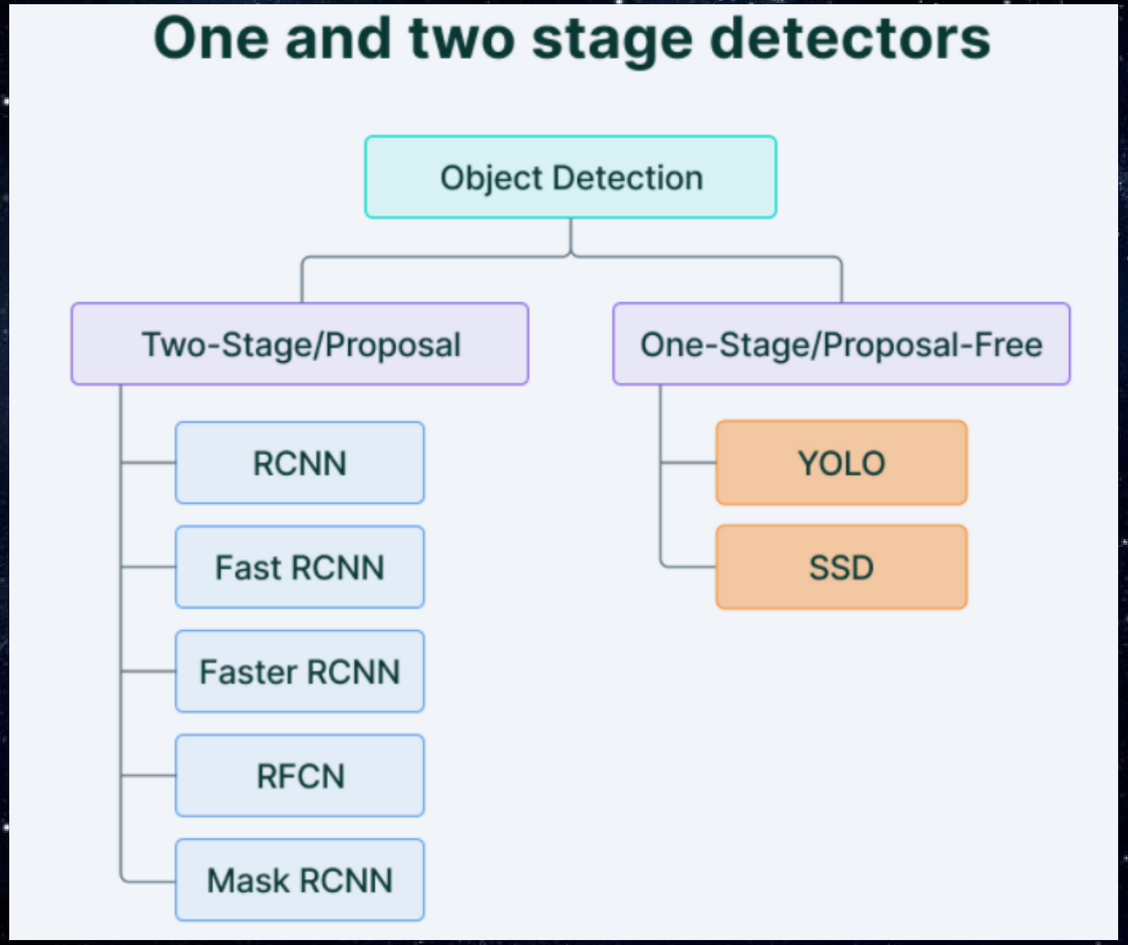
Yolo
You Only Look Once
Real-Time Object Detection

Object detection refers to the detection and localization of objects in an image that belong to a predefined set of classes



Previously people were using :

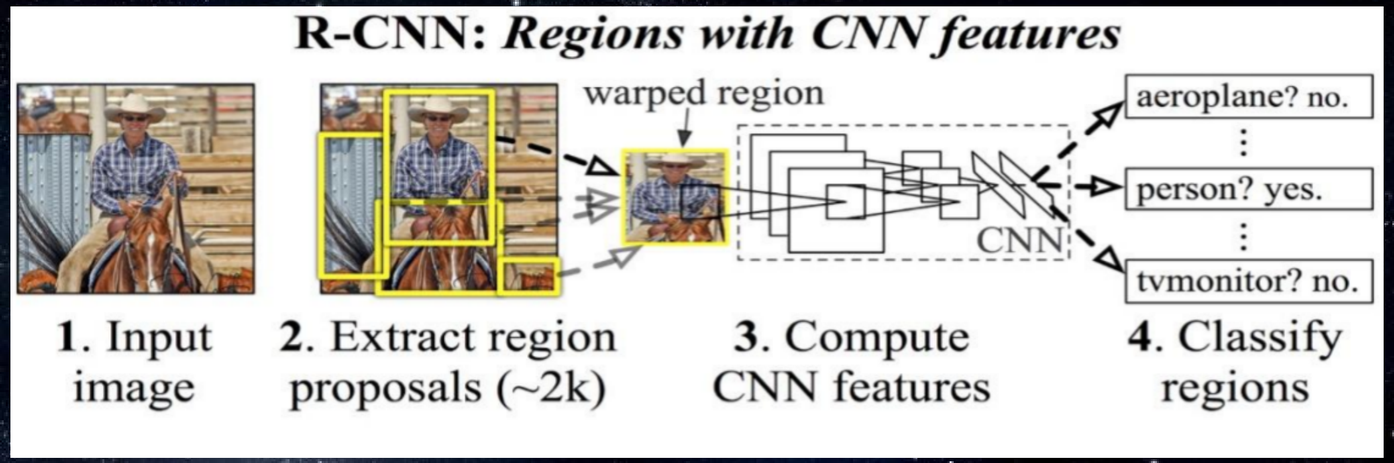
DEFORMABLE PARTS MODELS (DPM) :

