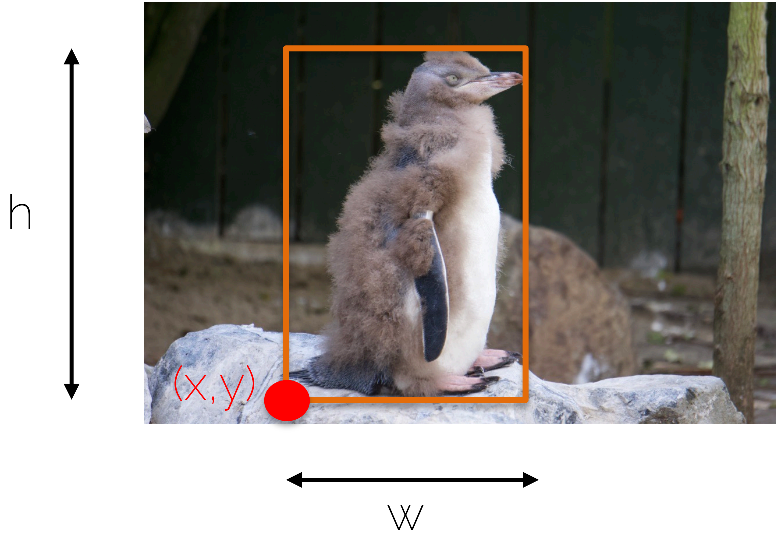
Object detection

Task definition

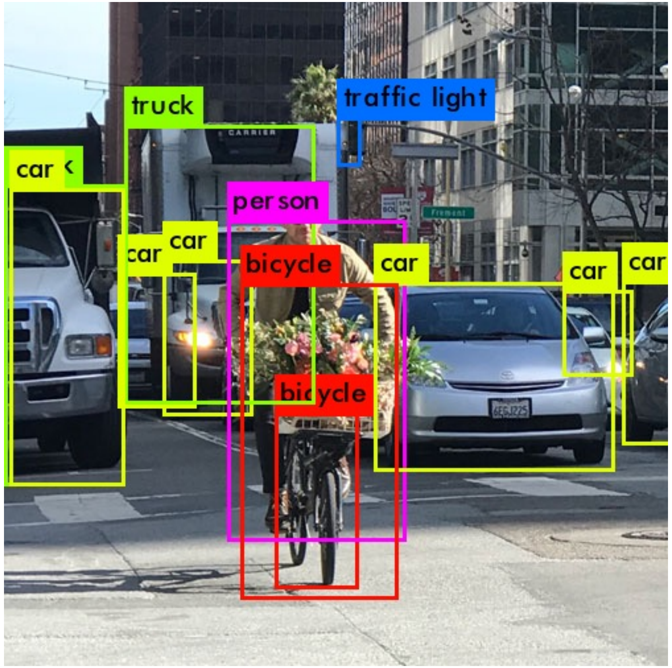
- Object detection problem



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

Task definition

- Object detection problem



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

+

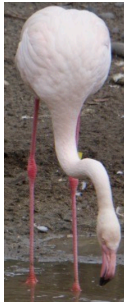
class

A bit of history

Traditional object detection methods

- 1. Template matching + sliding window

