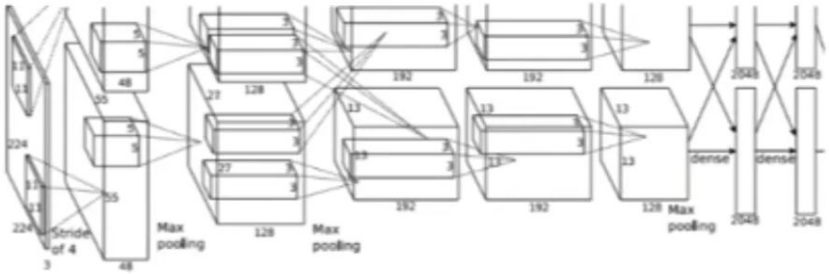
YOLO

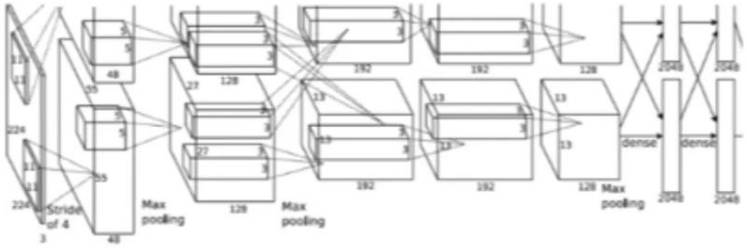
Marc Masias
Advanced Machine Learning 2024
Màster de Ciència de Dades

Object detection

Classification vs. Detection



CAT

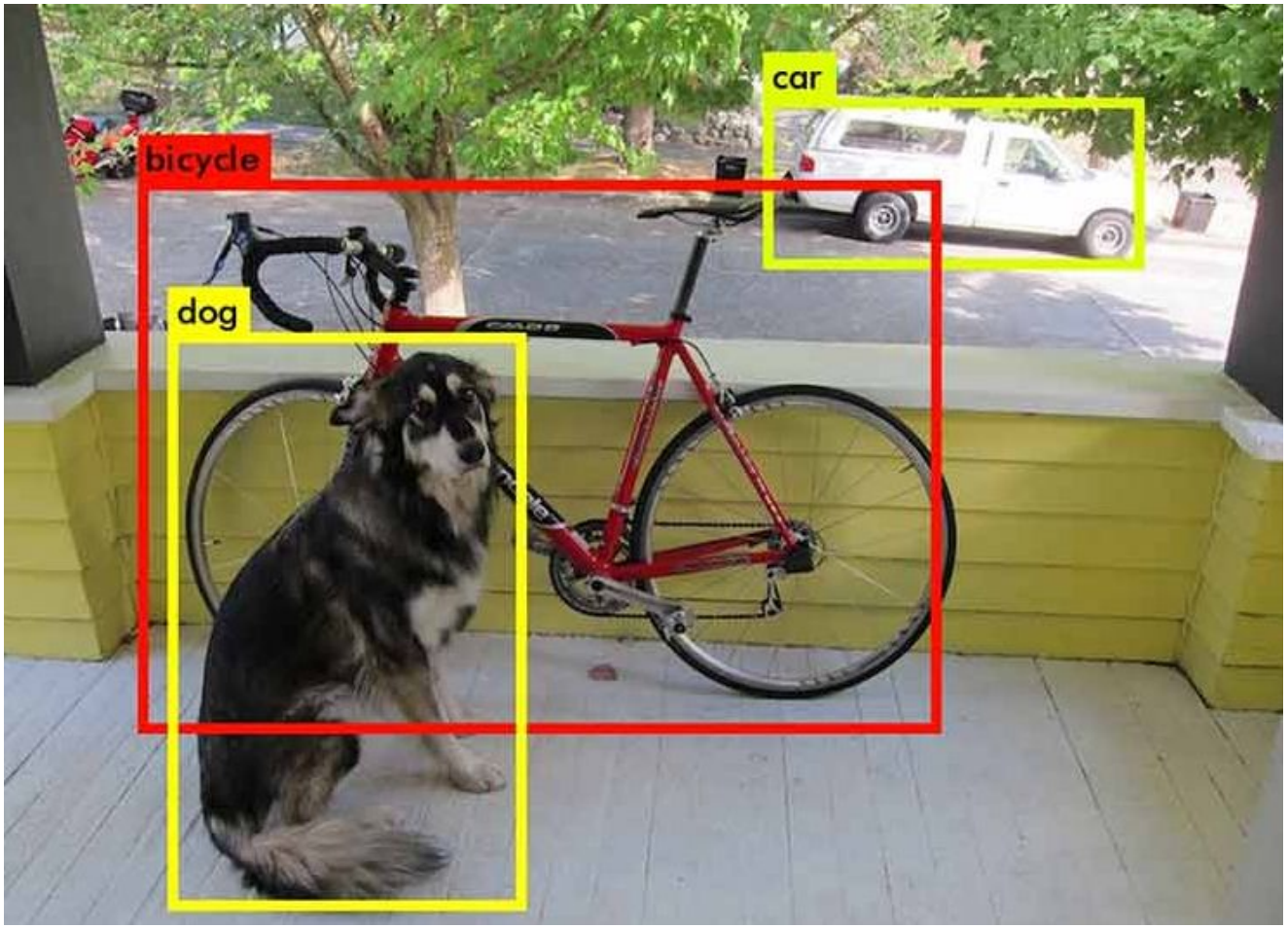


DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
....

