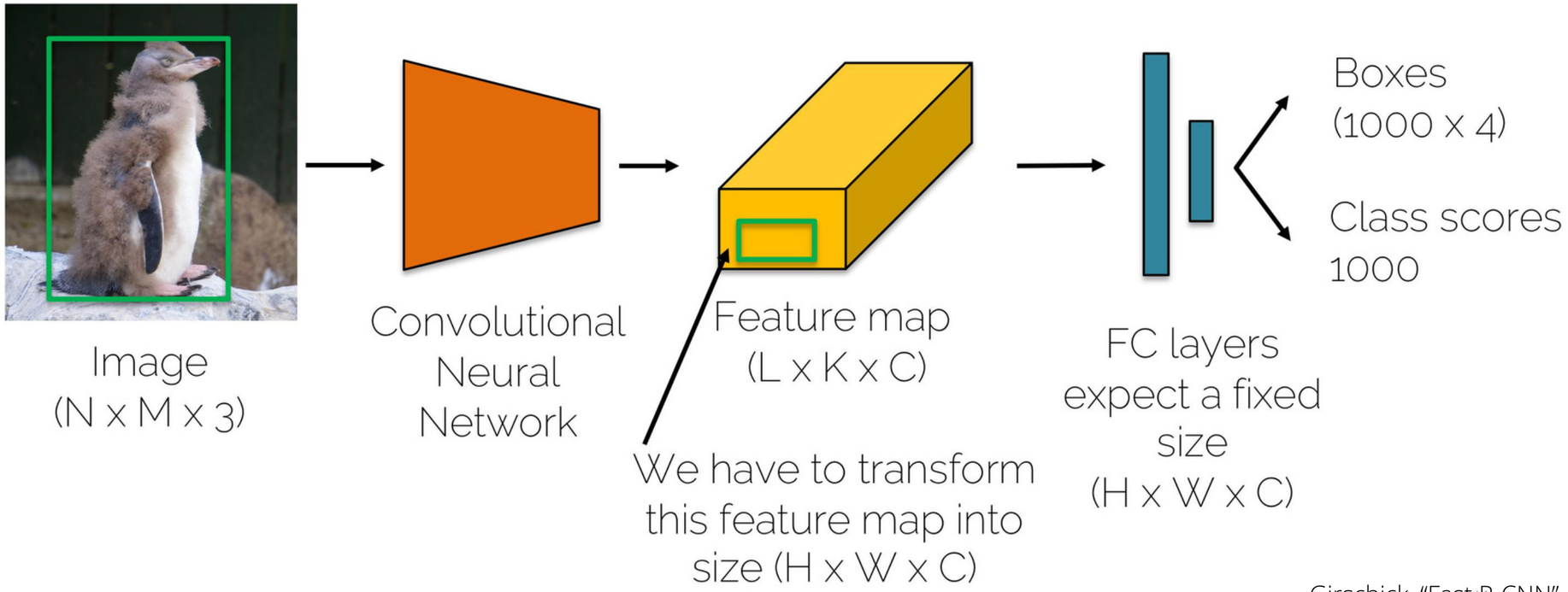
- Region of Interest Pooling



Girschick, "Fast R-CNN", ICCV 2015

Fast R-CNN: RoI Pooling

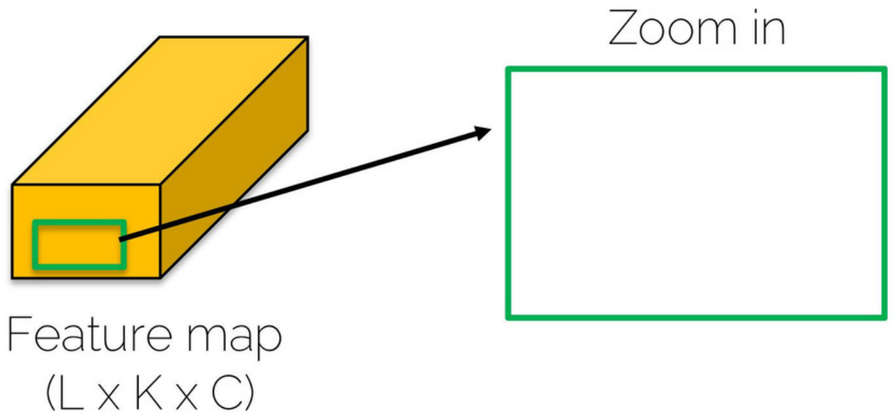
- Region of Interest Pooling

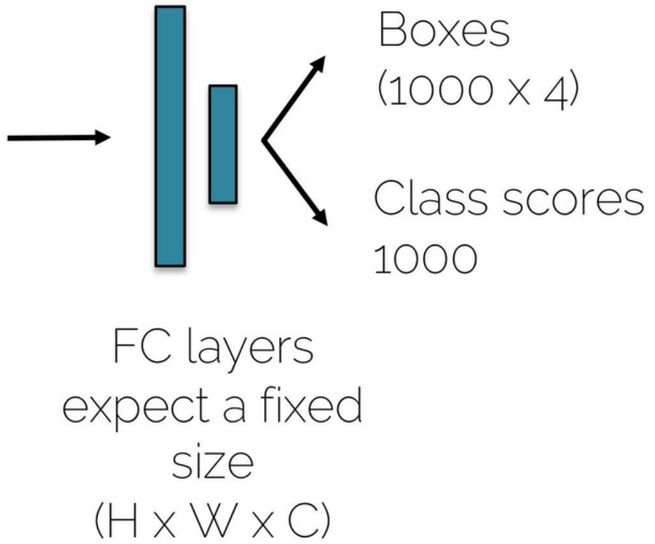


Girschick, "Fast R-CNN", ICCV 2015

Fast R-CNN: RoI Pooling

- Region of Interest Pooling

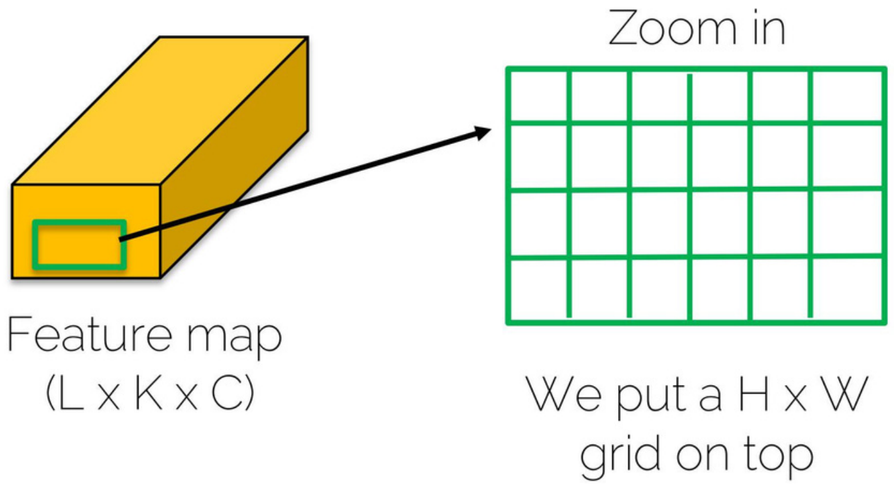