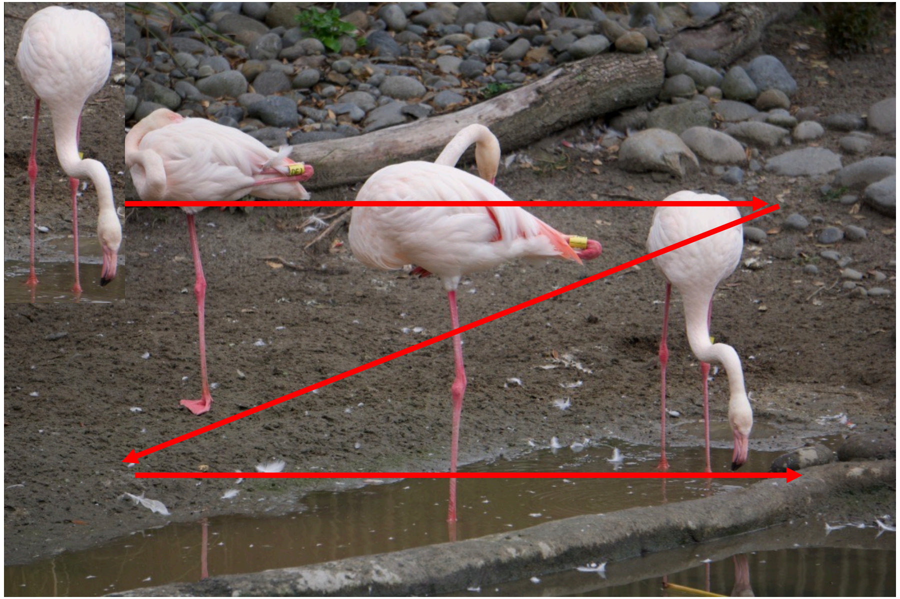
Image



Template

Traditional object detection methods

- 1. Template matching + sliding window



Image

Traditional object detection methods

- 1. Template matching + sliding window

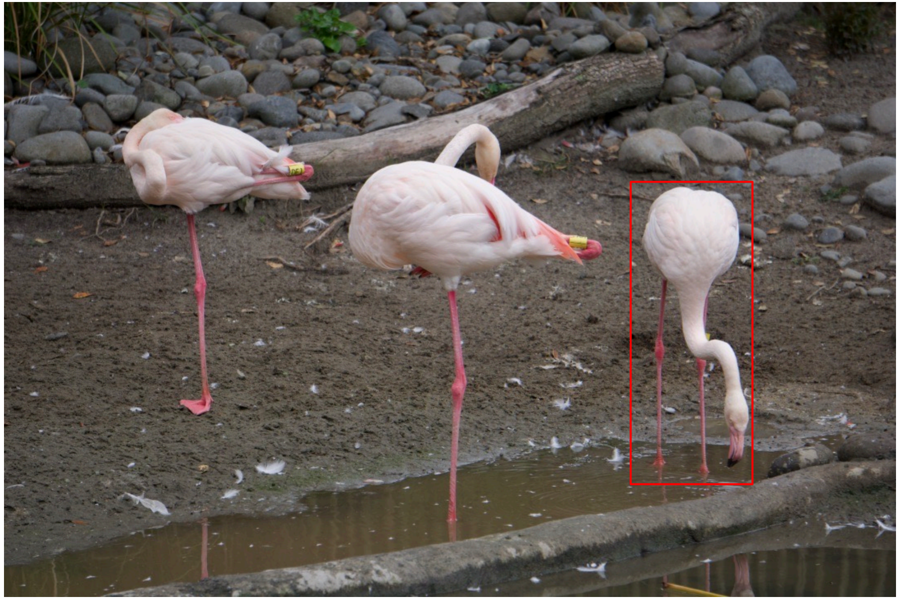
LOW correlation

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



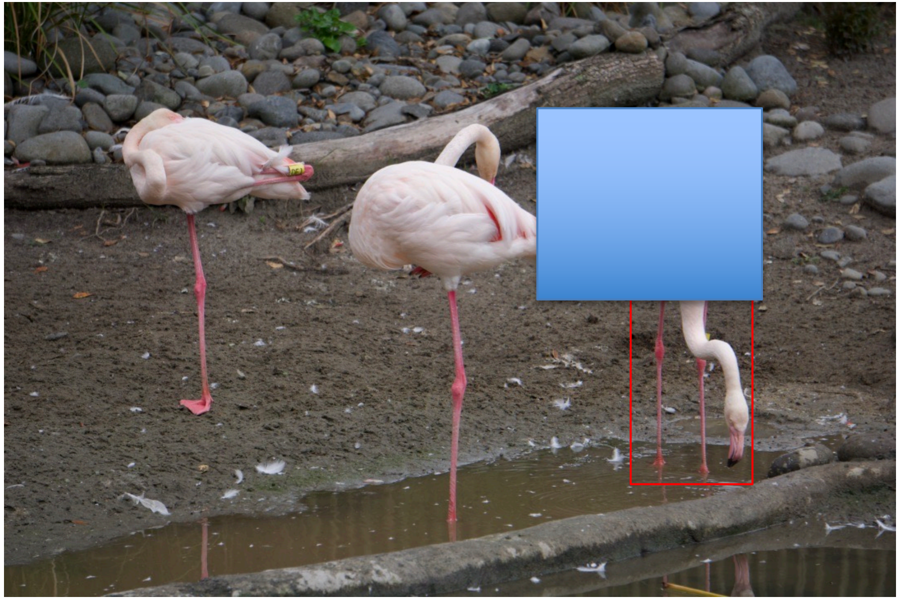
For every position you evaluate how much do the pixels in the image and template correlate