Object detection

Object detection: locating and classifying objects within an image.

1 image -> Multiple predictions (class + bounding box)



Object detection

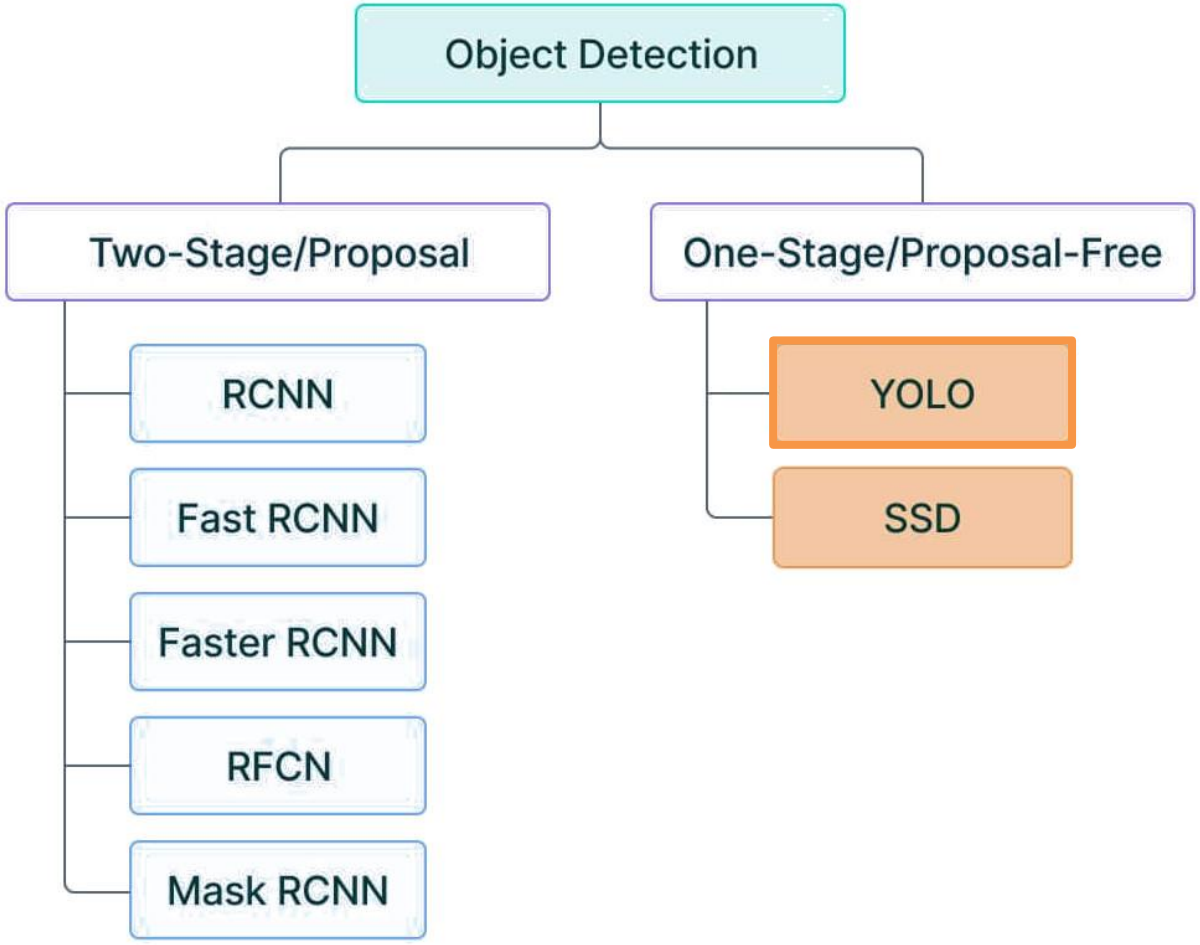
Types of object detectors:

- Two-stage
  - Region proposal: an algorithm (neural network-based or not) that generates potential bounding boxes that may contain objects.
  - Object classification: a neural network that classifies the bounding boxes detected into classes.

- **One-stage**
  - Simultaneous detection and classification: a neural network generates potential bounding boxes and classifies them.