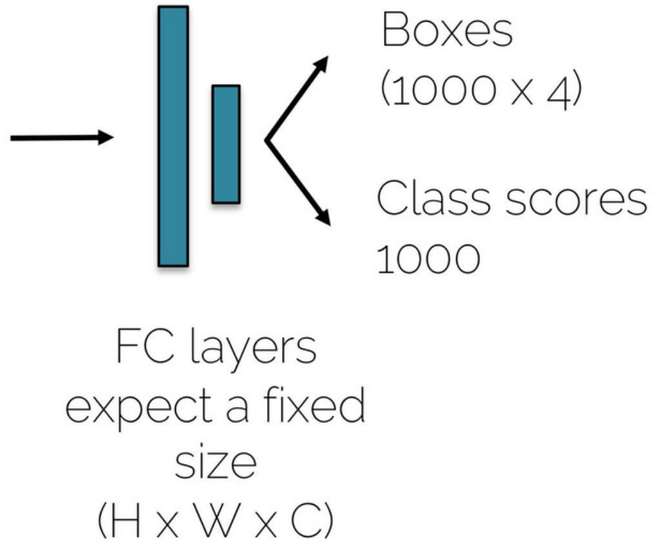


Girschick, "Fast R-CNN", ICCV 2015

Fast R-CNN: RoI Pooling

- Region of Interest Pooling

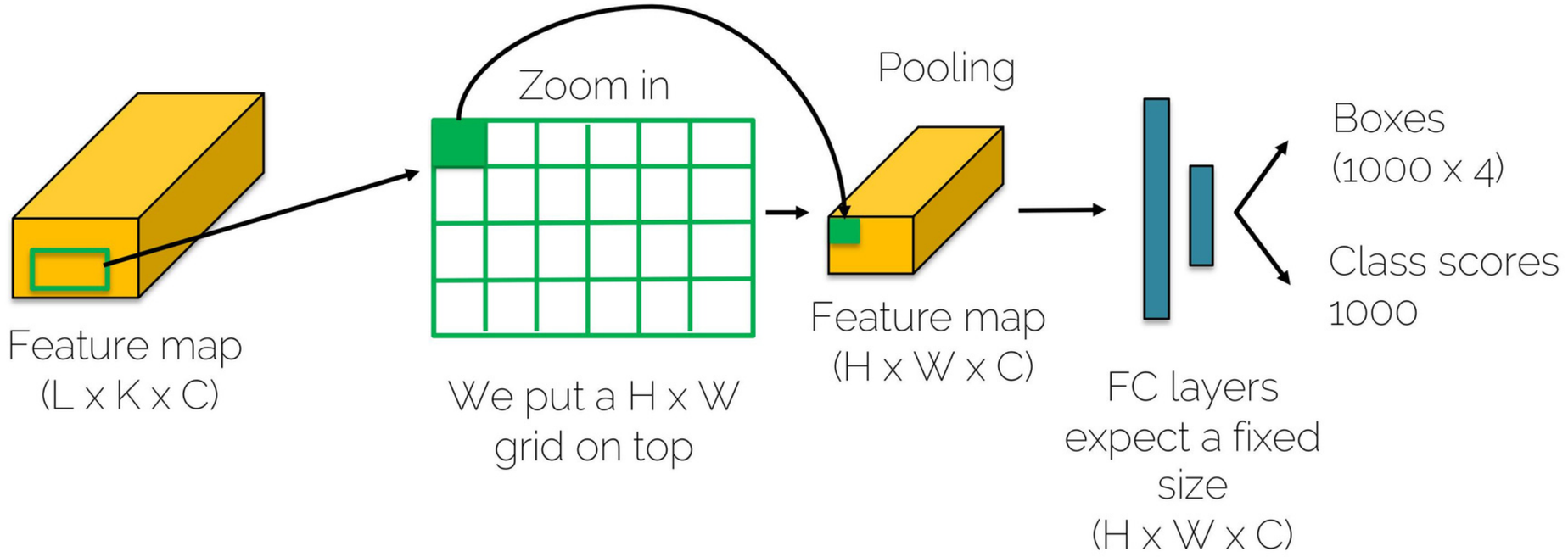




Girschick, "Fast R-CNN", ICCV 2015

Fast R-CNN: RoI Pooling

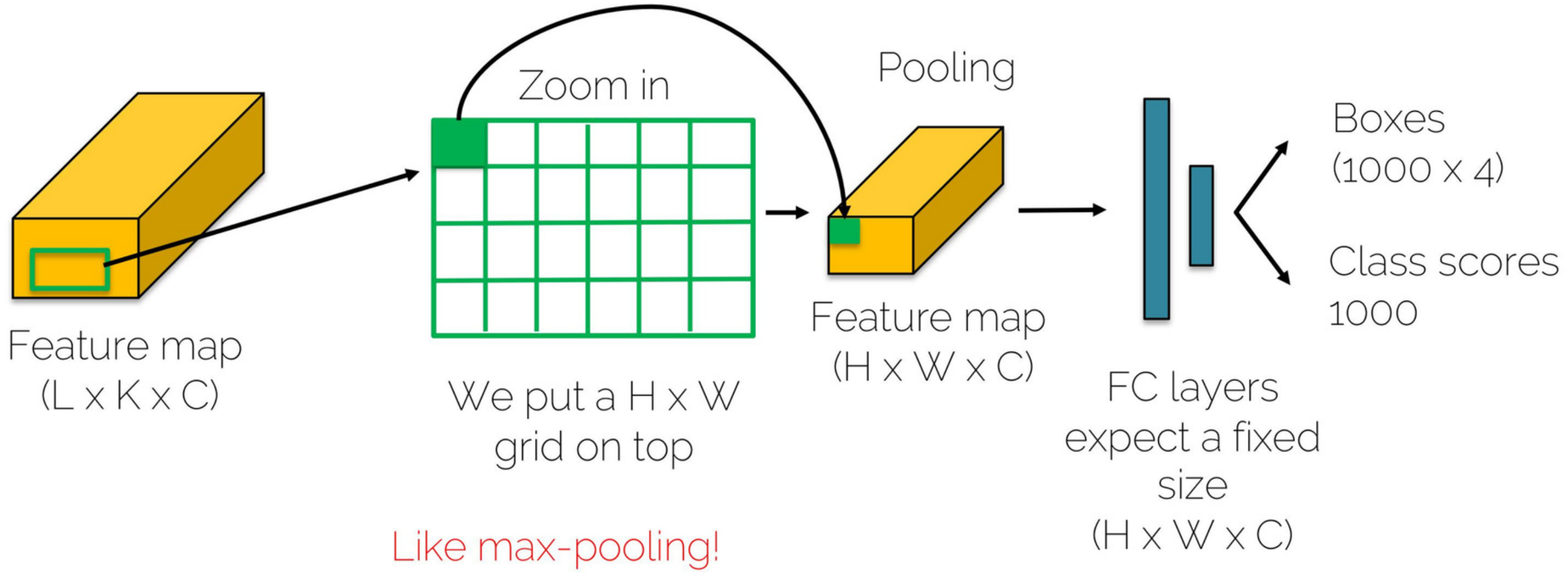
- Region of Interest Pooling



Girschick, "Fast R-CNN", ICCV 2015

Fast R-CNN: RoI Pooling

- RoI Pooling: how do you do backpropagation?