- Sliding window

R-CNN

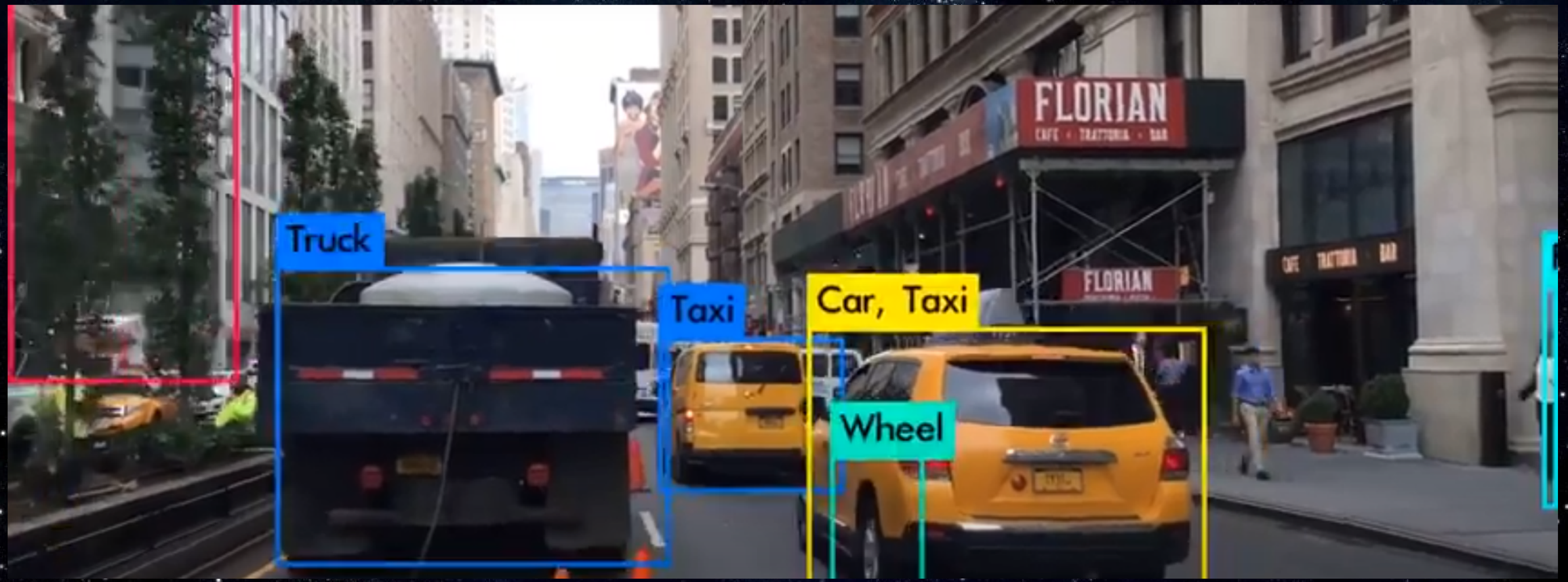
FAST R-CNN

FASTER R-CNN

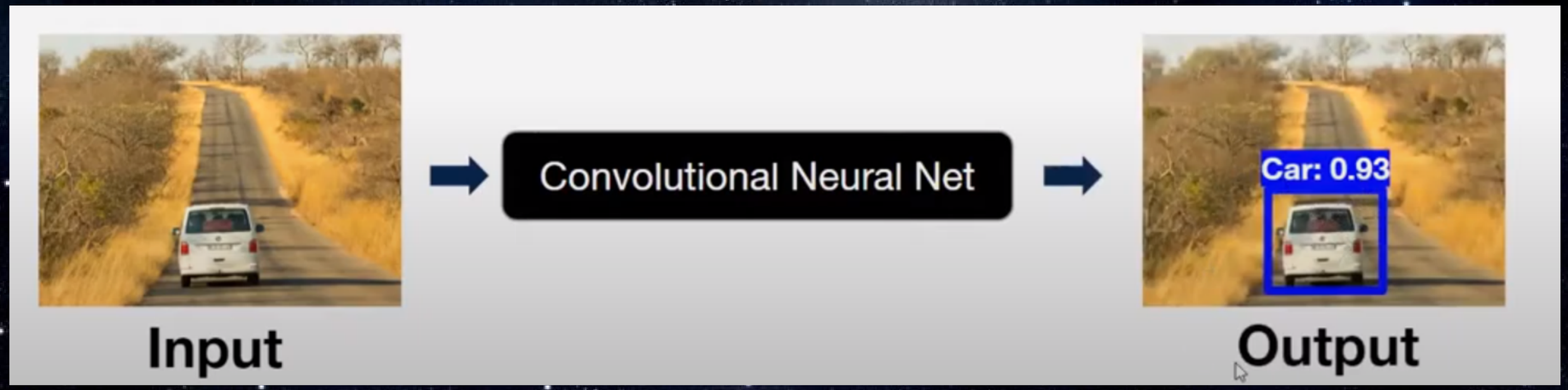


Slow
impossible for real-time detection
Hard to optimize

YOLO - You Only Look Once is an object detection algorithm .
It proposes the use of an end-to-end neural network that makes predictions