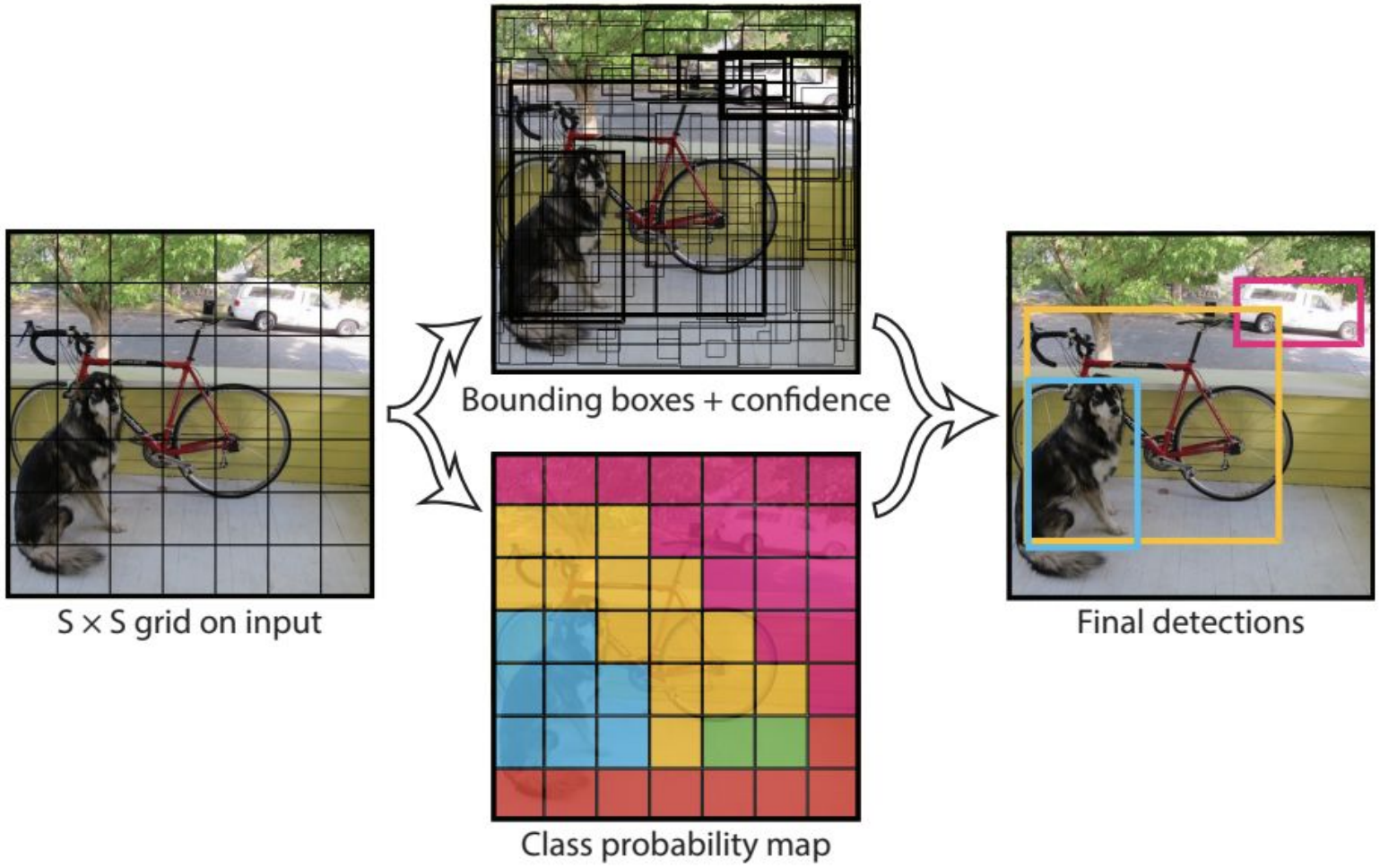
Object detection

One and two stage detectors



YOLO

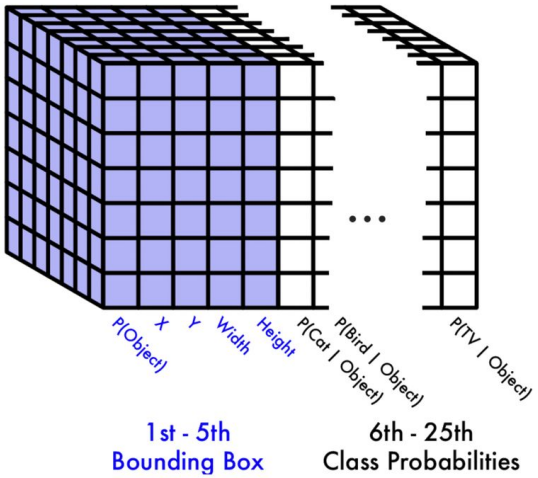
YOLO (You Only Look Once)