Girschick, "Fast R-CNN", ICCV 2015

Fast R-CNN: Results

- VGG-16 CNN on Pascal VOC 2007 dataset

Faster!

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

Girschick, “Fast R-CNN”, ICCV 2015

Fast R-CNN: Results

- VGG-16 CNN on Pascal VOC 2007 dataset

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

Faster!

FASTER!

Girschick, “Fast R-CNN”, ICCV 2015

Fast R-CNN: Results

- VGG-16 CNN on Pascal VOC 2007 dataset

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

Faster!

FASTER!

Better!

Girschick, “Fast R-CNN”, ICCV 2015

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
Training Time:	84 hours	9.5 hours
(Speedup)	1x	8.8x
Test time per image	47 seconds	0.32 seconds
(Speedup)	1x	146x
mAP (VOC 2007)	66.0	66.9

Faster!

FASTER!

Better!

Girschick, "Fast R-CNN", ICCV 2015

Fast R-CNN: Results

With proposals included

- VGG-16 CNN on Pascal VOC 2007 dataset

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

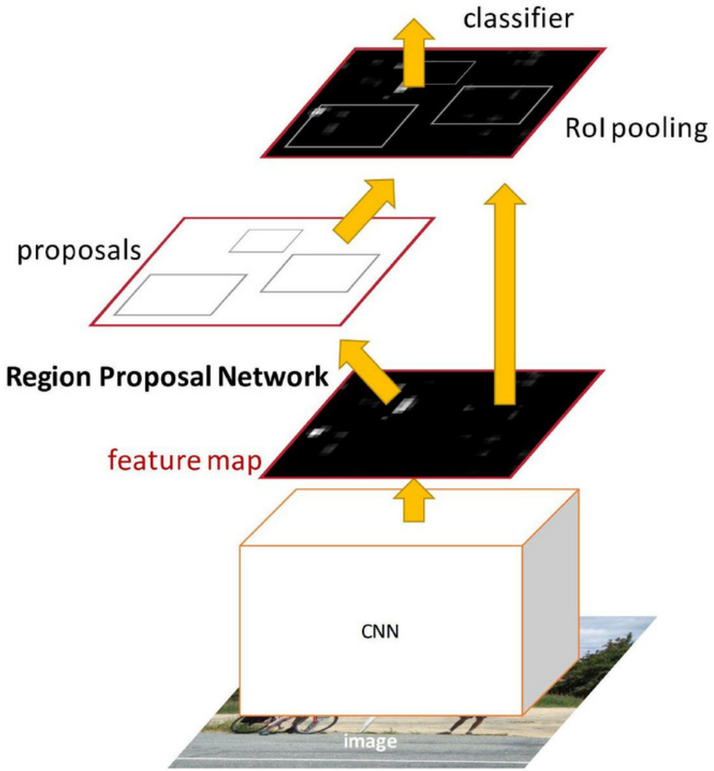
Faster!

FASTER!

Better!

Girschick, "Fast R-CNN", ICCV 2015

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

Region Proposal Network

- How to extract proposals

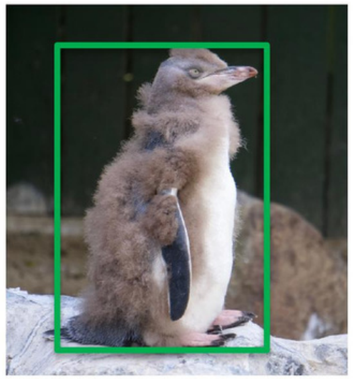
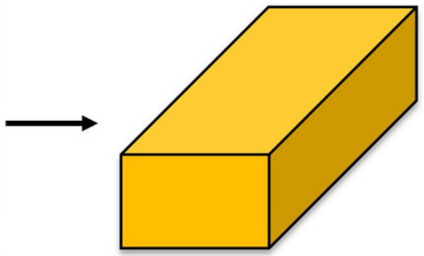


Image
(N x M x 3)



(H x W x 4096)

Zoom in

Extract proposals

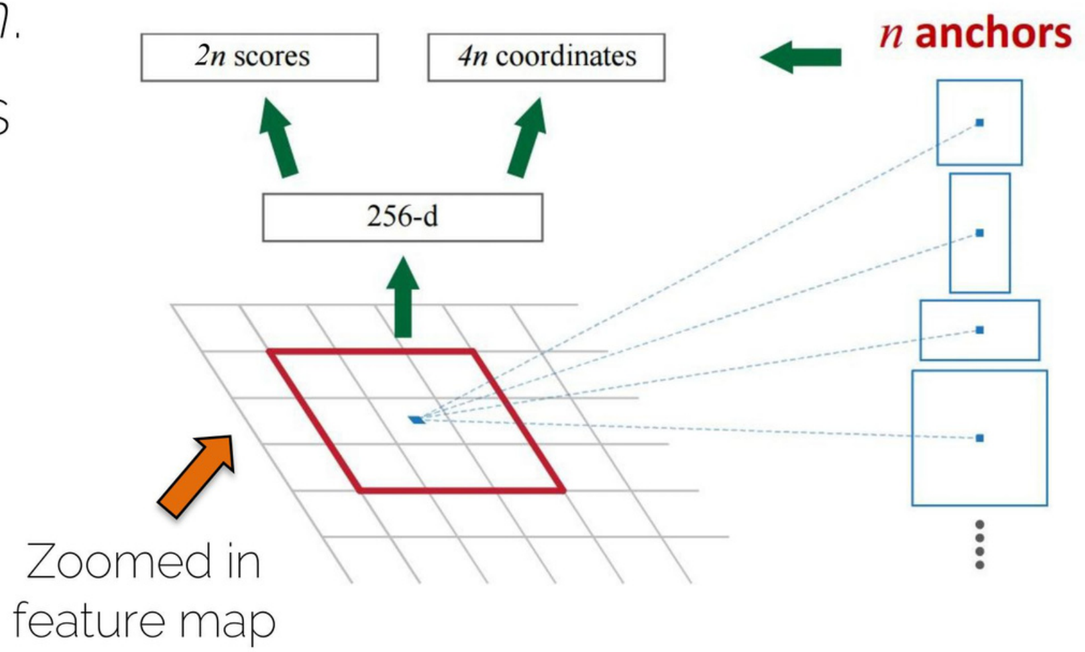
- How many proposals?
- ✓ We need to decide a fixed number

