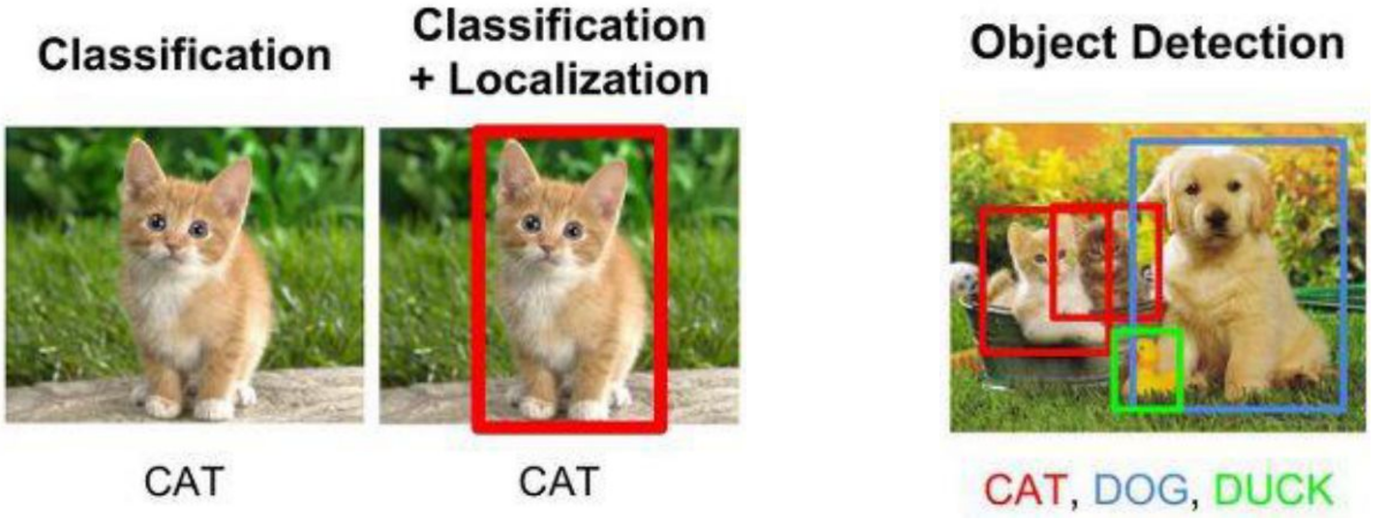
PHY654

Machine learning (ML) in particle physics

Swagata Mukherjee • IIT Kanpur
24th October 2024

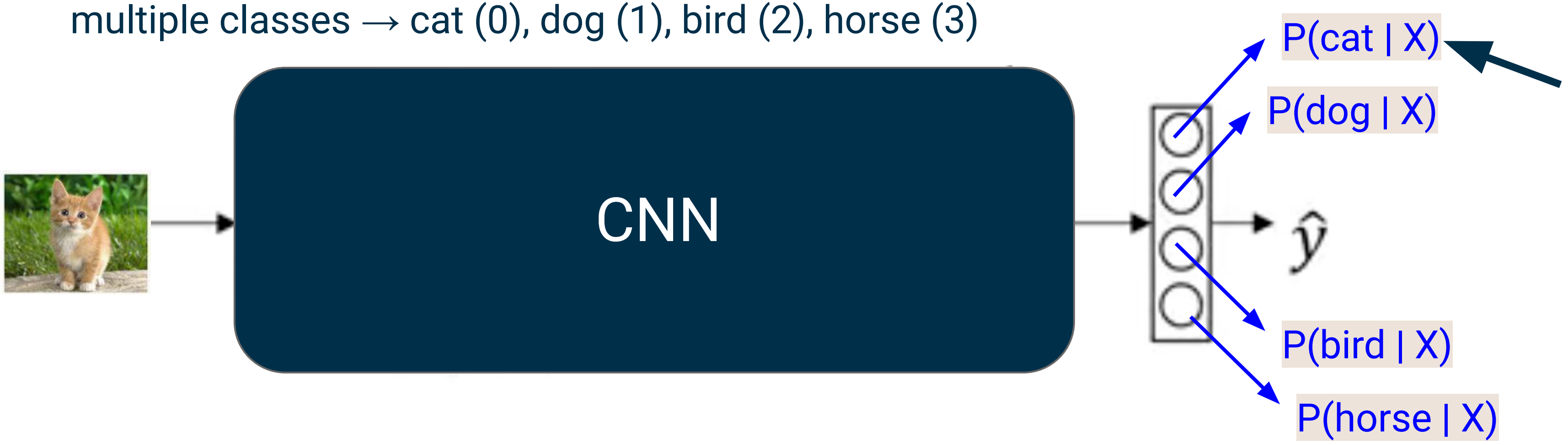
Different types of computer vision problems



