Université de Paris

Deep learning and applications – Part 2

Sylvain Lobry

11/11/2023

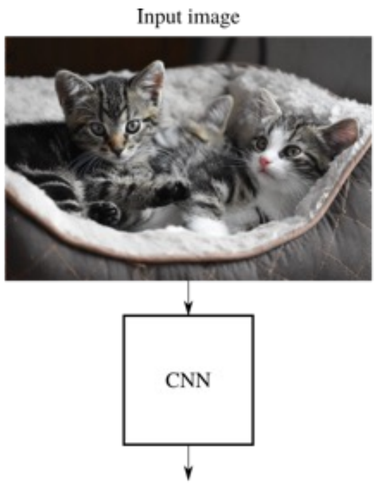
Object detection

What can you do with an image?

Input image

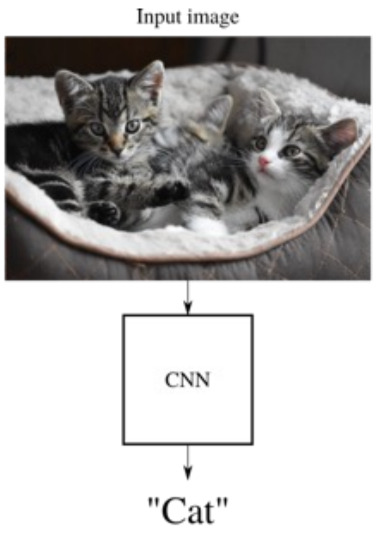
Object detection

What can you do with an image?



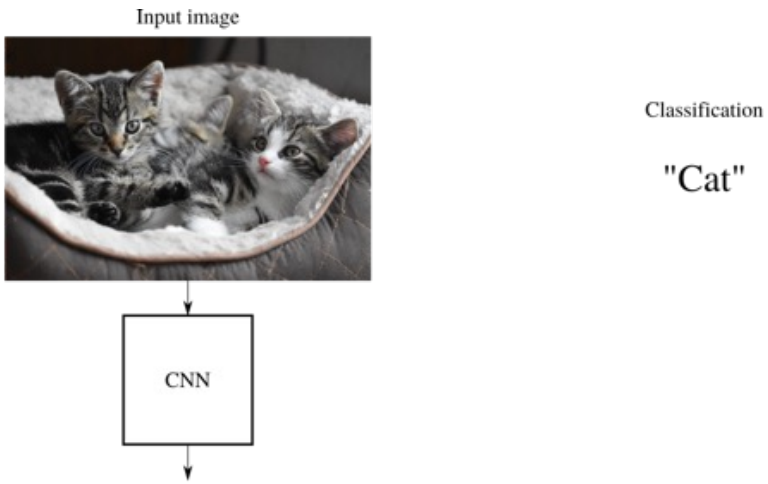
Object detection

What can you do with an image?



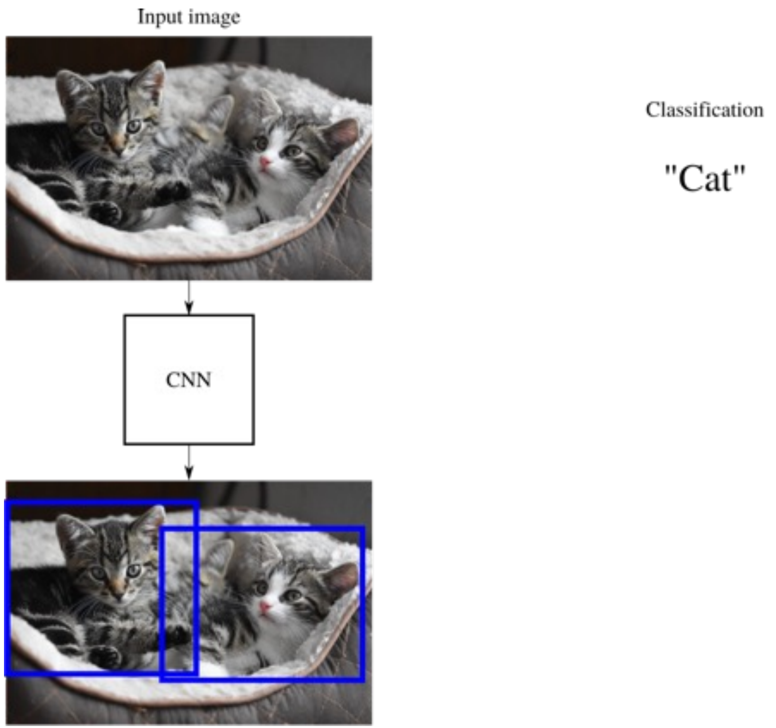
Object detection

What can you do with an image?



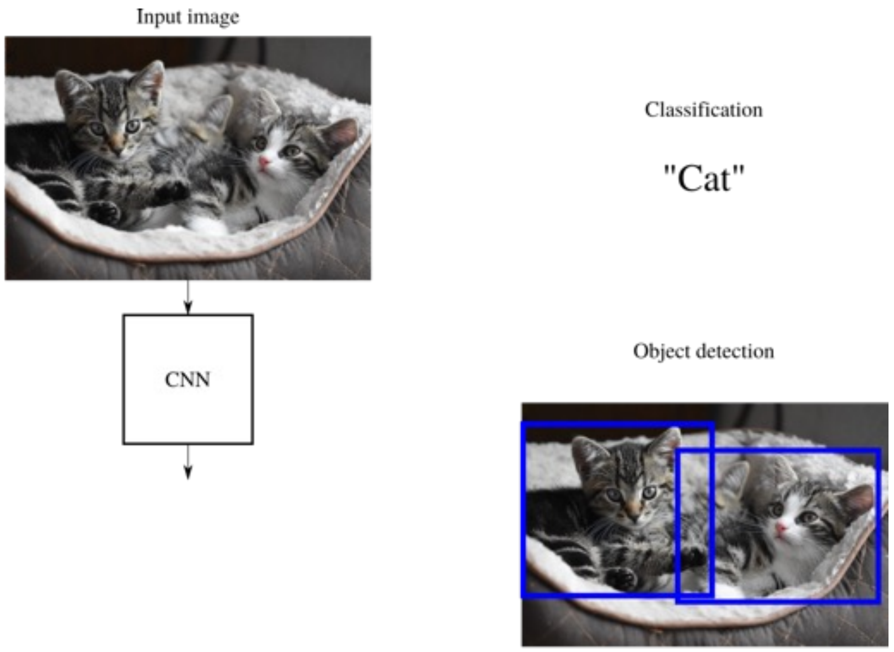
Object detection

What can you do with an image?