- Where are they placed?
- ✓ Densely

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

Region Proposal Network

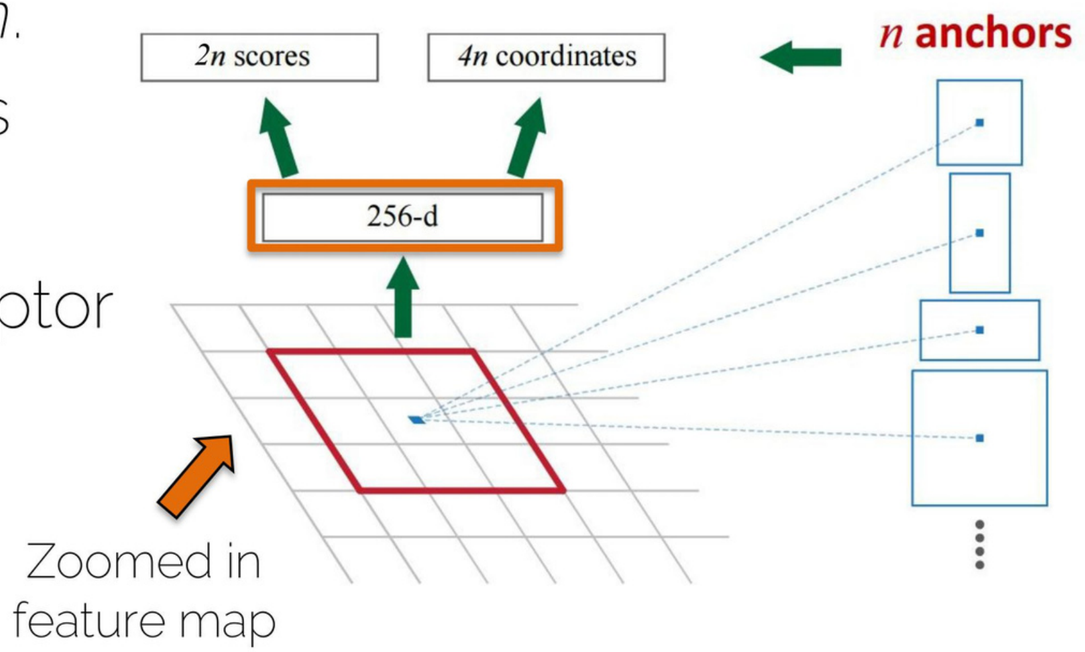
- We fix the number of proposals by using a set of n=9 anchors *per location*.
- 9 anchors = 3 scales and 3 aspect ratios



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

Region Proposal Network

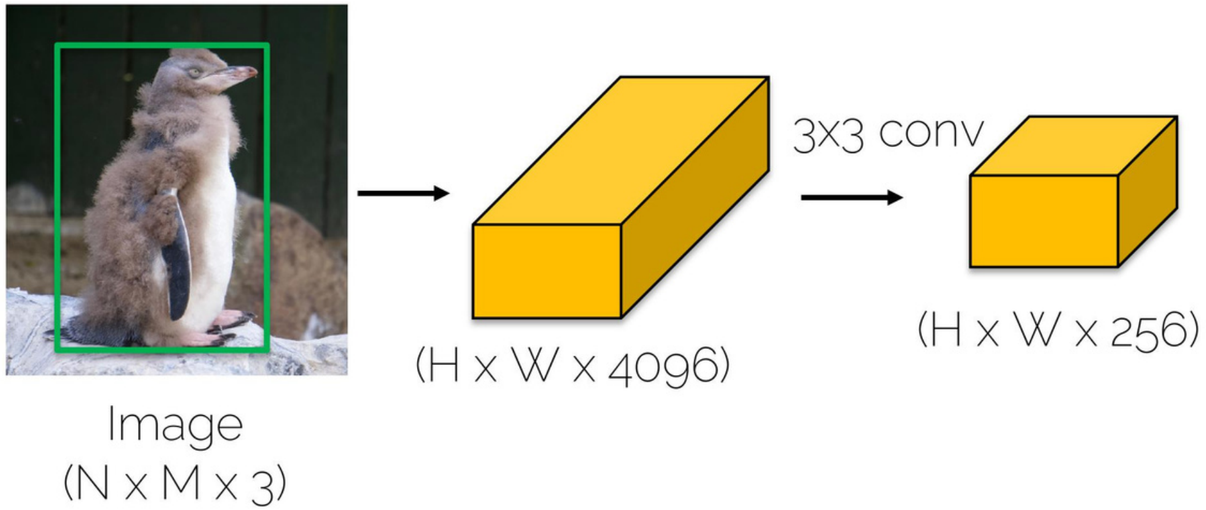
- We fix the number of proposals by using a set of n=9 anchors *per location*.
- 9 anchors = 3 scales and 3 aspect ratios
- We extract a descriptor per *location*