Image

HIGH correlation

Traditional object detection methods

- Problems of 1. Template matching + sliding window



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

Image

LOW correlation

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

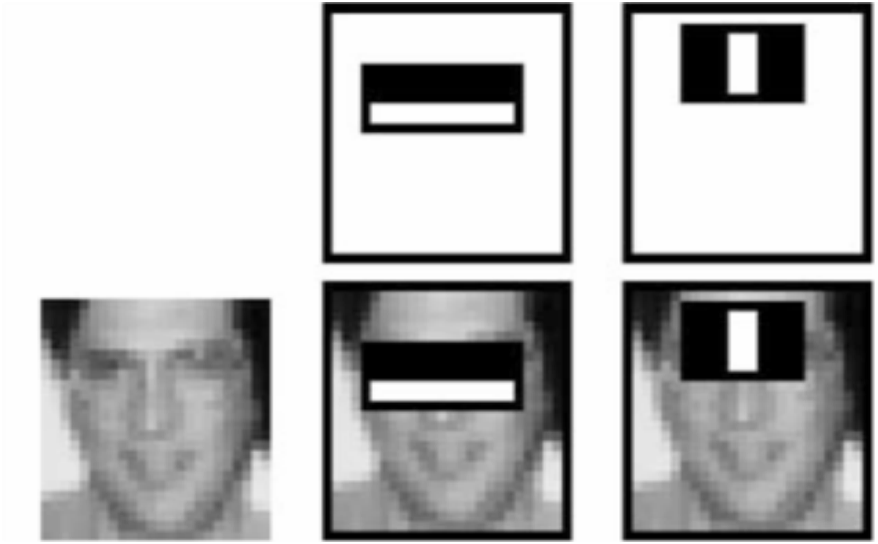
Viola-Jones detector

- 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

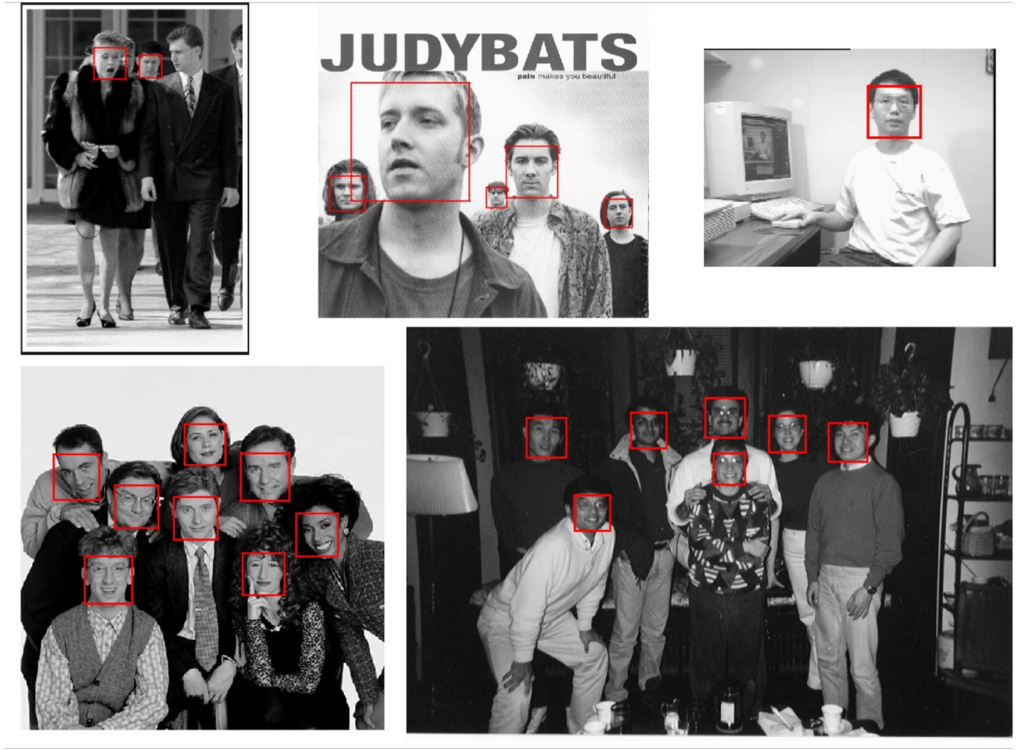
Viola and Jones. Rapid object detection using a boosted cascade of simple features. CVPR 2001.