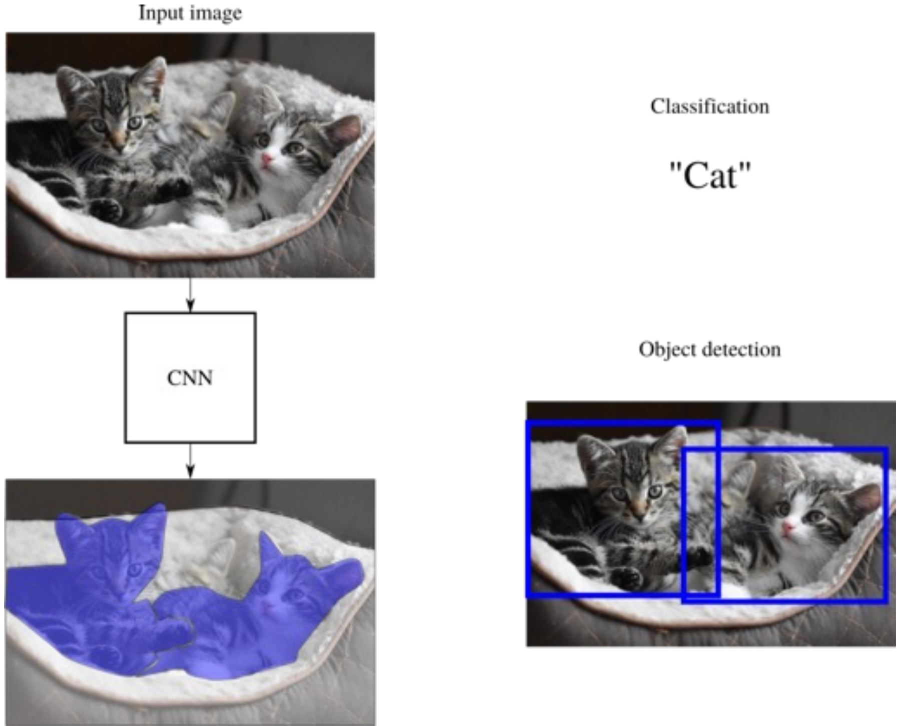
Object detection

What can you do with an image?



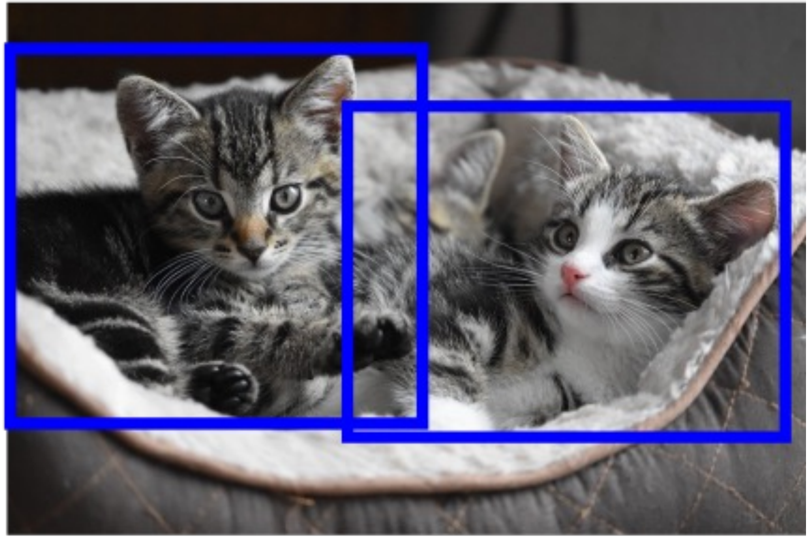
Object detection

What can you do with an image?



Object detection

Task of object detection

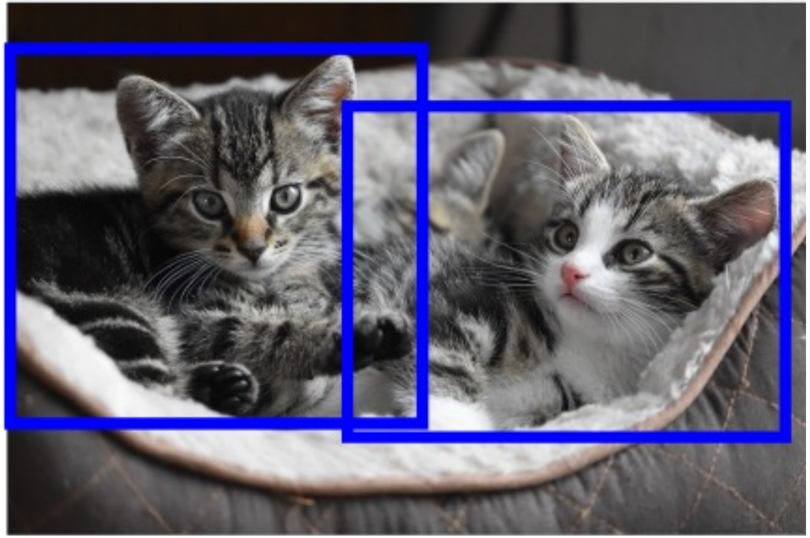


Objective:
get a set of bounding boxes associated to an objet

Output = $\{(\mathrm{Bbox}_1, class_1), \ldots (\mathrm{Bbox}_n, class_n)\}$

Object detection

Task of object detection



Compared to classification:
+ Spatial information
+ Can describe more
- More complicated

Compared to semantic segmentation:
+ Much more simple
+ Sufficient in most cases

Object detection

Why do we need it?

– Face recognition

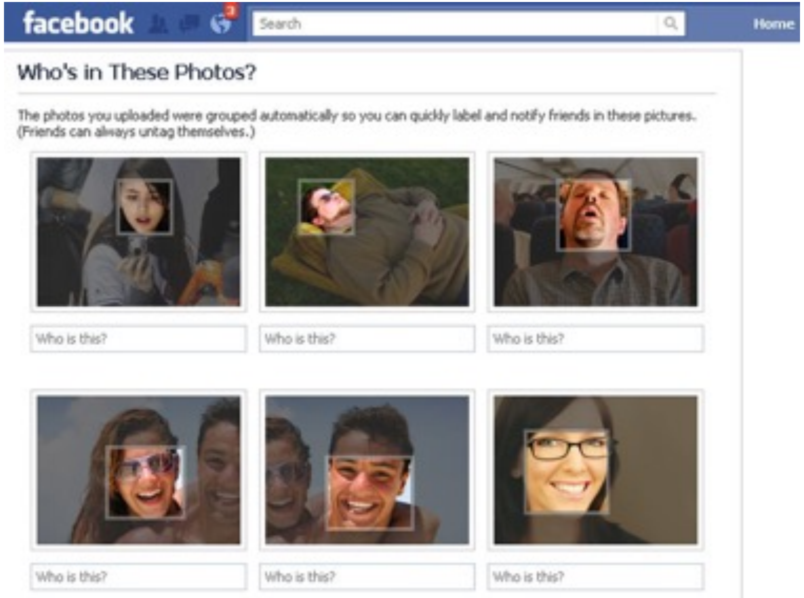
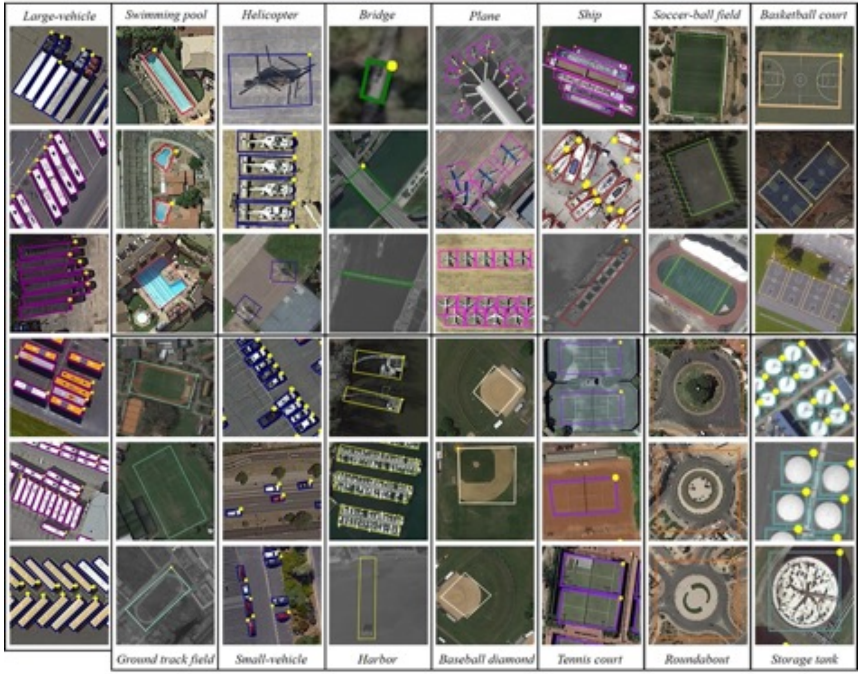


Image: CNET

Object detection

Why do we need it?