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

Region Proposal Network

- How to extract proposals

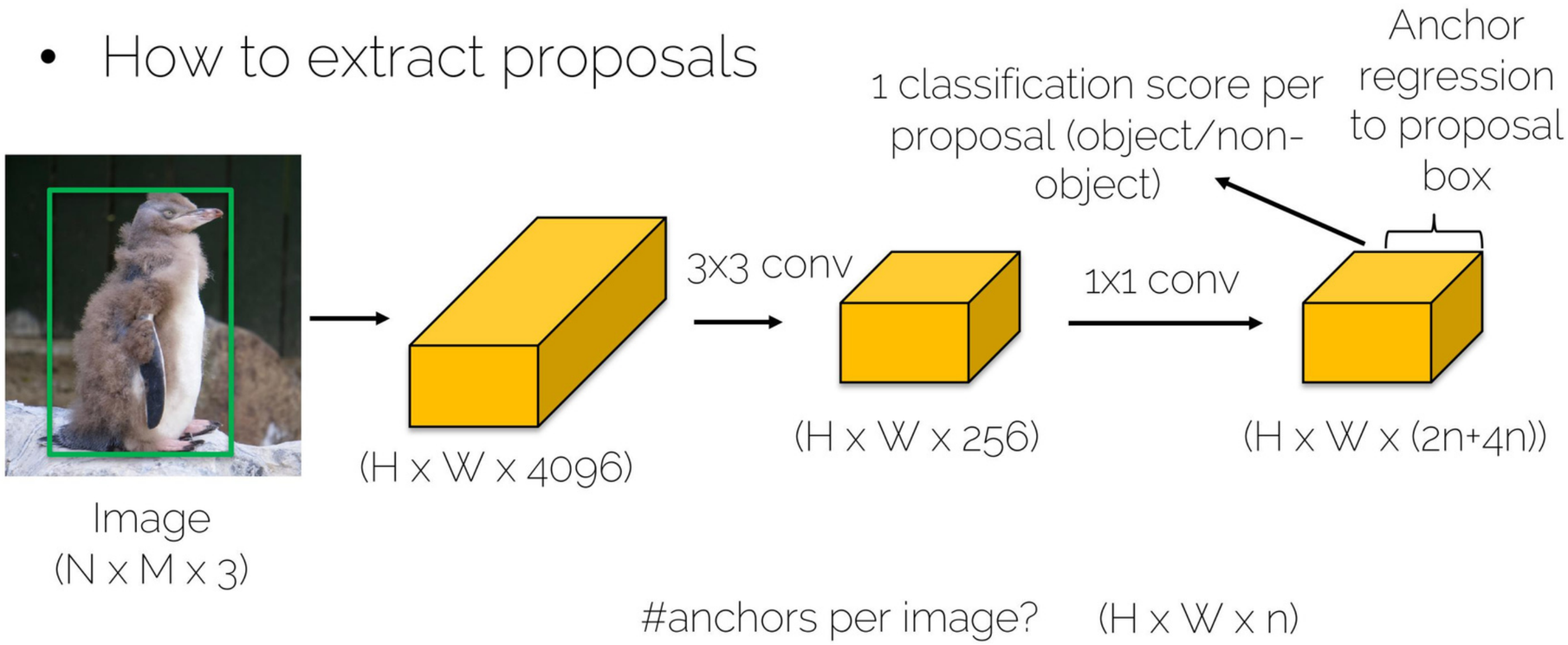


#anchors per image? (H x W x n)

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

Region Proposal Network

- How to extract proposals

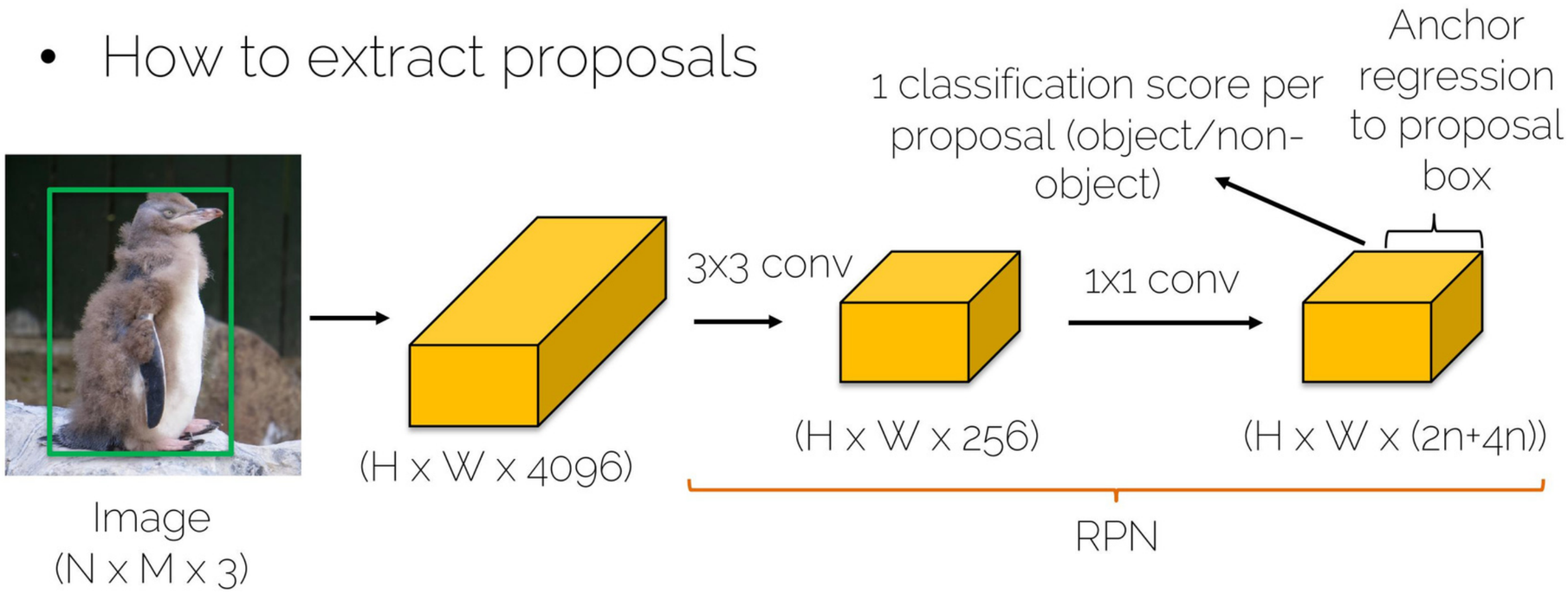


#anchors per image? (H x W x n)

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

Region Proposal Network

- How to extract proposals



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

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

RPN Training and Losses

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

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

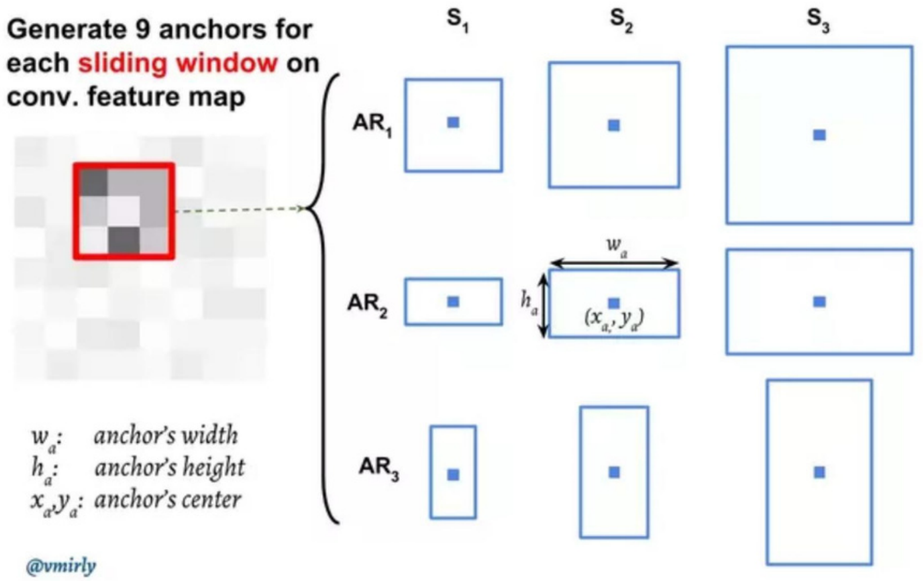
$$p^* = 0 \quad if \quad \text{IoU} < 0.3$$