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.

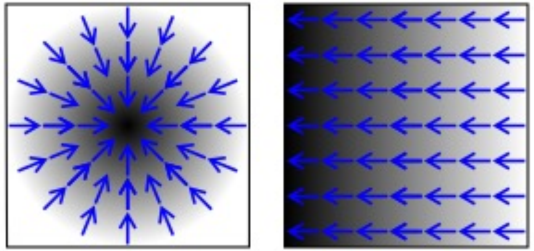
Viola-Jones detector



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

Histogram of Oriented Gradients

- 2. Feature extraction + classification



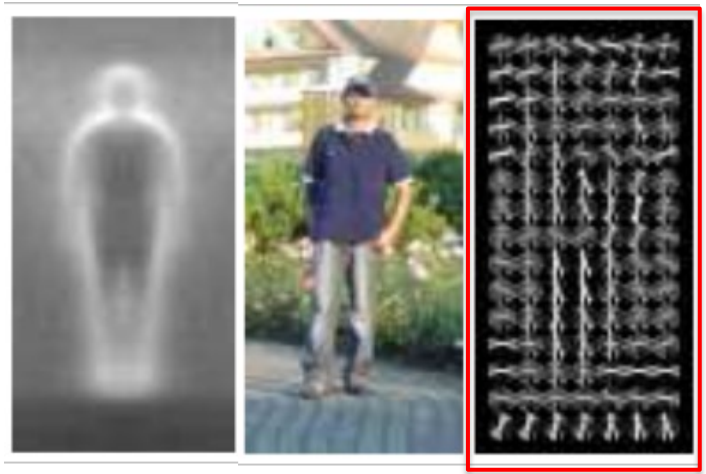
Gradient: blue arrows show the gradient, i.e., the direction of greatest change of the image.

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.