- Remote sensing

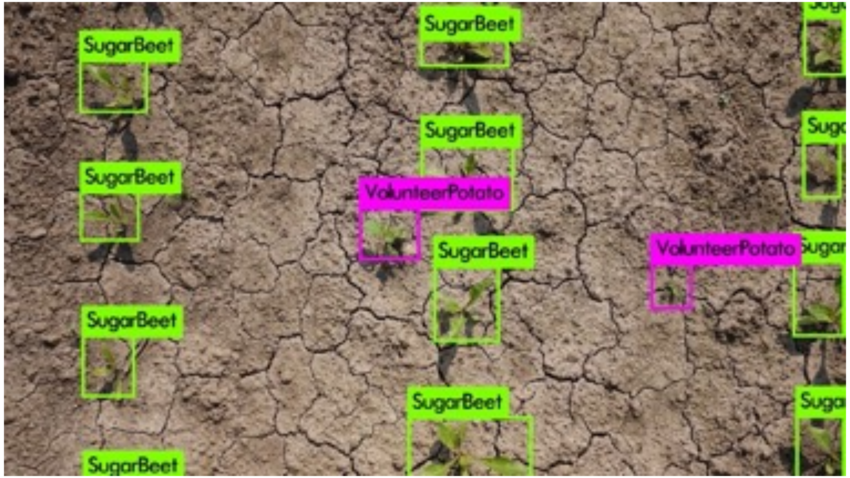


DOTA Dataset
Image source: https://captain-whu.github.io/DOTA/index.html

Object detection

Why do we need it?

- Security scans at airports
- Trash detection
- Crop monitoring
- Autonomous vehicles…



Object detection

Bounding box

Object detection

Bounding box

- Rectangle delineating the object

Object detection

Bounding box

- Rectangle delineating the object
- 3 options:
  - (x,y) of top left corner and width/height

Object detection

Bounding box

- Rectangle delineating the object
- 3 options:
  - (x,y) of top left corner and width/height
  - (x,y) of the center and width/height

Object detection

Bounding box

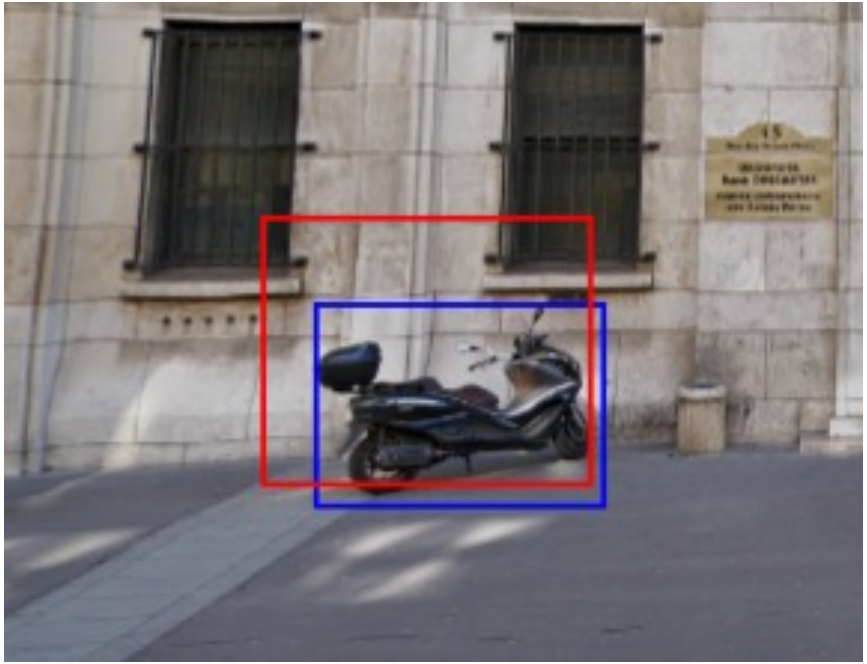
- Rectangle delineating the object
- 3 options:
  - (x,y) of top left corner and width/height
  - (x,y) of the center and width/height
  - (x,y) of top left and bottom right corners
- In any case, bounding box = 4 variables
- Coordinates and width/height are normalized by the size of the image.

Object detection

Evaluation of a bounding box

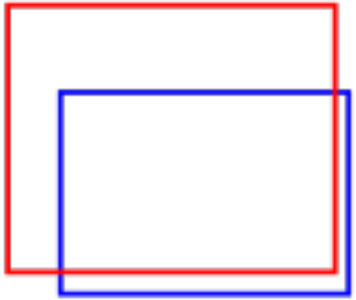
Object detection

Evaluation of a bounding box



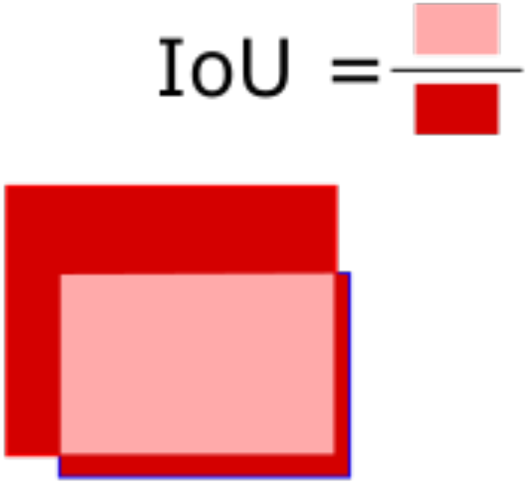
Object detection

Evaluation of a bounding box



Object detection

Evaluation of a bounding box



IoU (Intersection over Union)
a.k.a. Jaccard index

IoU is then thresholded to determine whether
The bounding box is accurate

Object detection

Naïve object detection

Idea: We know how to classify an image. Let’s classify bounding boxes (sliding window)!

Object detection

Naïve object detection

Idea: We know how to classify an image. Let’s classify bounding boxes (sliding window)!

Object detection

Naïve object detection

Idea: We know how to classify an image. Let’s classify bounding boxes (sliding window)!

(P, x, y, w, h, c1, c2)

(0, _, _, _, _, _,

Object detection

Naïve object detection

Idea: We know how to classify an image. Let's classify bounding boxes (sliding window)!

Object detection

Naïve object detection

Idea: We know how to classify an image. Let's classify bounding boxes (sliding window)!