Draw bounding box around object

Classification + Localisation

multiple classes → cat (0), dog (1), bird (2), horse (3)



b_x x-axis coordinate of the center of the bounding box

b_y y-axis coordinate of the center of the bounding box

b_h height of the bounding box

b_w width of the bounding box

+Four more numbers (b_x, b_y, b_h, b_w) for the bounding box

When localizing the object the output of the network contains **extra outputs for a defining bounding box**

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

Example 1: If there is an object of class c_2:

$$y = \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

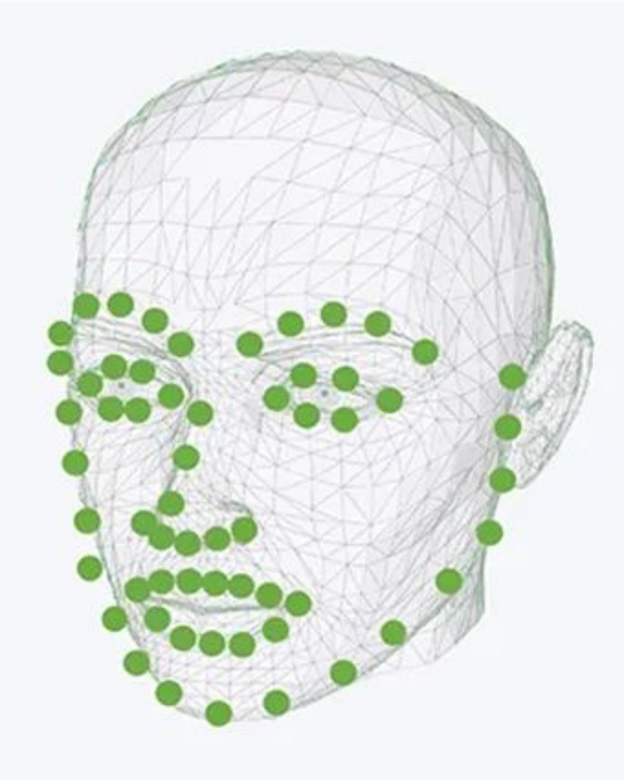
Example 2: If there is no object of any of the defined classes:

$$y = \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}$$

? are not taken into account in the loss function because we do not care these values while no object is detected