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

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

RPN Training and Losses

- Each anchor is described by the center position, width and height x_a, y_a, w_a, h_a



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

RPN Training and Losses

- Each anchor is described by the center position, width and height x_a, y_a, w_a, h_a
- What the network actually predicts are t_x, t_y, t_w, t_h

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

Normalized width $t_w = log(w/w_a),$ $t_h = log(h/h_a),$ Normalized height

- Smooth L1 loss on regression targets

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

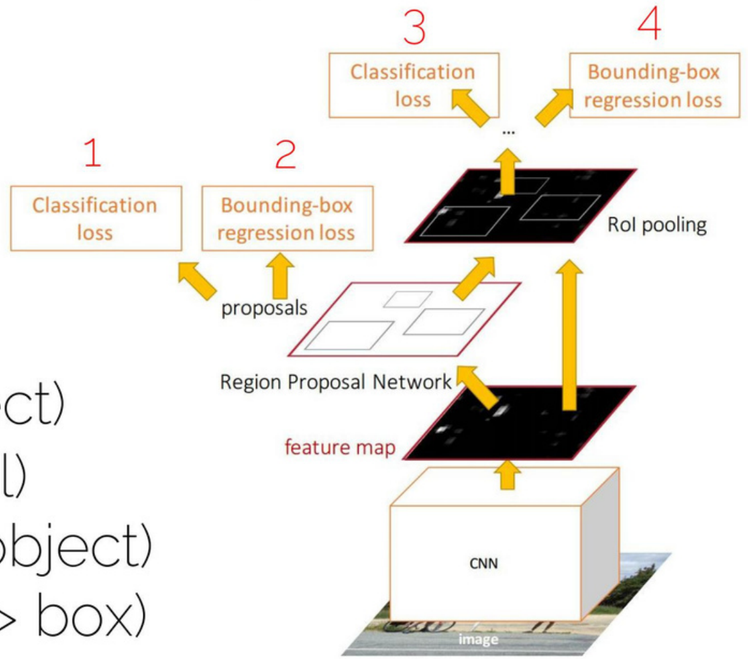
RPN Training and Losses

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

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

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)

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

Faster R-CNN

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

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

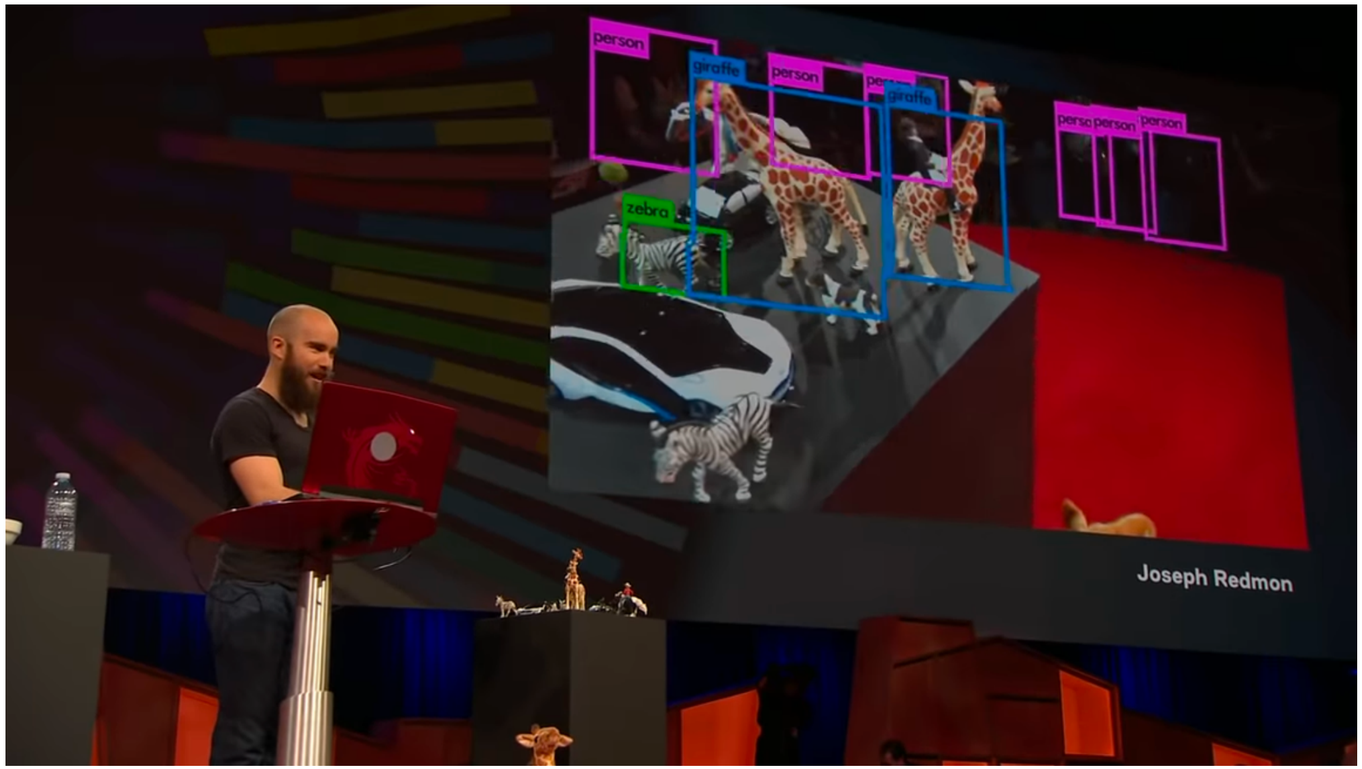
Faster R-CNN: Results

	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image (with proposals)	50 seconds	2 seconds	0.2 seconds
(Speedup)	1x	25x	250x
mAP (VOC 2007)	66.0	66.9	66.9

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.