of bounding boxes and class probabilities all at once.

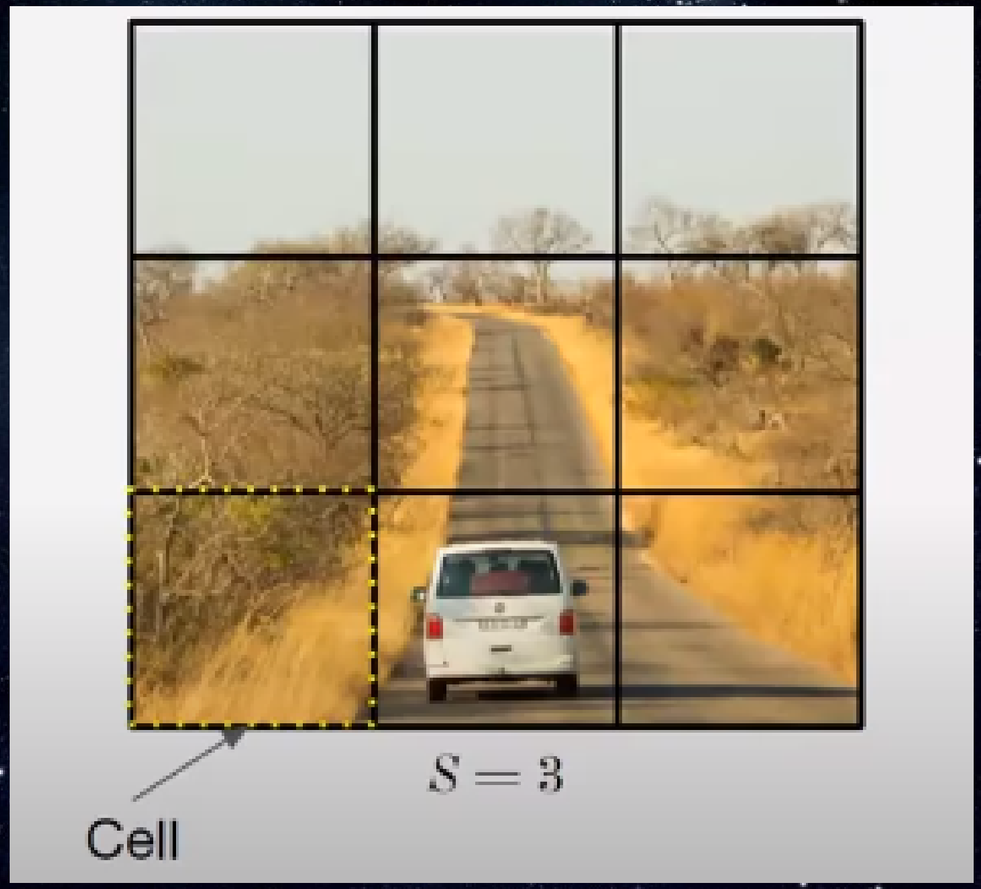


Instead of making predictions on many regions of an image, YOLO passes the entire image at once into a CNN => faster
The CNN that predicts the labels,bounding boxes and cinfidence probabilities for objects in the image.