Multi-task loss or multi-task learning

Landmark detection

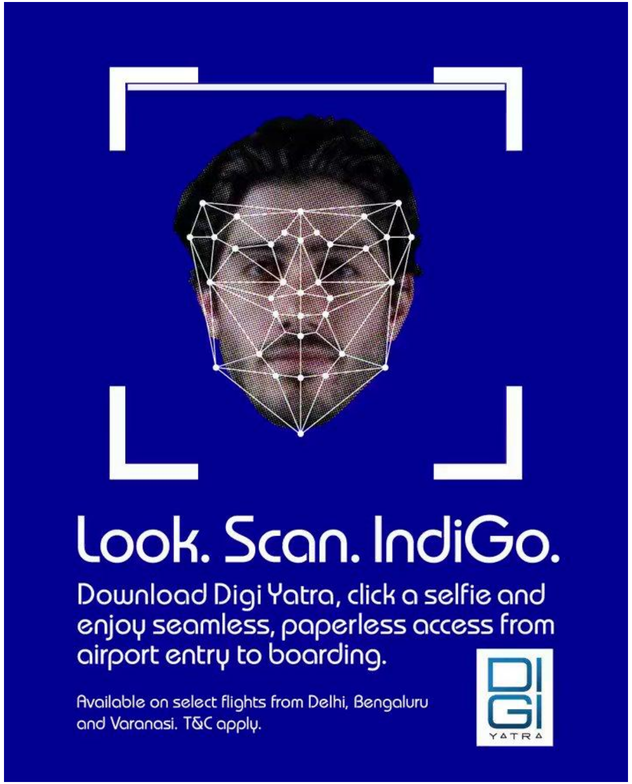


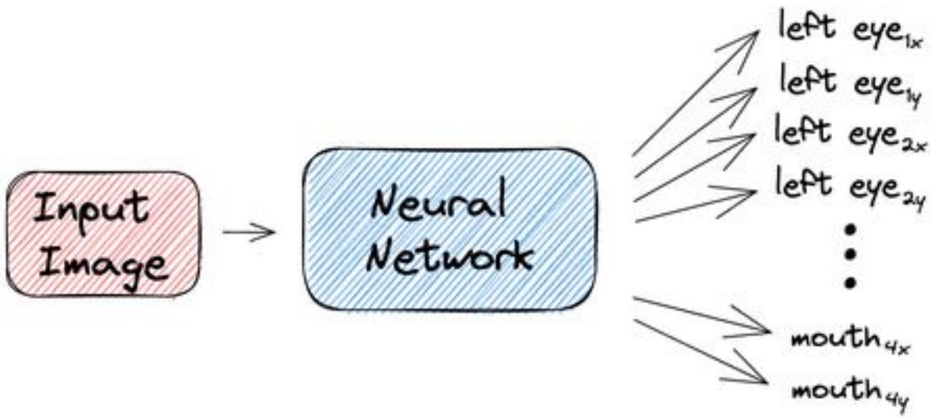
Landmark Detection : computer vision task

Detect and localize specific points or landmarks, say on a face, such as the eyes, nose, mouth, etc.

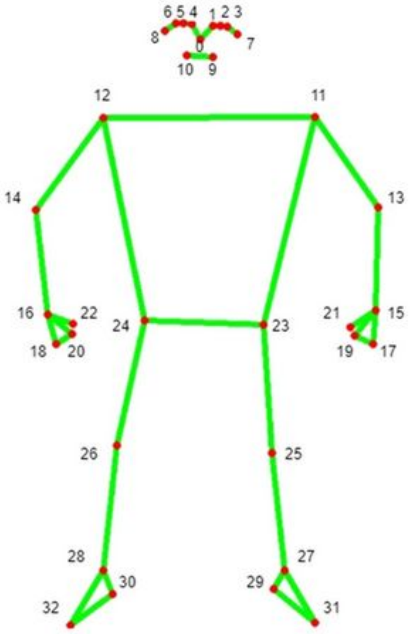
Goal: accurately identify these landmarks in images or videos in real-time and use them for face recognition, facial expression analysis, and head pose estimation.

Landmark detection



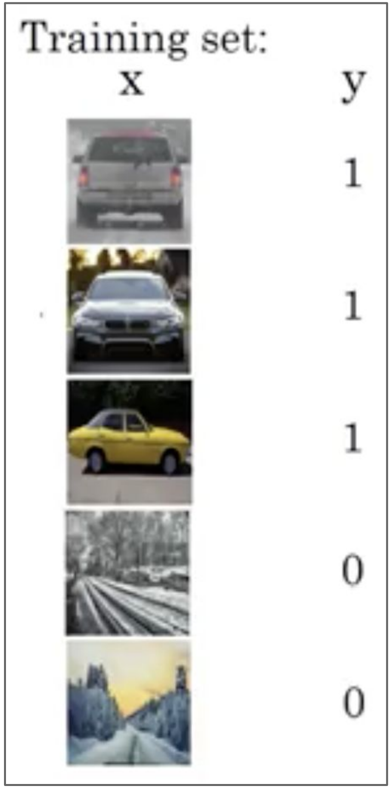


Landmark detection is not only about face



0. nose
1. left eye inner
2. left eye
3. left eye outer
4. right eye inner
5. right eye
6. right eye outer
7. left ear
8. right ear
9. mouth left
10. mouth right
11. left shoulder
12. right shoulder
13. left elbow
14. right elbow
15. left wrist
16. right wrist
17. left pinky
18. right pinky
19. left index
20. right index
21. left thumb
22. right thumb
23. left hip
24. right hip
25. left knee
26. right knee
27. left ankle
28. right ankle
29. left heel
30. right heel
31. left foot index
32. right foot index

Object detection example – car detection



Train CNN with cropped images of car.