Object detection

Naïve object detection

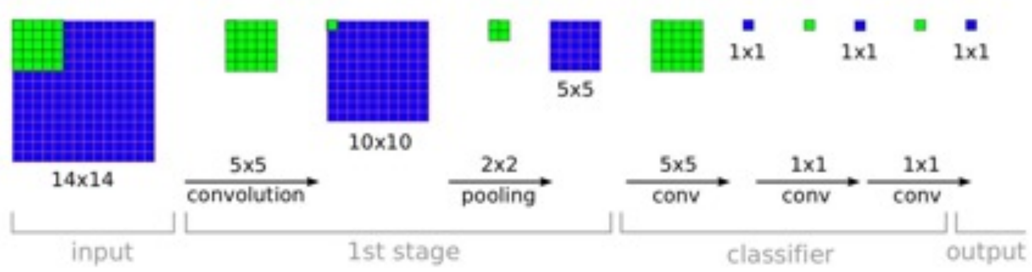
Idea: We know how to classify an image. Let’s classify bounding boxes (sliding window)!

- And you can slide a new window with a different bounding box size…
- Not efficient AT ALL…

Object detection

Naïve object detection – take 2

Previous approach does not really work as you have to make as many inference pass as the number of Bounding boxes tested…
Idea from Sermanet et. al. : sliding window can be seen as a convolution!
It could be as a Fully Convolutionnal Network (FCN).



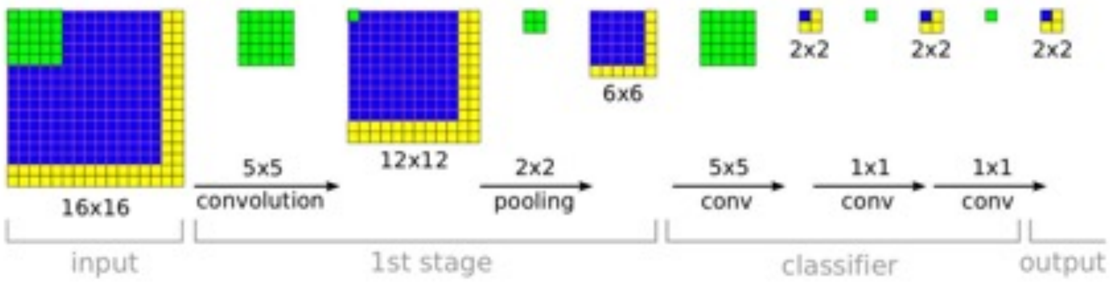
Careful: only spatial dimensions indicated.
Multi-dimensional output!

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Object detection

Naïve object detection – take 2

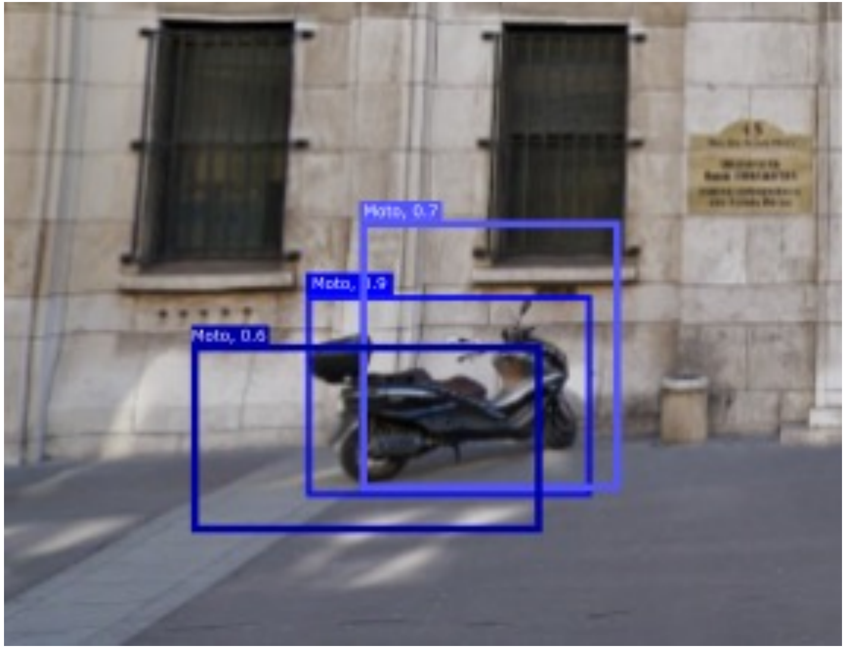
Previous approach does not really work as you have to make as many inference pass as the number of Bounding boxes tested…
Idea from Sermanet et. al. : sliding window can be seen as a convolution!
It could be as a Fully Convolutionnal Network (FCN).



Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2013). Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229*.

Object detection

Non-maximum suppresion



Problem: you might have more than one detection for each object…

Object detection

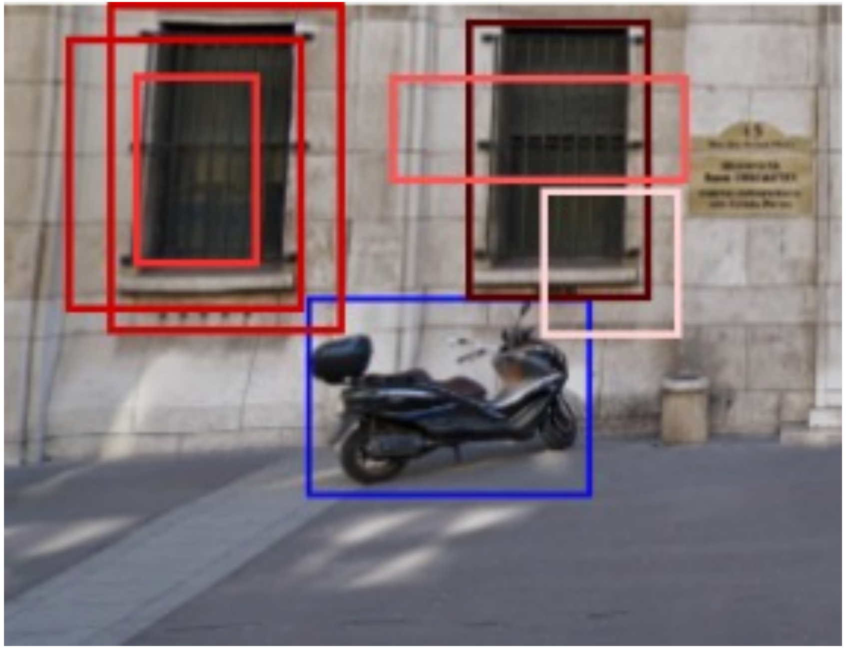
Non-maximum suppresion

Problem: you might have more than one detection for each object…

We want to remove least confident predictions: Non-maximum suppression (NMS)

Object detection

Non-maximum suppresion



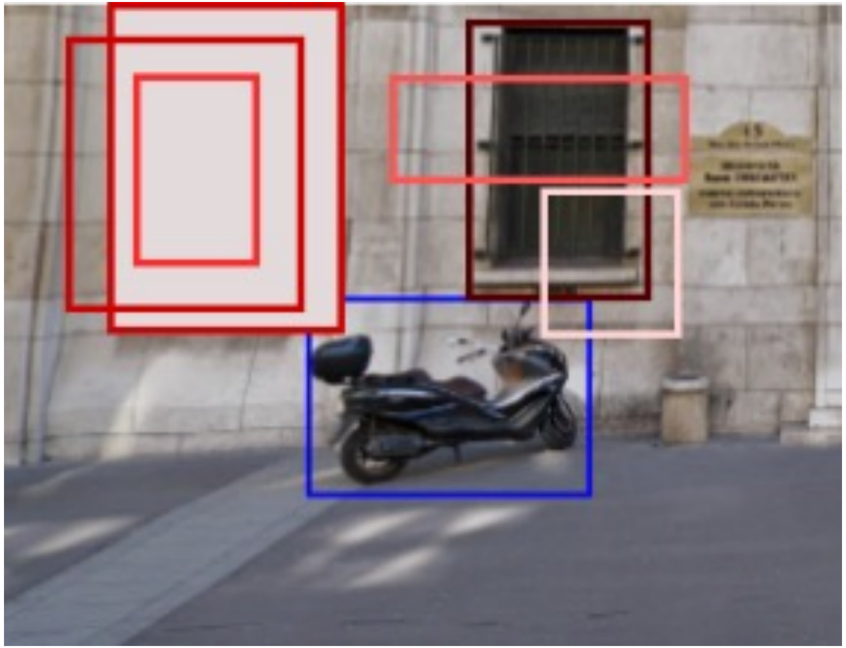
Example on windows: darker frame = better confidence on the detection.