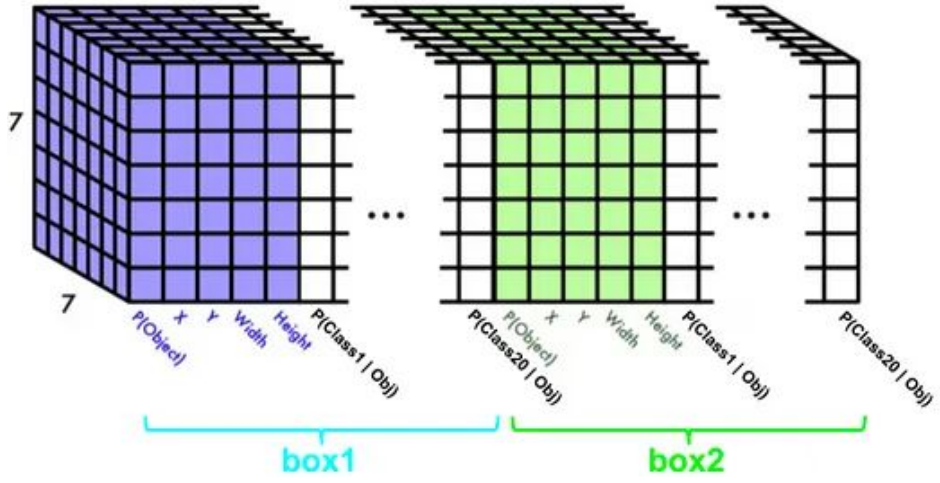


YOLO

YOLO (You Only Look Once)