Then do a sliding window search.

Choose stride meaningfully.

The CNN should detect the 2 cars.

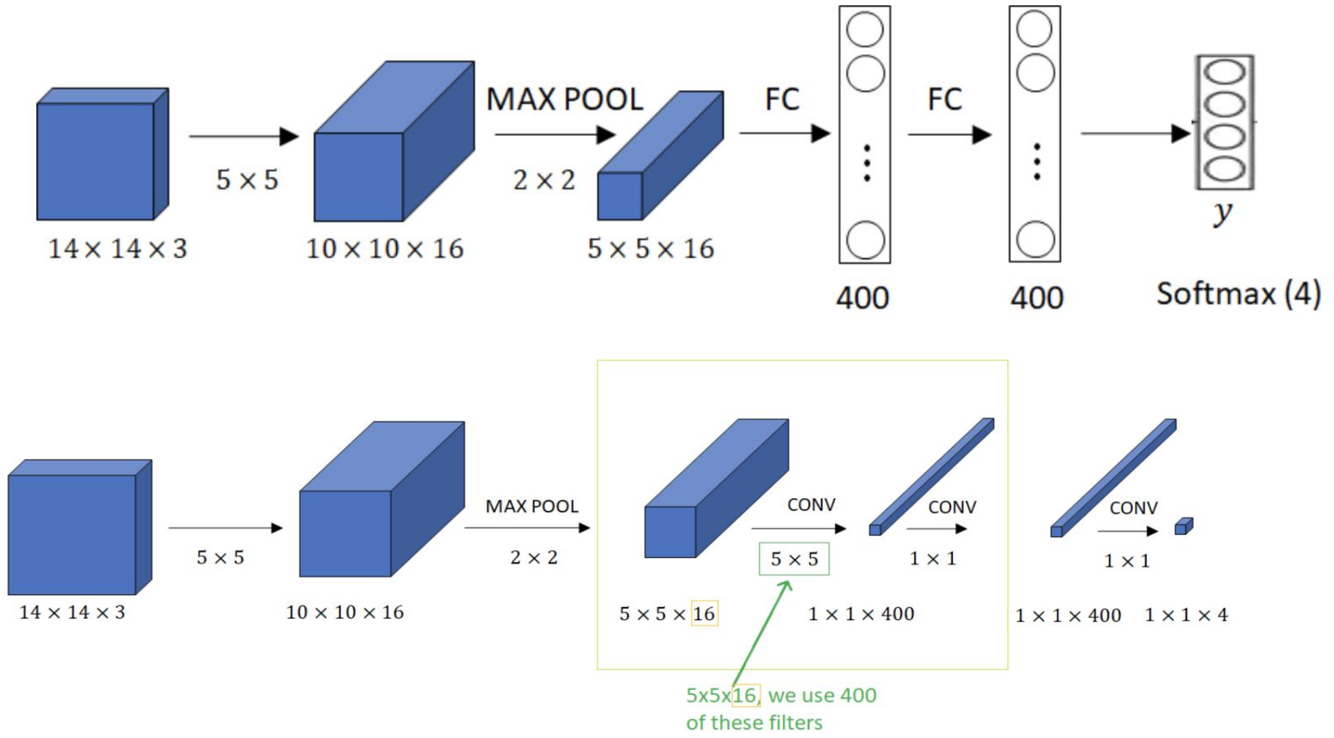
There are drawbacks.
Best window size not known. Computationally expensive.