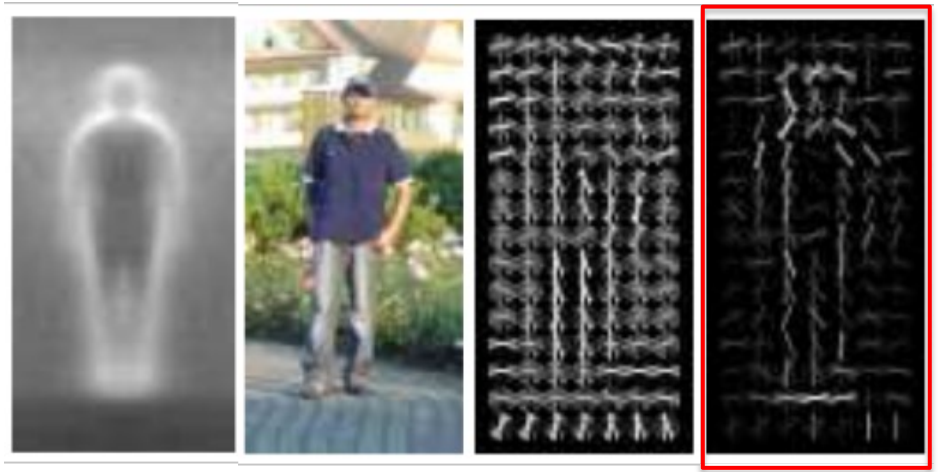
Histogram of Oriented Gradients

- 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

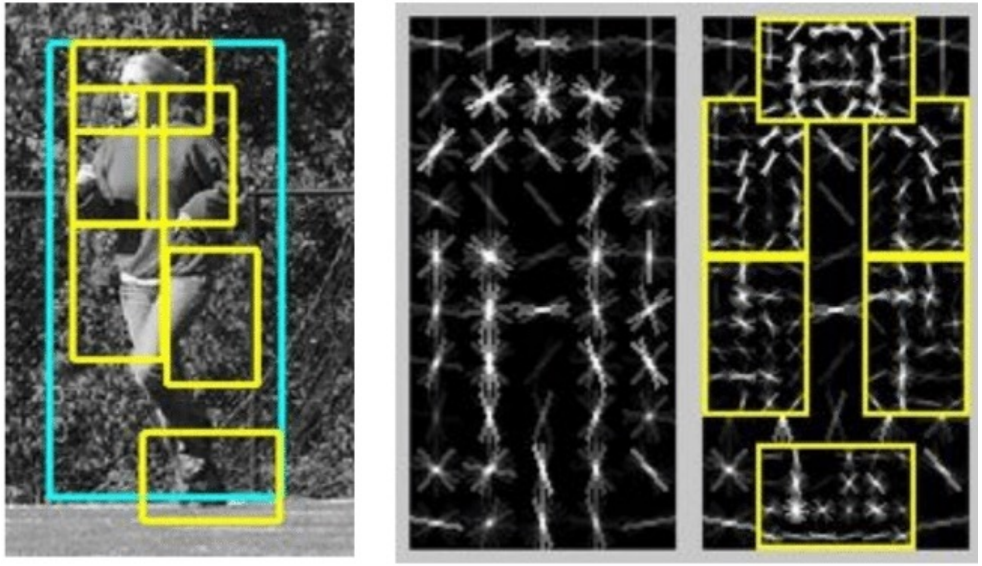


HOG features weighted by the positive SVM weights – the ones used for the pedestrian object classifier.

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

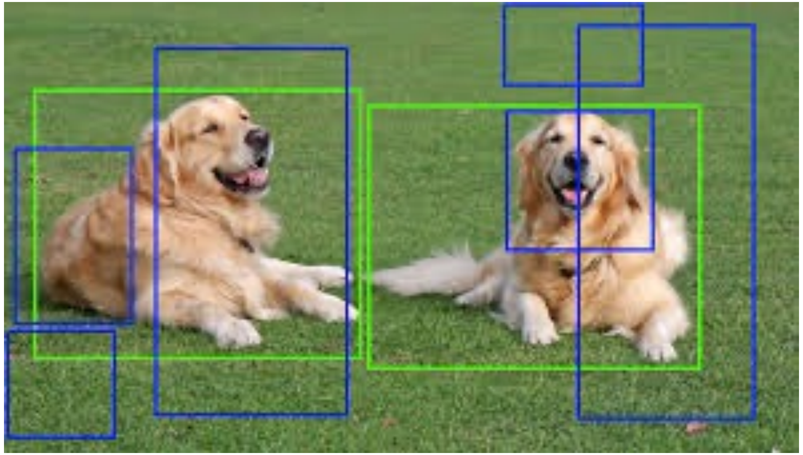


Felzenszwalb et al. A discriminatively trained, multiscale, deformable part model. CVPR 2008.

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

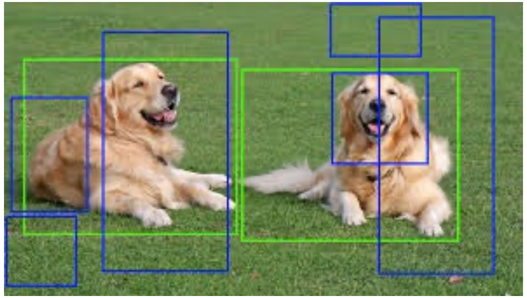


Very likely to be an object

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

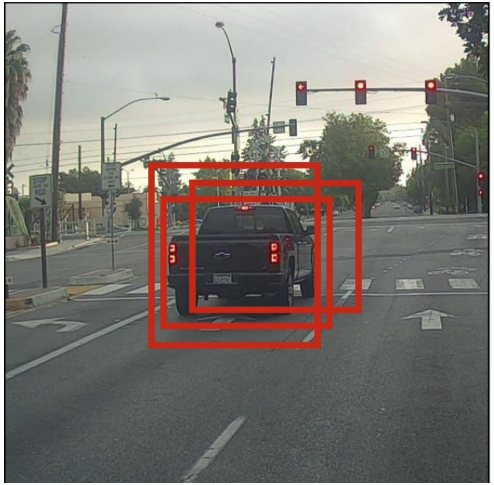
Object proposal methods

- **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$(b_i, b_j) > \lambda_{\mathbf{nms}}$ **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