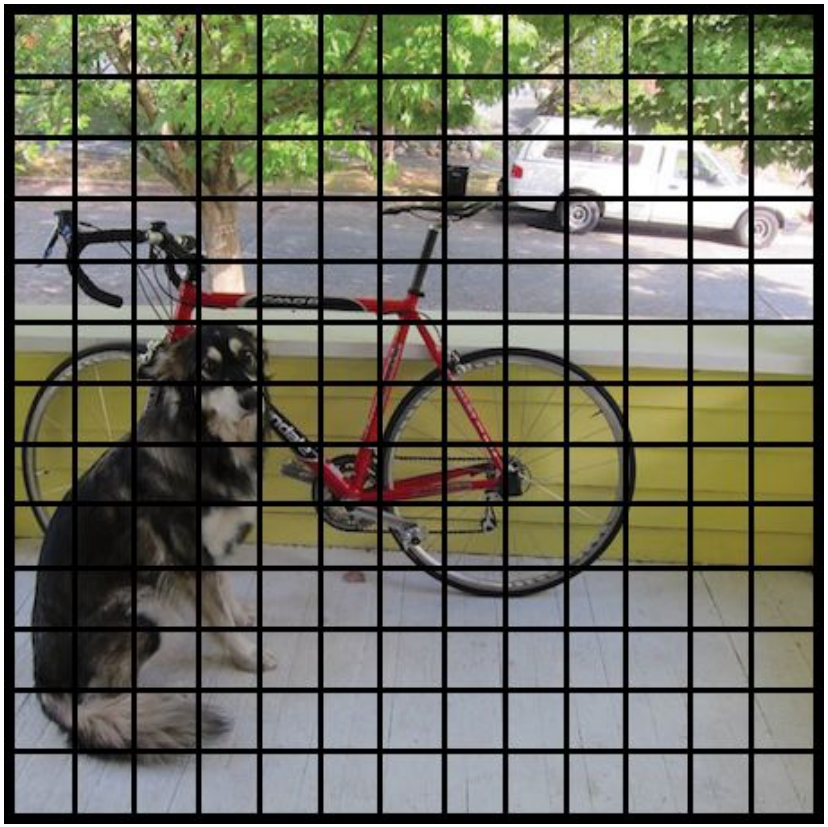
Predictions: 1 anchor vs. 2 anchors





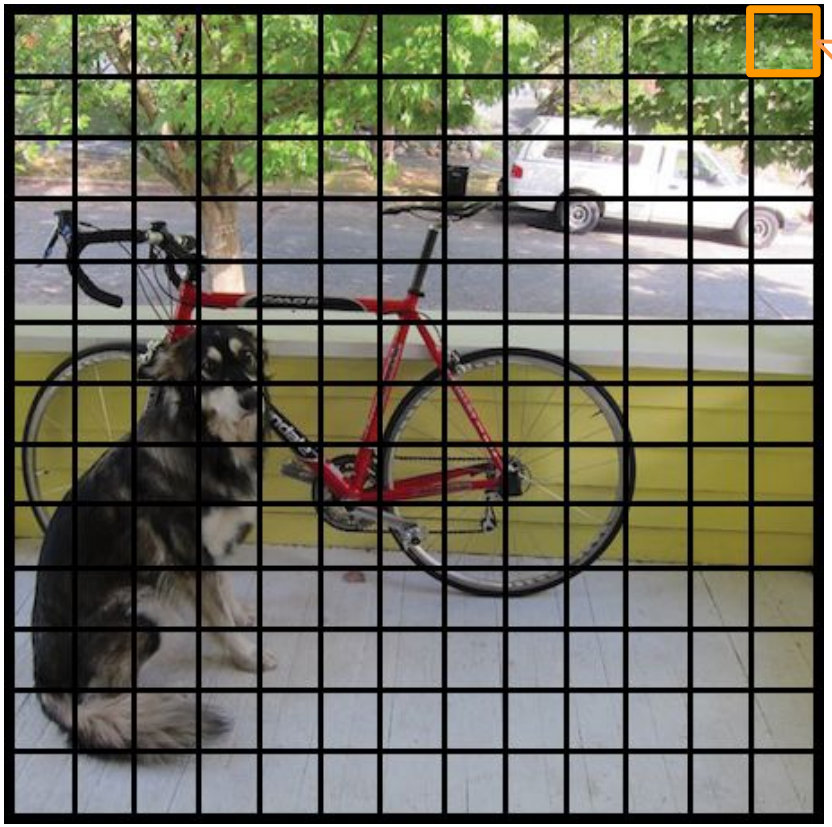
YOLO

1. Dividing the image into NxN grid cells.
   - Each cell in the grid is responsible for detecting and predicting the class of the object that covers (if any).



YOLO

2. For each grid cell, several vectors corresponding to bounding boxes are predicted.



Object probability | Bounding box coordinates | Classification

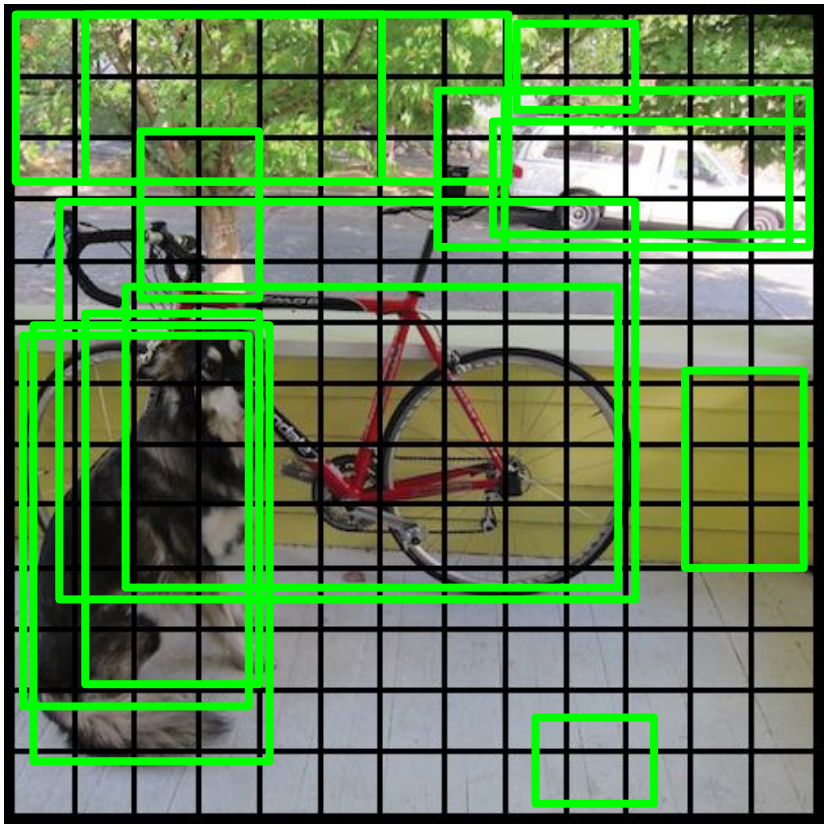
BB1: [pc, bx, by, bw, bh, c1, c2, c3]

BB2: [pc, bx, by, bw, bh, c1, c2, c3]