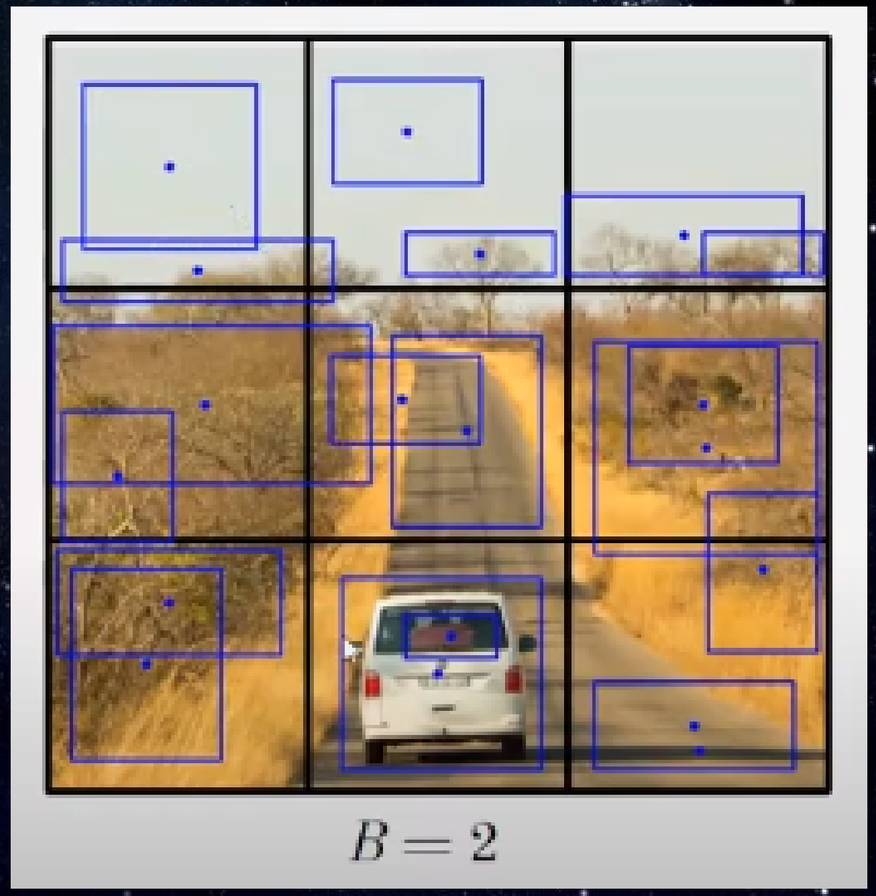
YOLO Steps

1-Devide the image into cells with an S x S grid



YOLO Steps

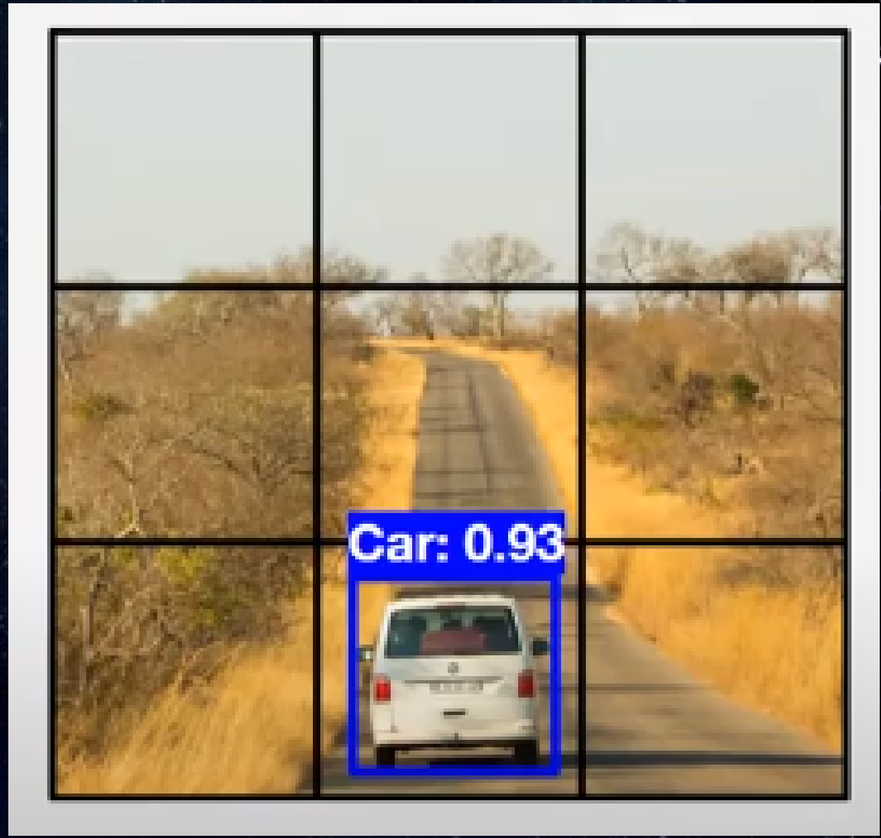
2-Each cell predicts B bounding boxes



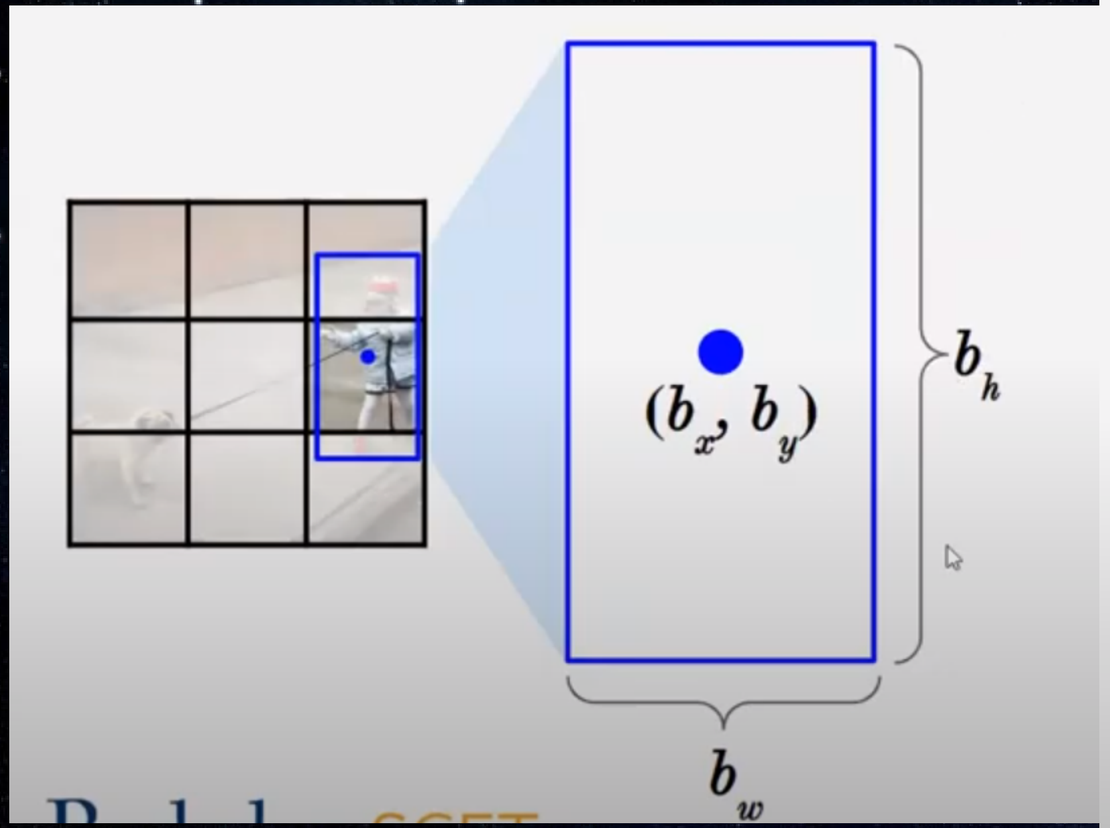
$B = 2$

YOLO Steps

3-Return bounding boxes above confidence threshold



Encoding of bounding boxes



$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \end{bmatrix}$$

- p_c: Probability the bounding box contains an object
- b_x, b_y: Coordinates of the bounding box's center
- b_h, b_w: Width (height) of bounding box as a percent of the cell's width or (height)
- c_1, c_2: Probability the cell contains an object that belongs to class 1 (or 2) given the bounding box contains an object

For each bounding box we need 5 variables:

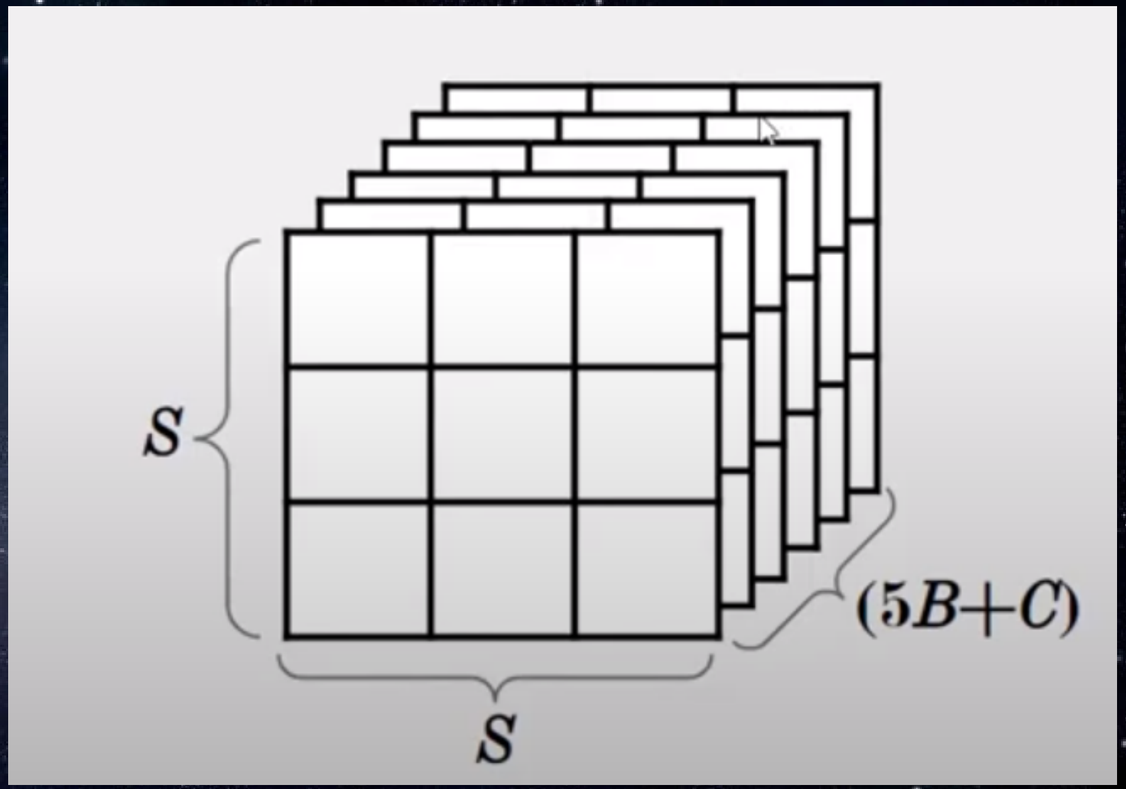
- p : the probability that it contains an object
- bx, by
- bh and bw

Each cell contains B bounding box

- 5B variable
- C variable to predict for each class of the object

The size of the output tensor :

S x S x (5 B + C)



Implementing Yolo

- Pretrained Models

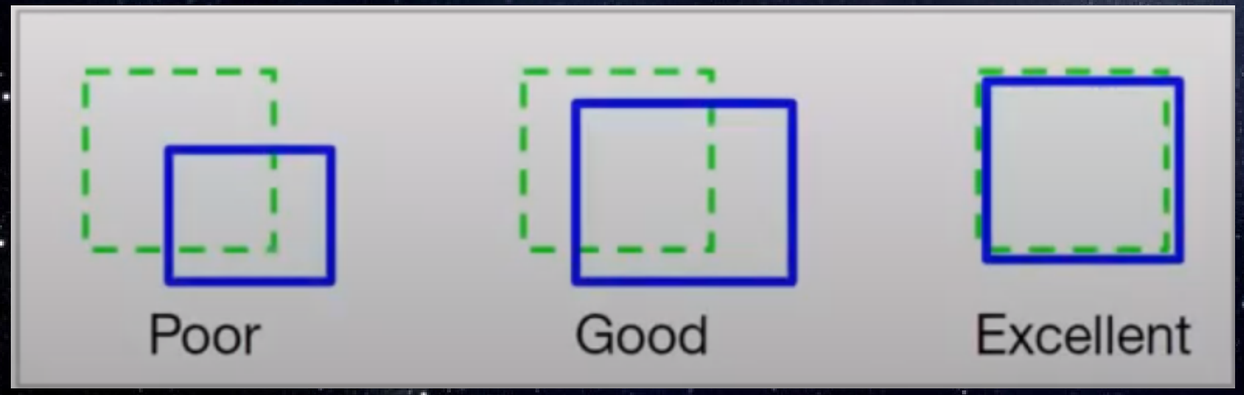
Images with bounding boxes prelabeled are often used to train object detection models

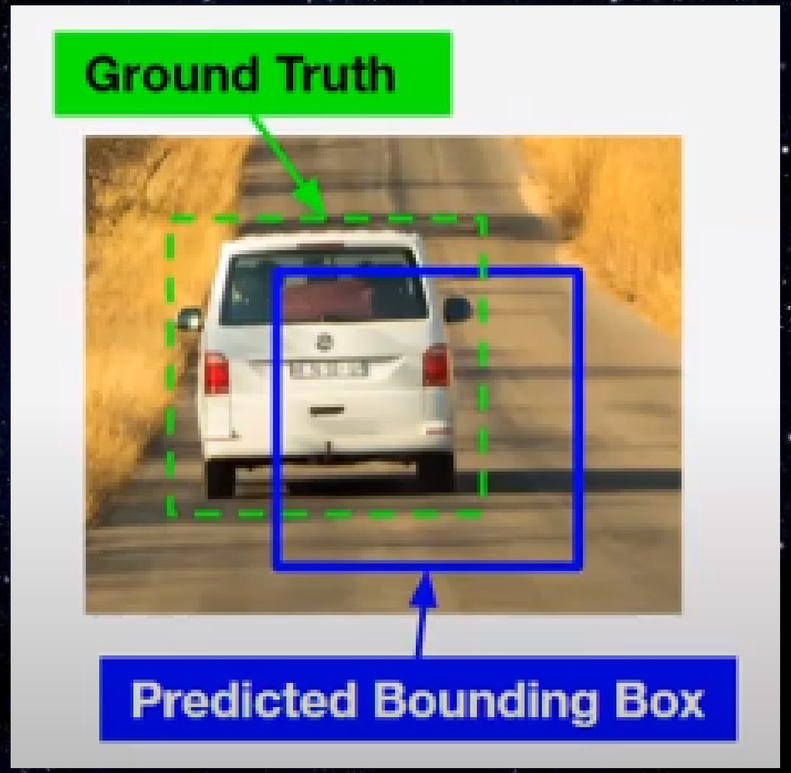
Example: COCO (Common Objects in Context) : a popular computer vision database of images with many labeled objects