Step1: for each class, order bounding boxes by decreasing order of confidence

Step2: select the most confident box as the reference.

Object detection

Non-maximum suppresion

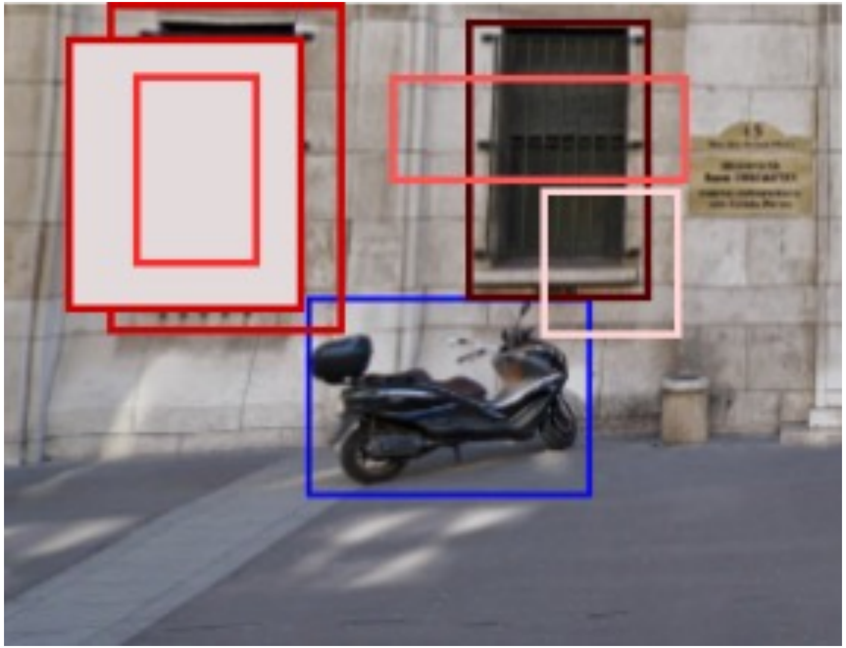


Example on windows: darker frame = better confidence on the detection.

Step1: for each class, order bounding boxes by decreasing order of confidence
Step2: select the most confident box as the reference.
Step3: select the second most confident box:
-- IoU with reference > threshold ?
--- remove
-- else
--- keep

Object detection

Non-maximum suppresion

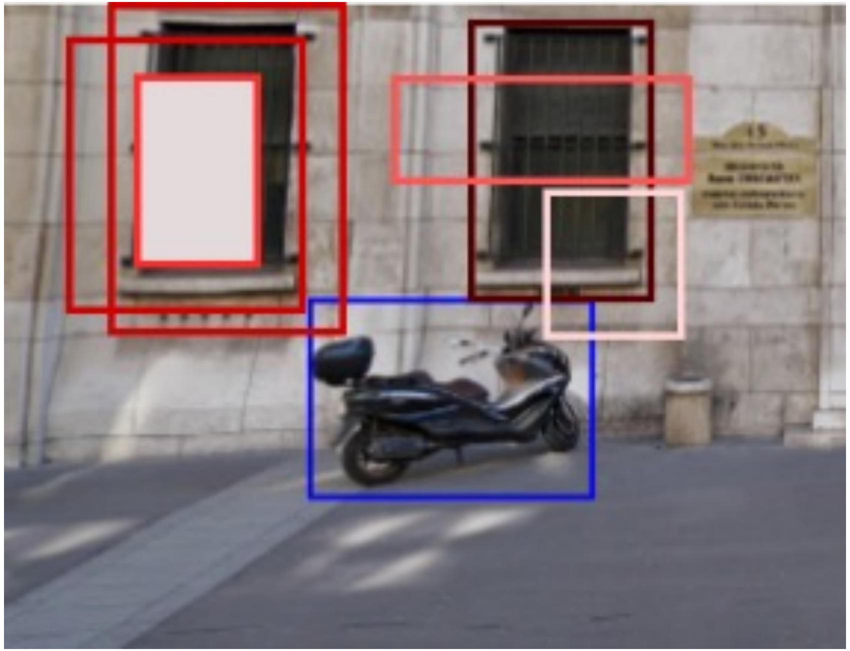


Example on windows: darker frame = better confidence on the detection.

Step1: for each class, order bounding boxes by decreasing order of confidence
Step2: select the most confident box as the reference.
Step3: select the second most confident box:
-- IoU with reference > threshold ?
--- remove
-- else
--- keep

Object detection

Non-maximum suppresion



Example on windows: darker frame = better confidence on the detection.

Step1: for each class, order bounding boxes by decreasing order of confidence

Step2: select the most confident box as the reference.

Step3: select the second most confident box:

-- IoU with reference > threshold ?