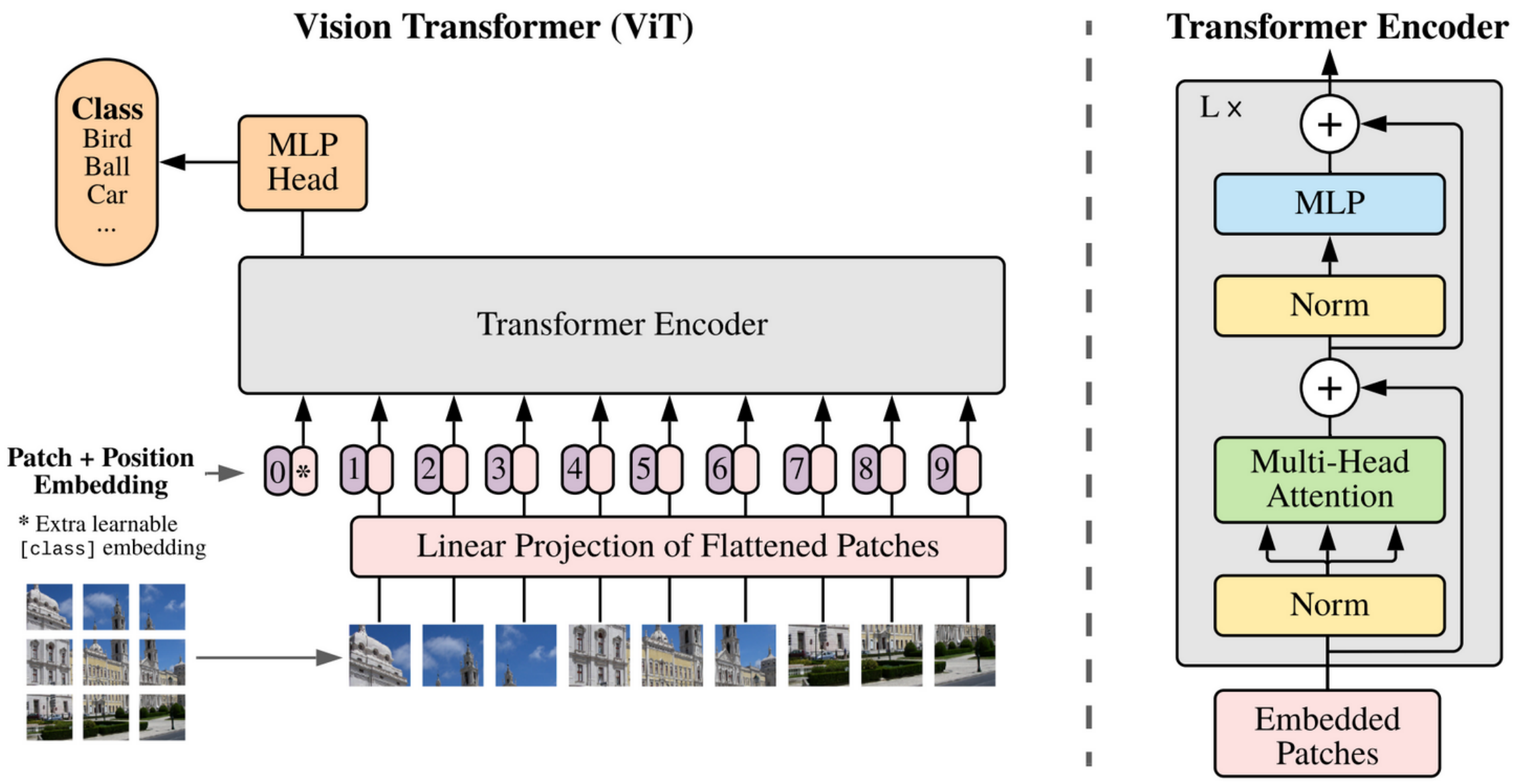
You Only Look Once (YOLO)



Joseph Redmon's How computers learn to recognize objects instantly, TED 2017

Vision Transformer

Vision Transformer



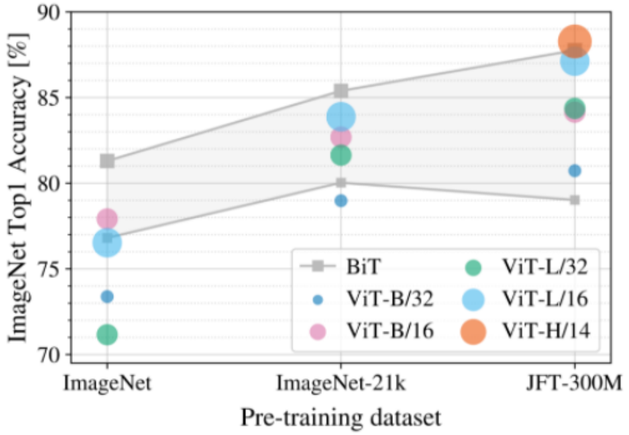
Dosovitskiy et al, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, ICLR 2021

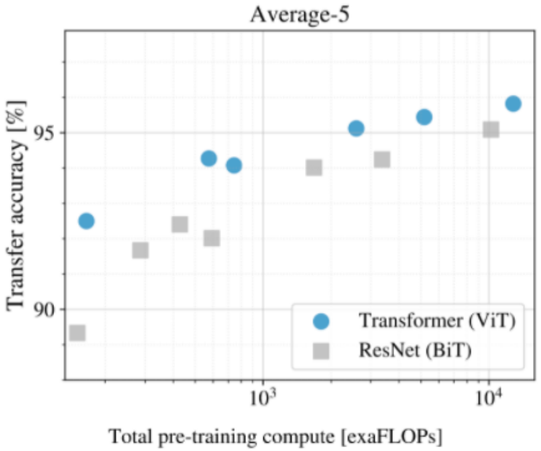
Vision Transformer

- Split an image into fixed-size patches, linearly embed each of them, add position embeddings
- Feed the resulting sequence of vectors to a standard Transformer encoder
- To perform classification, add an extra learnable “classification token” to the sequence

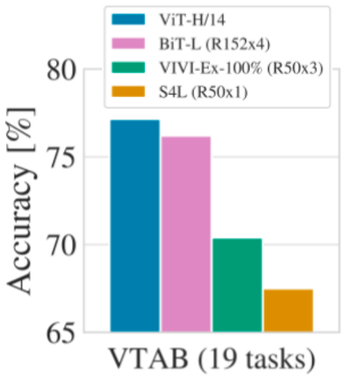
Dosovitskiy et al, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, ICLR 2021