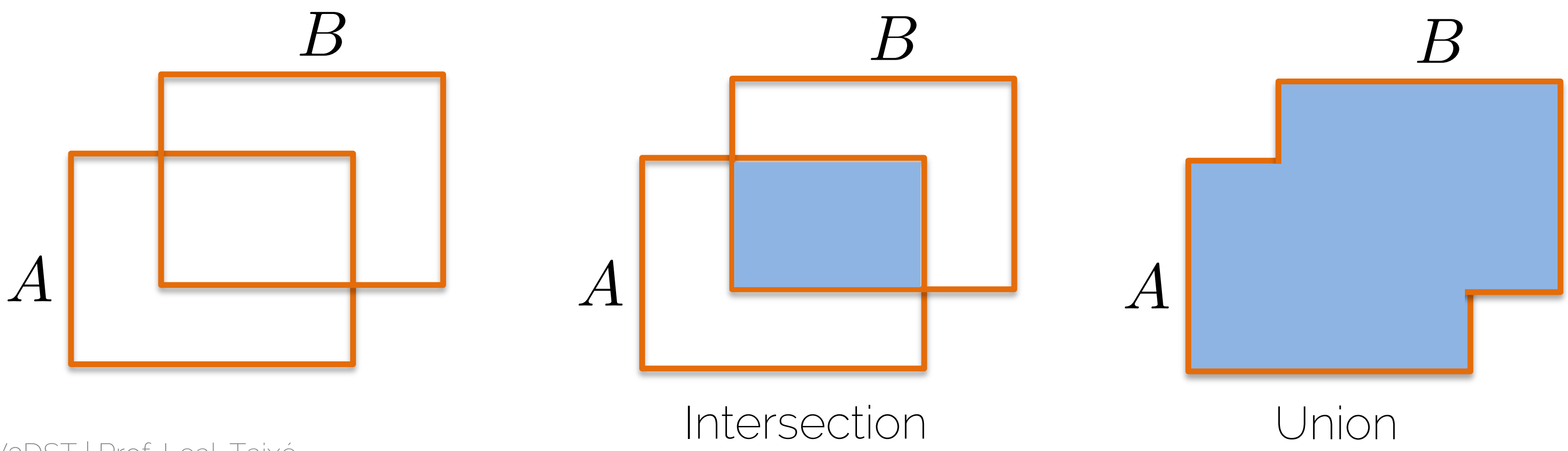
Overlap = to be defined

Score = depends on the task

Region overlap

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

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



Non-Maximum Suppression (NMS)