- If the oject is in COCO : we download the model
- If the object is not in COCO: We should train a custom model :
  - find images of objects
  - label bounding boxes
  - train the YOLO model

Mesuring the Performance

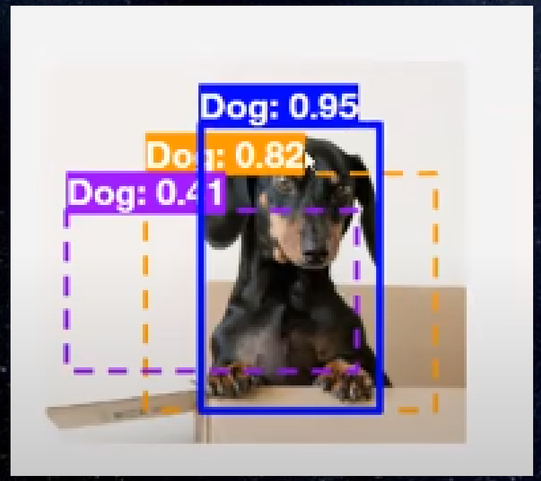
$$\text{Union over Intersection} = \frac{\text{Area of intersection}}{\text{Area of Union}}$$

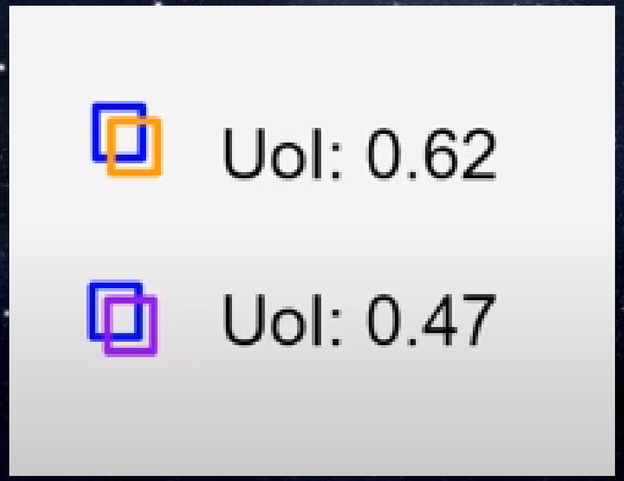


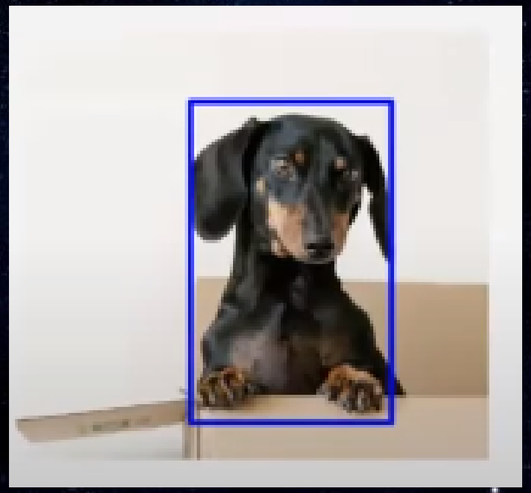


Problem: Double Counting objects

The same object will be detected multiple times.
Non-max suppression solves multiples counting by removing the box with lower confidence probability when the UoI between 2 boxes with the same label is above some threshold







some problems :

- One object is partially/fully covered by several boxes.
- Most boxes has no objects.
- Multi-task training problem: location & class
- Small objects need more accurate location & box size.
- Makes more localization errors

YOLO is not as accurate as RCNN-series models

multi-task problem:

YOLO wins in less background error,

however, loses in localization error.