Vision Transformer



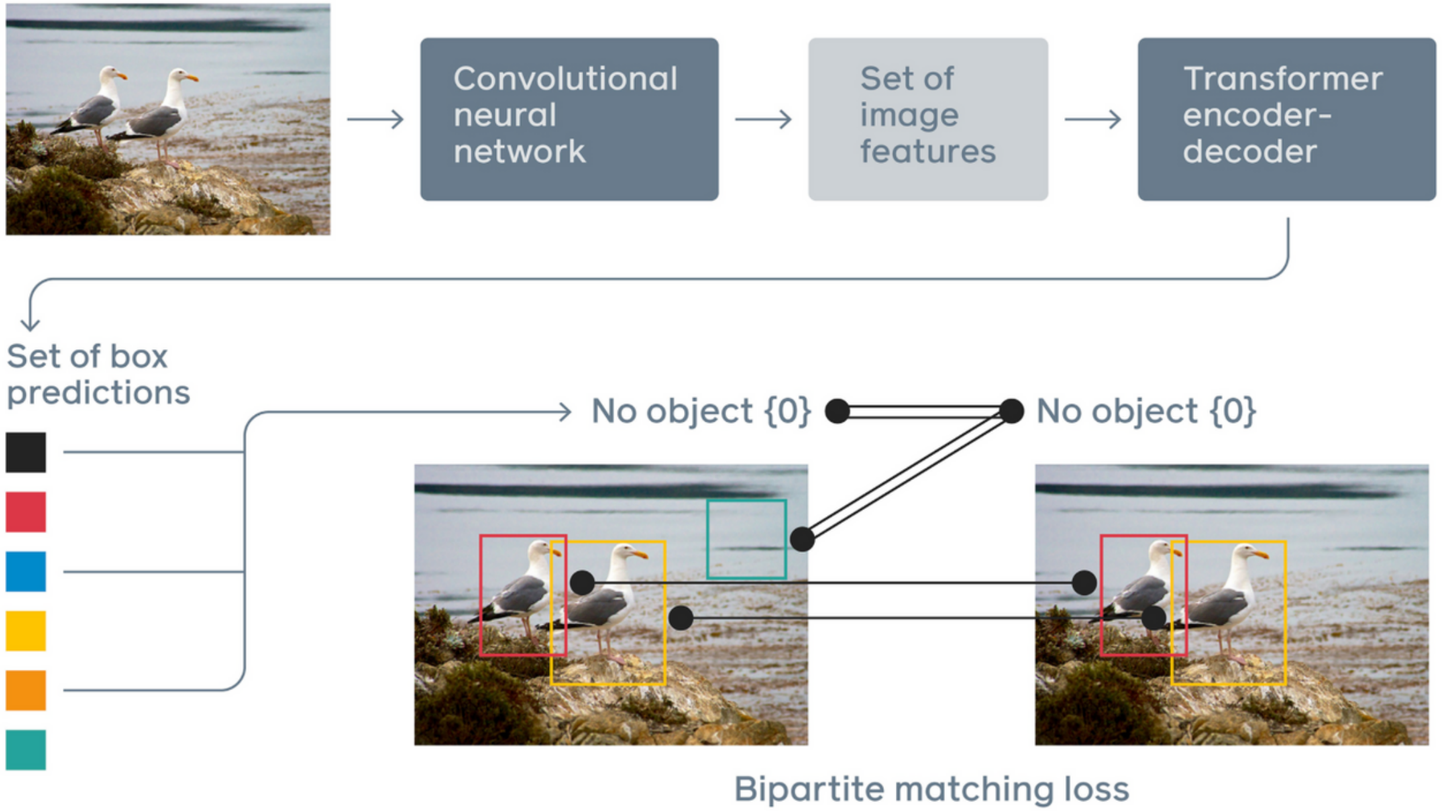


	ViT-H	Previous SOTA
ImageNet	88.55	88.5
ImageNet-ReaL	90.72	90.55
Cifar-10	99.50	99.37
Cifar-100	94.55	93.51
Pets	97.56	96.62
Flowers	99.68	99.63



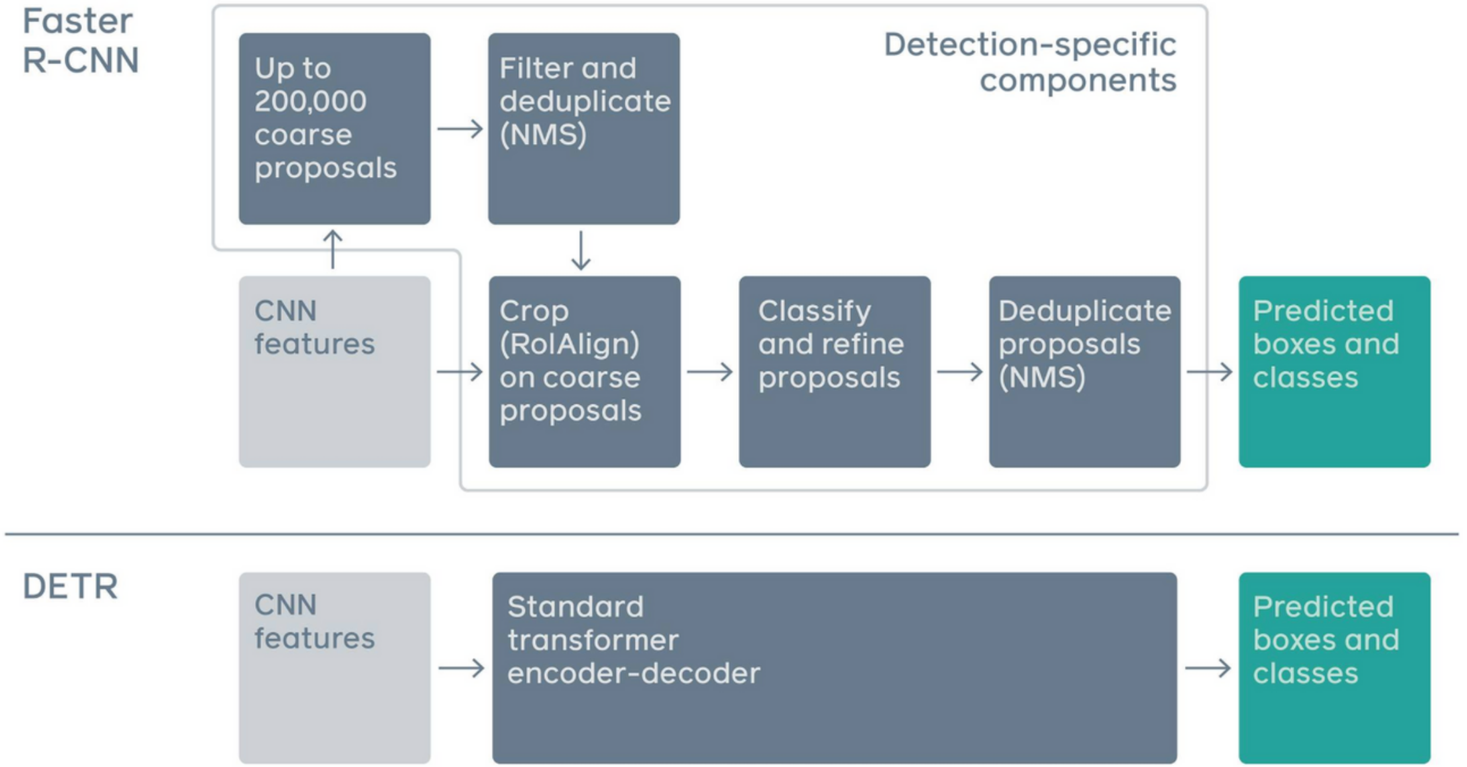
Dosovitskiy et al, “An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, ICLR 2021

DETR: Object Detection with Transformers



Carion et al, “End-to-End Object Detection with Transformers", ECCV 2020

DETR: Object Detection with Transformers



Carion et al, “End-to-End Object Detection with Transformers", ECCV 2020

Object Detection with Deep Learning

S22.CS7.505 Computer Vision
International Institute of Information Technology Hyderabad

March 25, 2022 | 1400 - 1530 | Himalaya Bodh 105