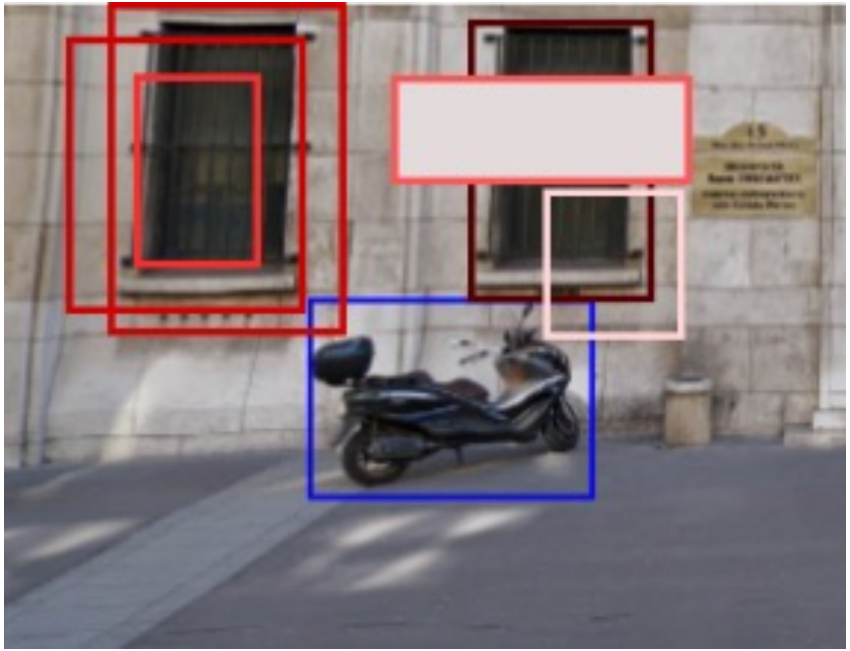
--- remove

-- else

--- keep

Object detection

Non-maximum suppresion

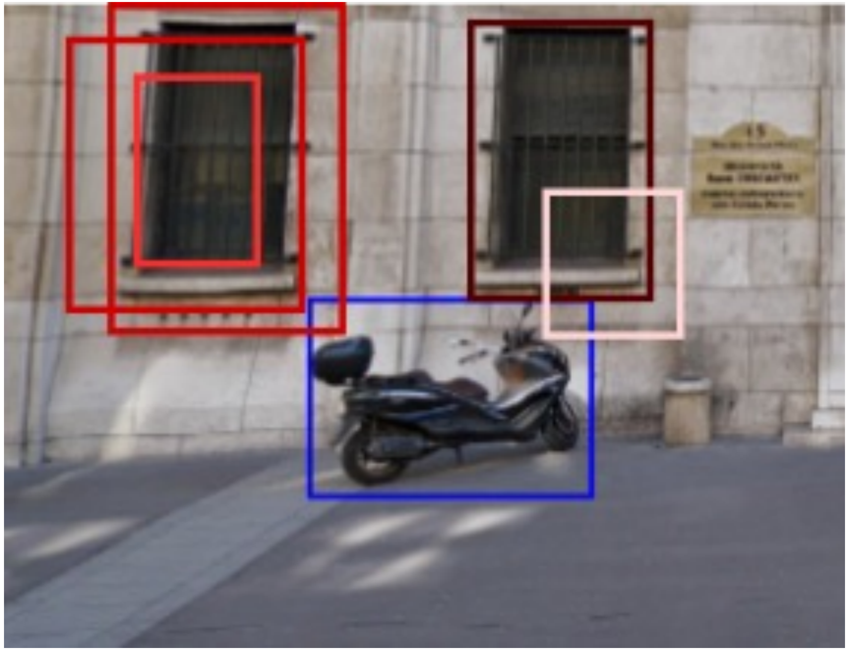


Example on windows: darker frame = better confidence on the detection.

Step1: for each class, order bounding boxes by decreasing order of confidence
Step2: select the most confident box as the reference.
Step3: select the second most confident box:
-- IoU with reference > threshold ?
--- remove
-- else
--- keep

Object detection

Non-maximum suppresion

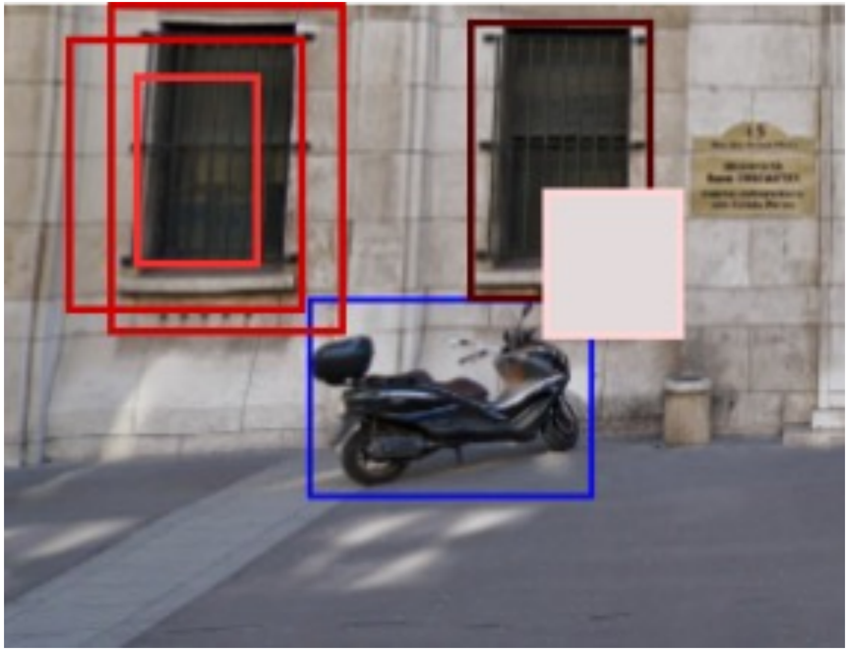


Example on windows: darker frame = better confidence on the detection.

Step1: for each class, order bounding boxes by decreasing order of confidence
Step2: select the most confident box as the reference.
Step3: select the second most confident box:
-- IoU with reference > threshold ?
--- remove
-- else
--- keep

Object detection

Non-maximum suppresion

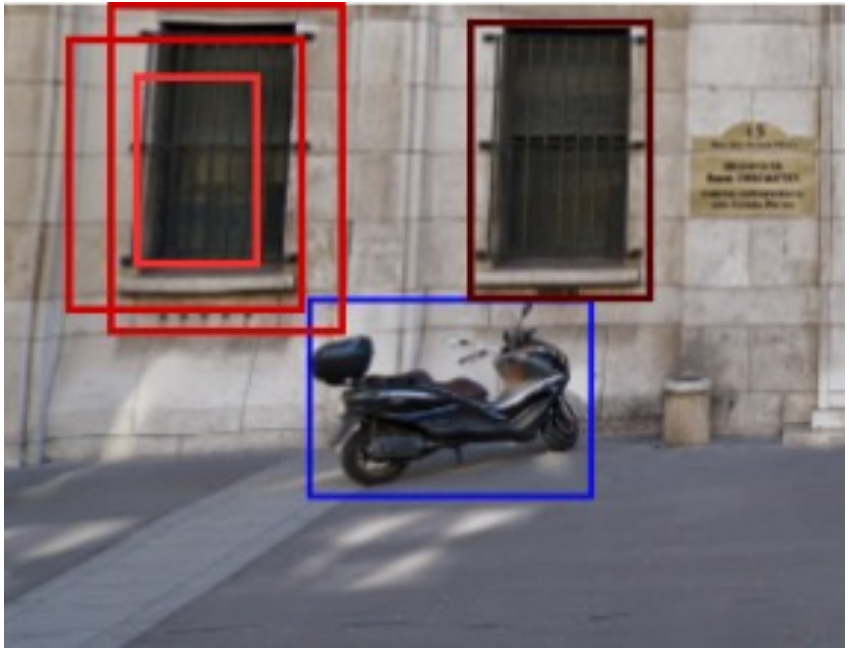


Example on windows: darker frame = better confidence on the detection.