YOLO is poor for detecting small objects

CNN: training on ImageNet may not generalize well for small objects (classification)

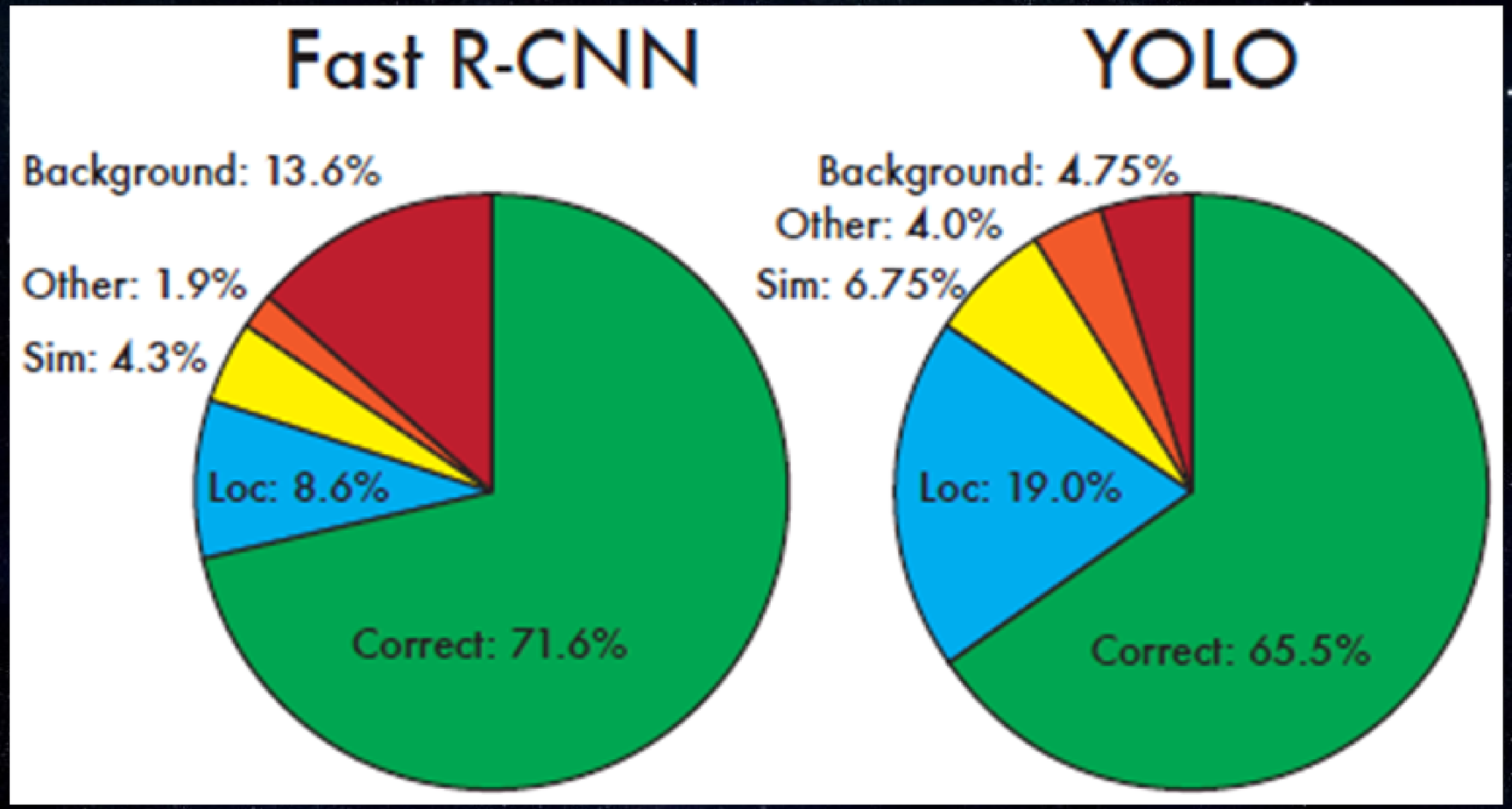
loss function equalizes location weights for small & large objects (localization)

YOLO is not good at crowd objects

non-maximal suppression. See an improvement: Adaptive NMS (arXiv:1904.03629)

YOLO is bad when encountering strange aspect ratio

pre-defined anchors, or anchors learned from data. Go anchor-free (arXiv:1904.01355).


Fast R-CNN
Background: 13.6%
Other: 1.9%
Sim: 4.3%
Loc: 8.6%
Correct: 71.6%
YOLO
Background: 4.75%
Other: 4.0%
Sim: 6.75%
Loc: 19.0%
Correct: 65.5%

strengths

- Fast: 45fps, smaller version 155fps
- End2end training
- Background error is low
- YOLO generalizes well to new domains

	Pascal 2007 mAP	Speed	
DPM v5	33.7	.07 FPS	14 s/img
R-CNN	66.0	.05 FPS	20 s/img
Fast R-CNN	70.0	.5 FPS	2 s/img
Faster R-CNN	73.2	7 FPS	140 ms/img
YOLO	63.4	45 FPS	22 ms/img