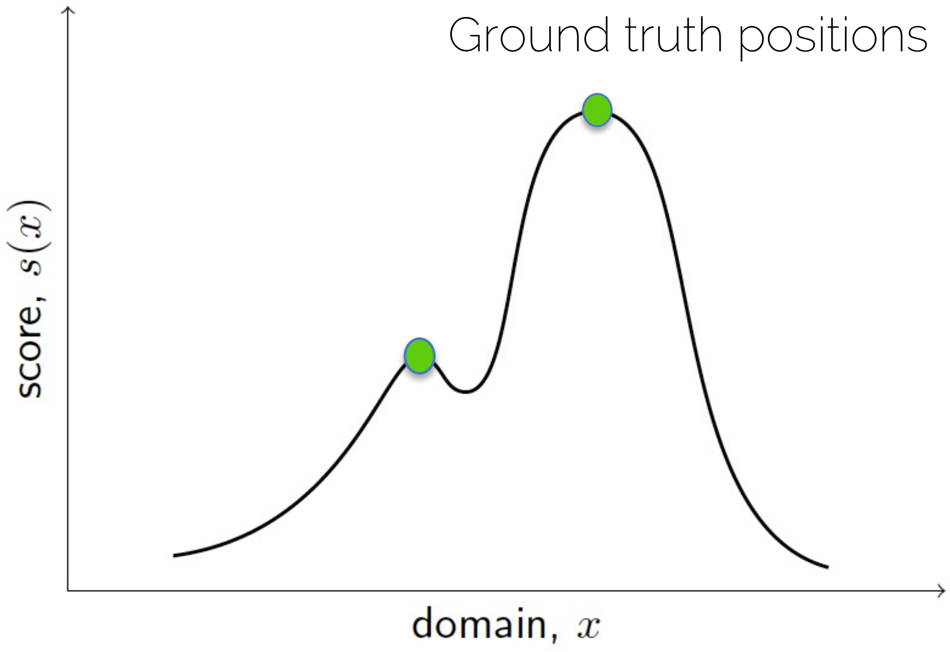
Algorithm 1 Non-Max Suppression

procedure NMS(B,c)
$B_{nms} \leftarrow \emptyset$
for $b_i \in B$ **do** — Start with anchor box i
$discard \leftarrow$ False
for $b_j \in B$ **do** — For another box j
if same$(b_i, b_j) > \boxed{\lambda_{\mathbf{nms}}}$ **then** — If they overlap
if score$(c, b_j) >$ score(c, b_i) **then**
$discard \leftarrow$ True — Discard box i if the score is lower than the score of j
if not $discard$ **then**
$B_{nms} \leftarrow B_{nms} \cup b_i$
return B_{nms}