…

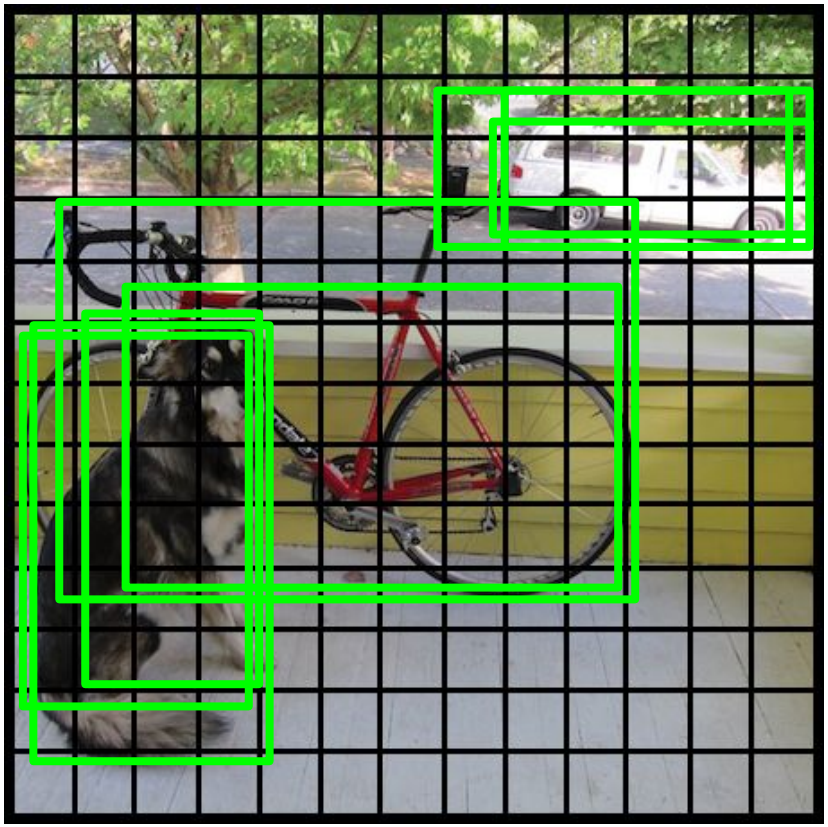
YOLO

3. Discarding bounding boxes with low probability of containing an object.
   - Thresholding (pc).



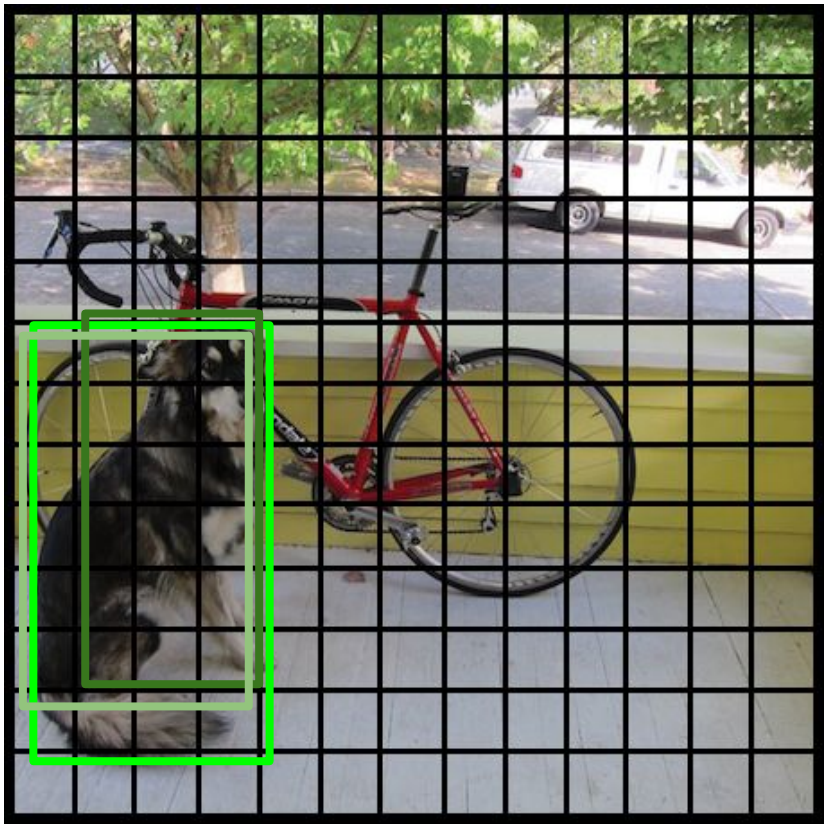
YOLO

3. Discarding bounding boxes with low probability of containing an object.
   - Thresholding (pc).



YOLO

4. Discarding overlapping bounding boxes.
   – Non-maximum suppression (NMS).



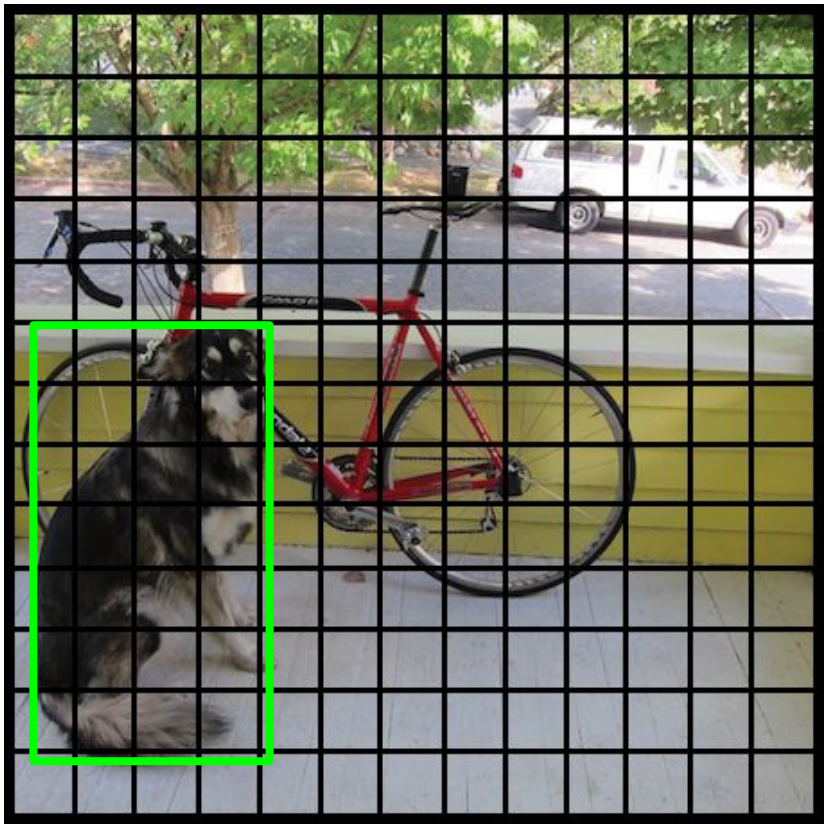
p1: pc = 0.94

p2: pc = 0.86

p3: pc = 0.91

YOLO

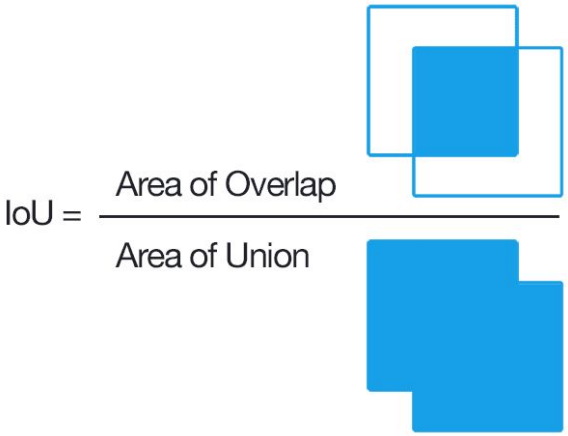
3. Discarding overlapping bounding boxes.
   - Non-maximum suppression (NMS).



p1: pc = 0.94

~~p2: pc = 0.86~~

~~p3: pc = 0.91~~



if IoU(p1,p2) > threshold:

p = argmax(pc(p1), pc(p2))

YOLO

Training YOLO

- 3 loss functions:
  - cls loss: classification loss
  - box loss: bounding box regression
  - dfl loss: distribution focal loss, bounding box regression

- Metrics:
  - For each class:
    - TP, FP, FN, TN (IoU(predictions, labels))
    - Precision
    - Recall

    } Precision-recall curve (at different confidences) → AP (average precision) (area under the curve)

  - Generally:
    - mAP (mean average precision)

      (mean of the average precisions of each class)

Project 2:
Object detection

Marc Masias

Advanced Machine Learning 2024

Màster de Ciència de Dades

Object detection

Models:

- YOLOv8
  - **yolov8n, yolov8s, yolov8m, yolov8l and/or yolov8x**
  - (optional) yolov8-seg
  - (optional) yolov8-pose, yolo-world, …

Framework: ultralytics + colab

PyTorch

Dataset

Custom dataset (whichever you choose*)