Step1: for each class, order bounding boxes by decreasing order of confidence
Step2: select the most confident box as the reference.
Step3: select the second most confident box:
-- IoU with reference > threshold ?
--- remove
-- else
--- keep

Object detection

Non-maximum suppresion

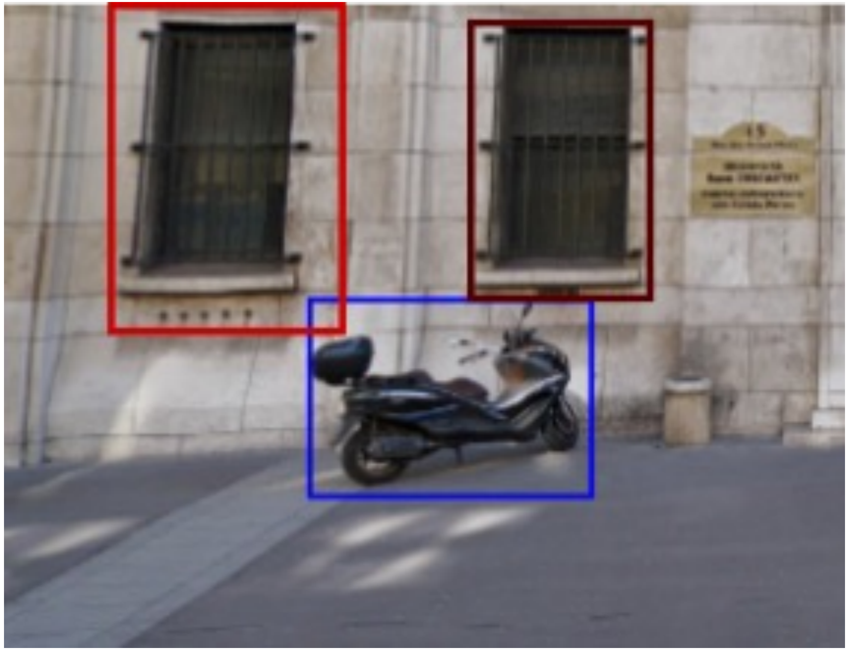


Example on windows: darker frame = better confidence on the detection.

Step1: for each class, order bounding boxes by decreasing order of confidence
Step2: select the most confident box as the reference.
Step3: select the second most confident box:
-- IoU with reference > threshold ?
--- remove
-- else
--- keep

Object detection

Non-maximum suppresion



Example on windows: darker frame = better confidence on the detection.

Step1: for each class, order bounding boxes by decreasing order of confidence

Step2: select the most confident box as the reference.

Step3: select the second most confident box:

-- IoU with reference > threshold ?

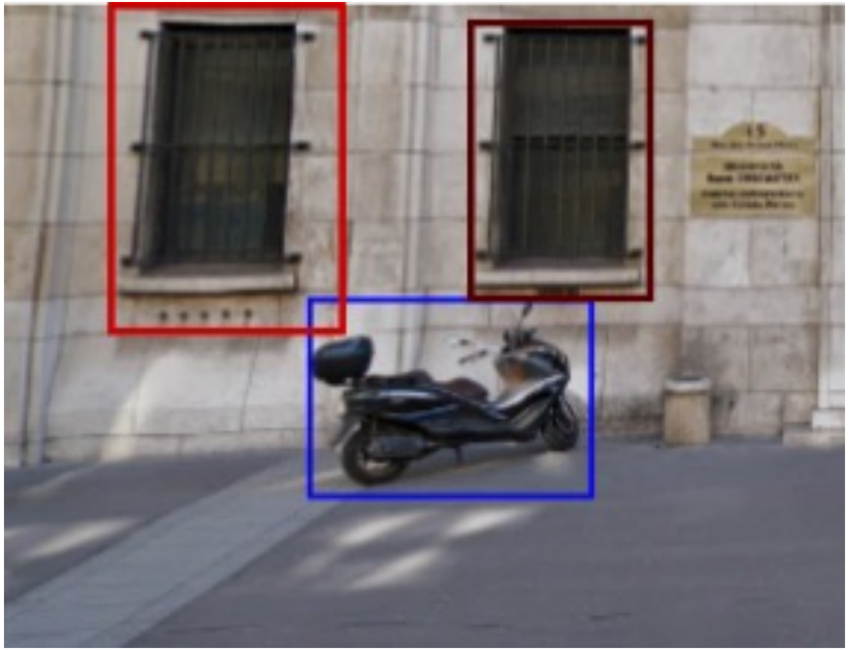
--- remove

-- else

--- keep

Object detection

Non-maximum suppresion



Important: to be done for each object separatly !

Object detection

Recap

We have seen:

- What is a bounding box
- How to evaluate its accuracy
- How to suppress multiple detections

Questions?

Object detection

Recap

We have seen:

- What is a bounding box
- How to evaluate its accuracy
- How to suppress multiple detections

But… How do you get the bounding boxes?
2 options:

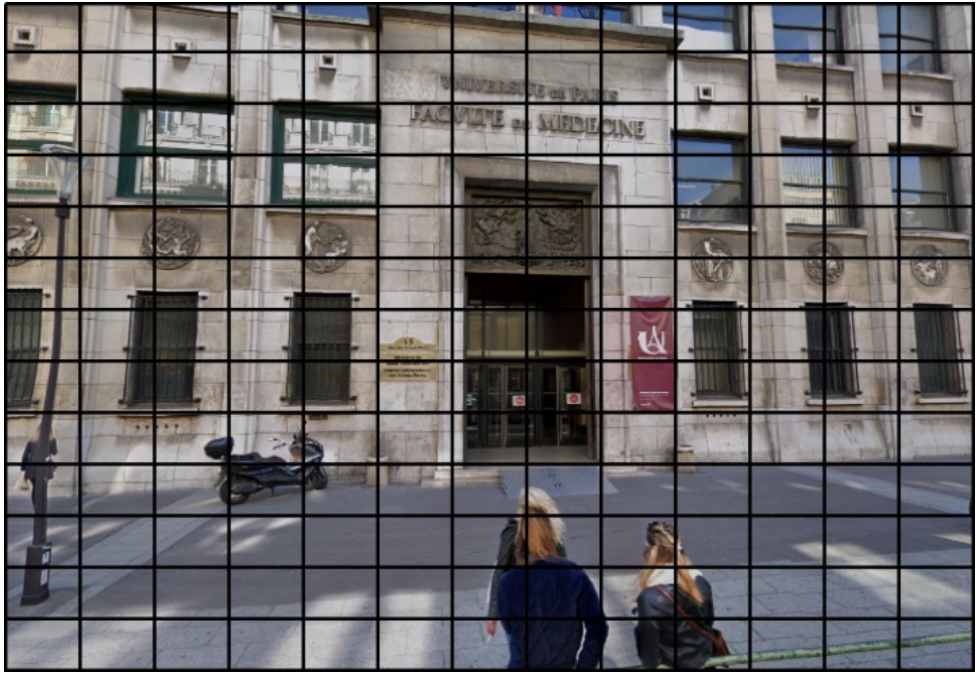
- Hardcoded bounding boxes (a.k.a. anchor boxes)
- Try to predict the bounding boxes (a.k.a. region proposal)