Overlap = to be defined

Score = depends on the task

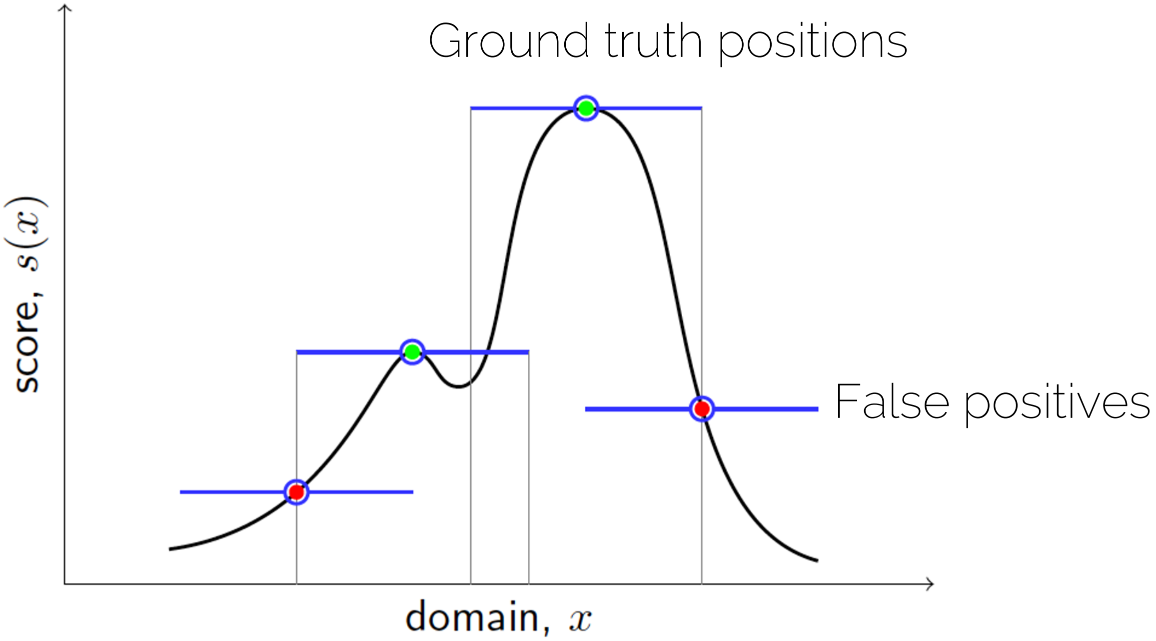
NMS: the problem



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

NMS: the problem

- Choosing a narrow threshold

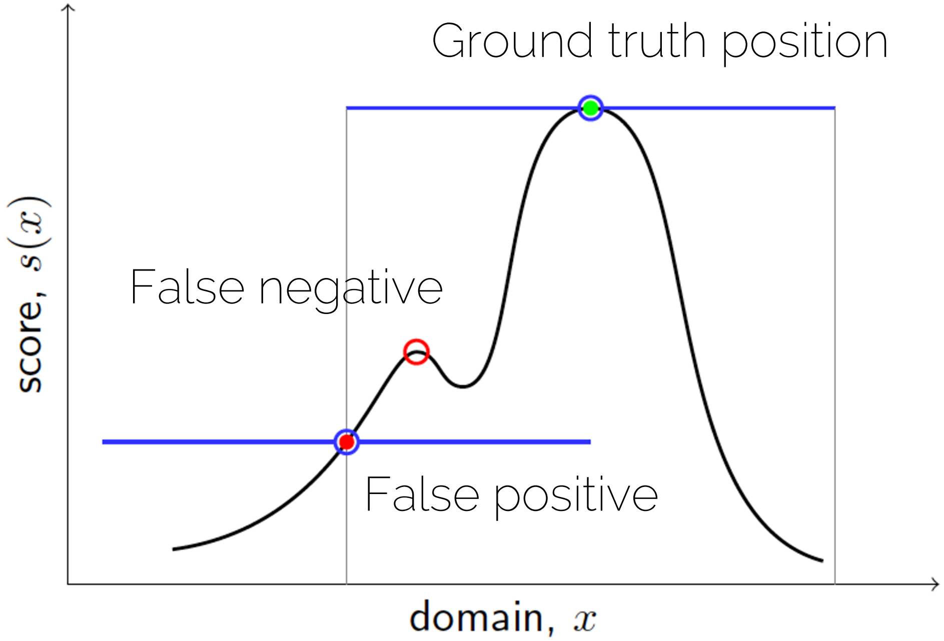


Low Precision

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

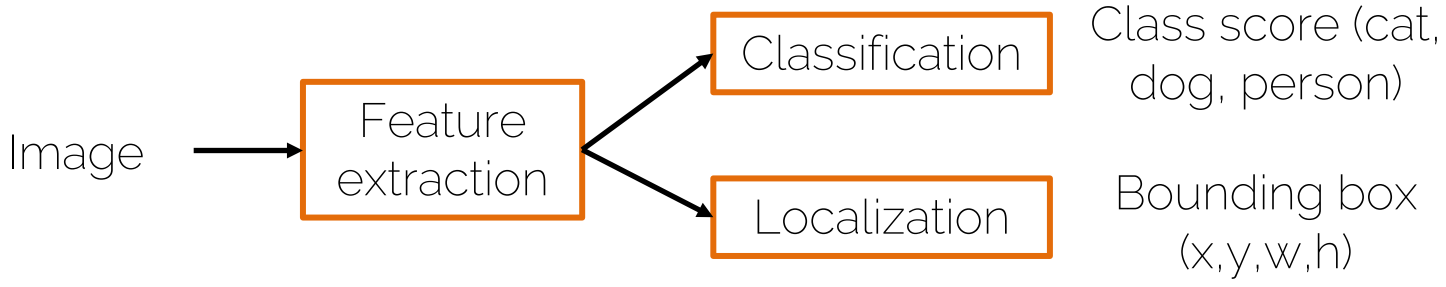
Non-Maximum Suppression (NMS)

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

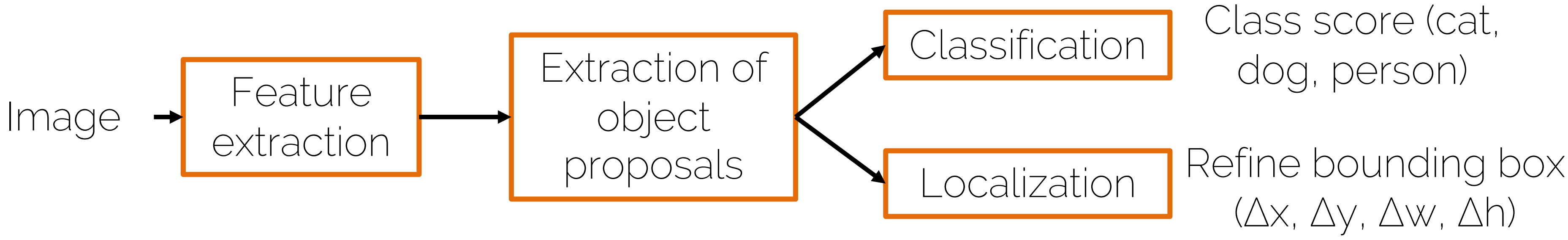
Learning-based detectors

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

Object detection