roboflow

https://roboflow.com/universe

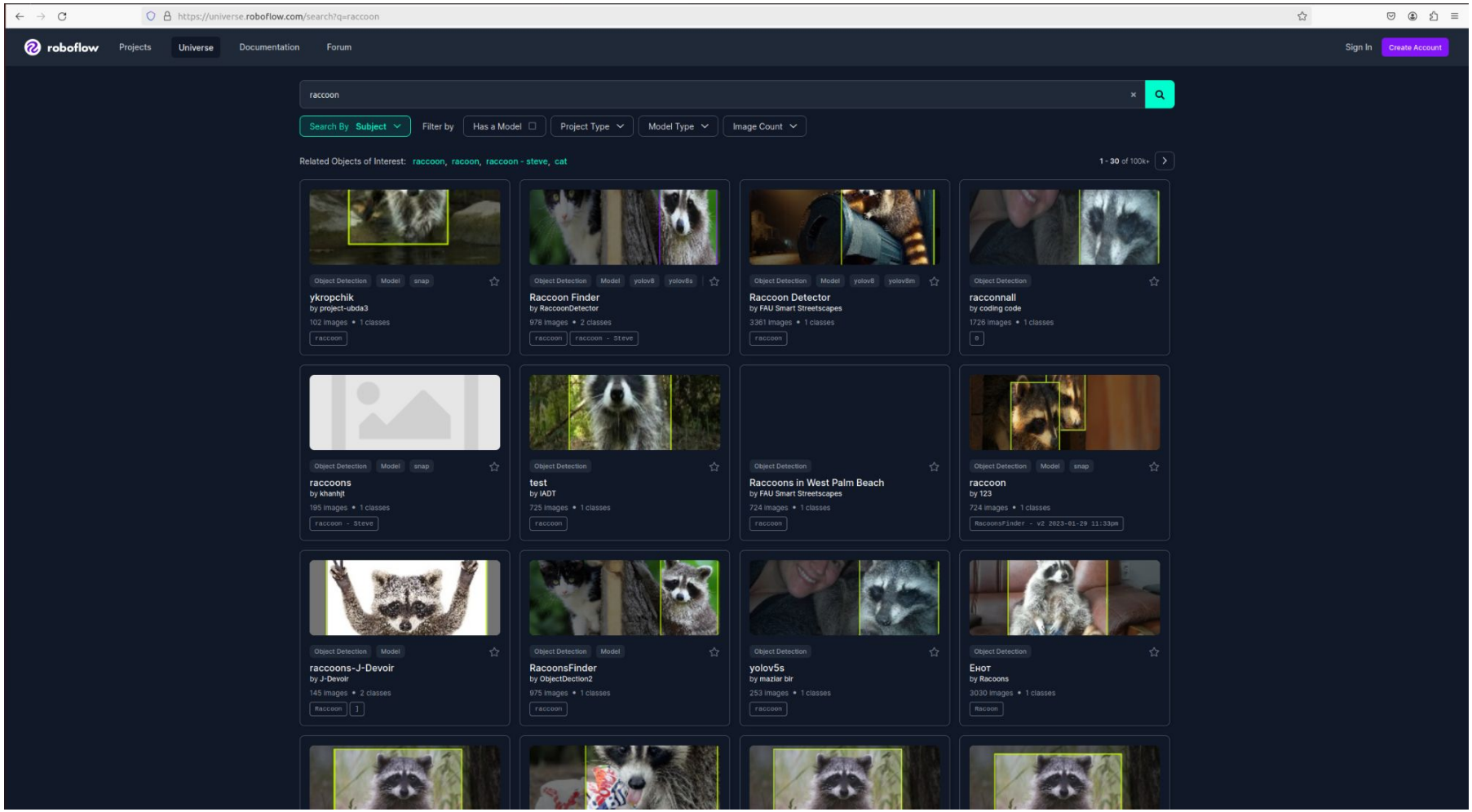
*at least one class not present in the COCO dataset.

person	fire hydrant	elephant	skis	wine glass	broccoli	dining table	toaster
bicycle	stop sign	bear	snowboard	cup	carrot	toilet	sink
car	parking meter	zebra	sports ball	fork	hot dog	tv	refrigerator
motorcycle	bench	giraffe	kite	knife	pizza	laptop	book
airplane	bird	backpack	baseball bat	spoon	donut	mouse	clock
bus	cat	umbrella	baseball glove	bowl	cake	remote	vase
train	dog	handbag	skateboard	banana	chair	keyboard	scissors
truck	horse	tie	surfboard	apple	couch	cell phone	teddy bear
boat	sheep	suitcase	tennis racket	sandwich	potted plant	microwave	hair drier
traffic light	cow	frisbee	bottle	orange	bed	oven	toothbrush

COCO dataset classes. Your custom dataset must include at least one class different from these 80.

Dataset

Custom dataset (whichever you choose*)



Universitat de Girona

Dataset

Custom dataset (whichever you choose*)

Dataset:

- train
  - images
  - labels
- valid
  - images
  - labels
- test
  - images
  - labels
- data.yaml

Image

0.jpg

Label (c, x, y, w, h)

```
0 0.61015625 0.20859375 0.5015625 0.3171875
```

0.txt

```
train: ../train/images
val: ../valid/images
test: ../test/images

nc: 1
names: ['raccoon']

roboflow:
  …
```

data.yaml

Expected tasks

- To apply inference on test images and videos.
  - Get familiarized with the YOLOv8 pretrained models.

- To use transfer learning on a custom dataset to detect at least **one novel class**.
  - Select one or several **custom datasets*** (roboflow).
  - Perform **transfer learning** on pretrained YOLOv8 models. Tweak the different parameters in order to achieve the best possible performance.
  - Test the new models you trained on testing images and videos. Analyze **quantitatively** (confusion matrices, mAP, precision, recall, …) and **qualitatively** the results obtained on validation and testing images.

- Documentation
  - Report

Submission

Submission through proper links on Moodle (La Meva UdG)

- Report (PDF file or notebook)
  - Presentation of the custom dataset.
  - Experimental section and results
    Qualitative/quantitative analysis (**images**, **videos**, plots, tables, discussion, time).
  - Conclusions.

- Code with comments

Evaluation

26% of the final mark

- Report: 90%
- During the lab sessions: 10%

DEADLINE: 21/04/2024 (23:59)