Object detection

Anchor box

Object detection

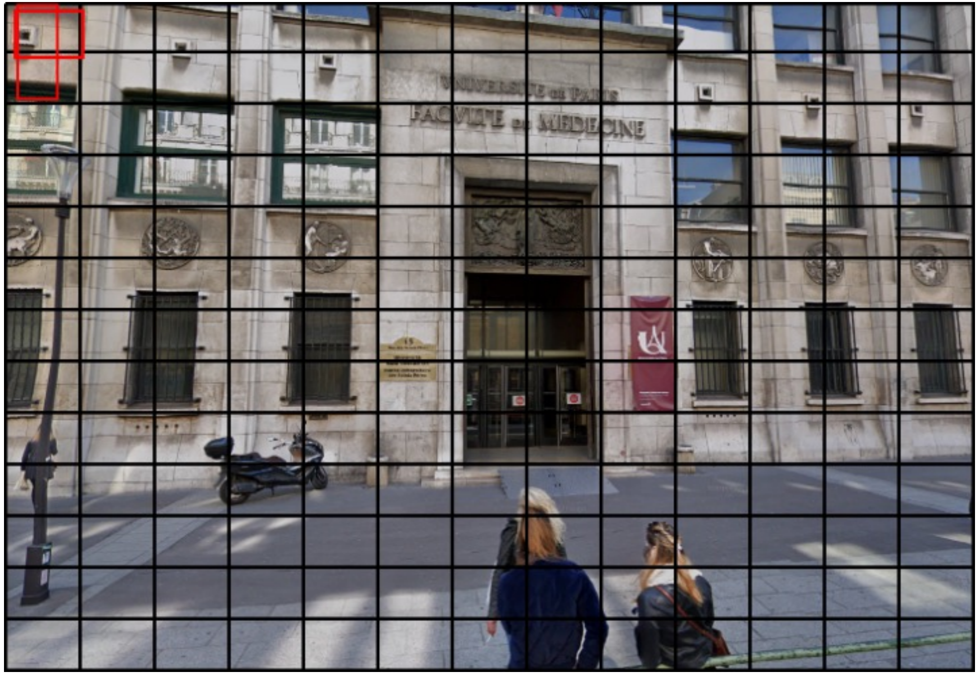
Anchor box



- Divide the image into a grid

Object detection

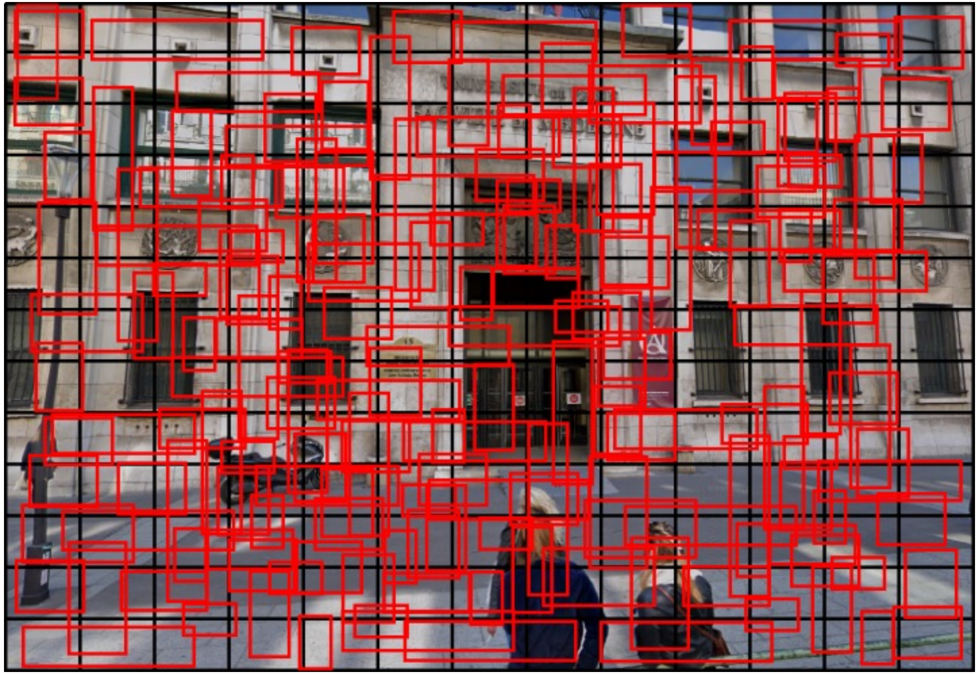
Anchor box



- Divide the image into a grid
- For each cell, define B bounding boxes (here B = 2)

Object detection

Anchor box



- Divide the image into a grid
- For each cell, define B bounding boxes (here B = 2)
- Prediction for each grid cell:

$$(\underbrace{x_1, y_1, w_1, h_1, p_1}_{\text{Box 1}}, \dots, \underbrace{x_B, y_B, h_B, w_B, p_B}_{\text{Box B}}, \underbrace{c_1, c_2}_{\text{Classes}})$$

Object detection

YOLO

Put everything we have seen until now (prediction based on anchor boxes, NMS): YOLO

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 779-788). ISO 690

Object detection

Recap

We have seen:

- What is a bounding box
- How to evaluate its accuracy
- How to suppress multiple detections

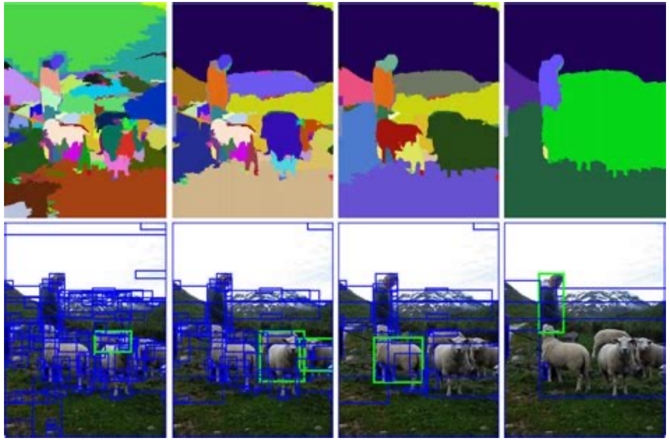
But… How do you get the bounding boxes?
2 options:

- Hardcoded bounding boxes (a.k.a. anchor boxes)
- Try to predict the bounding boxes (a.k.a. region proposal)

Object detection

R-CNN

1) Region proposal algorithm: selective search [1]: semantic segmentation + grouping

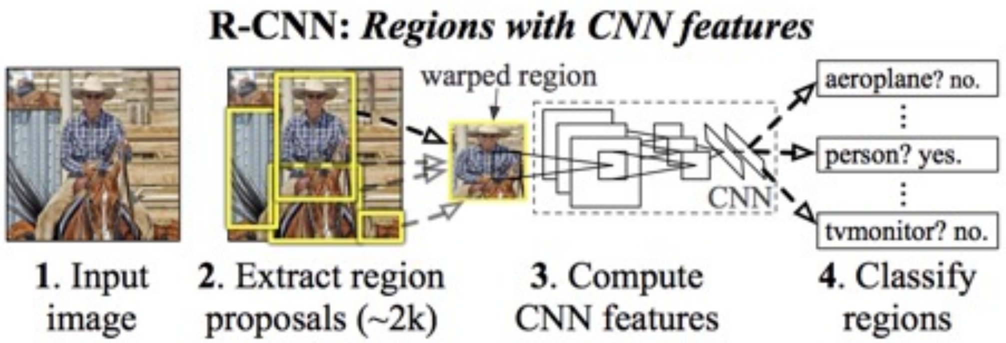


[1] J.Uijlings, K.van de Sande,T.Gevers, and A.Smeulders. Selective search for object recognition. *IJCV*, 2013
[2] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

Object detection

R-CNN

1) Region proposal algorithm: selective search [1]: semantic segmentation + grouping -> select ~2000 regions
2) Classification of the regions by a CNN -> R-CNN [2]



[1] J.Uijlings, K.van de Sande,T.Gevers, and A.Smeulders. Selective search for object recognition. *IJCV*, 2013
[2] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

Object detection

R-CNN vs YOLO

R-CNN: Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. CVPR
YOLO: Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. CVPR

Improvements over the two types of algorithms in the past years

[1] J.Uijlings, K.van de Sande,T.Gevers, and A.Smeulders. Selective search for object recognition. *IJCV*, 2013
[2] Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 580-587).

Université de Paris

Let’s code!