Object detection example – car detection

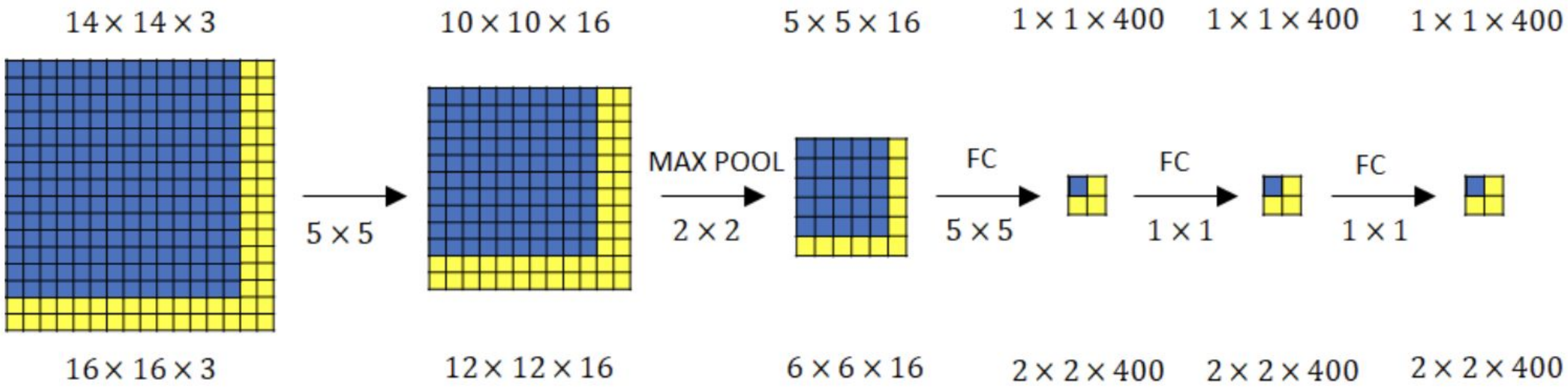
Do we really need this sequential method?

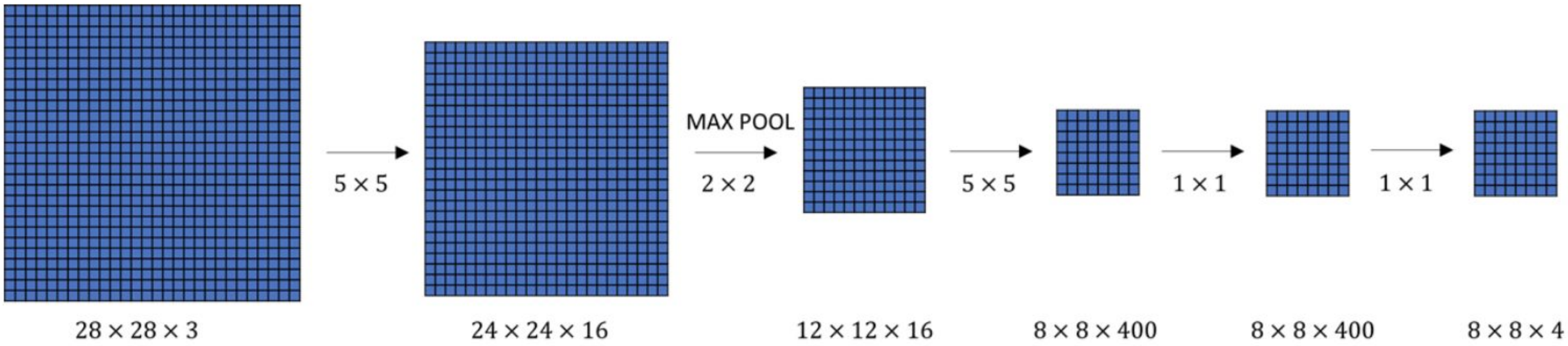
Can’t we do this in one-shot?

Can we turn FC layer into Conv layer?

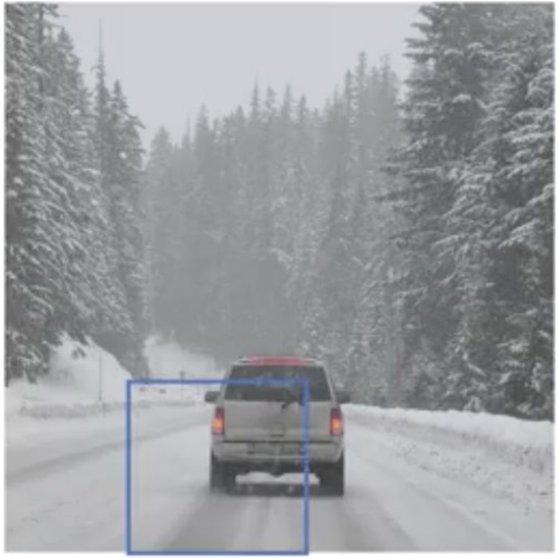


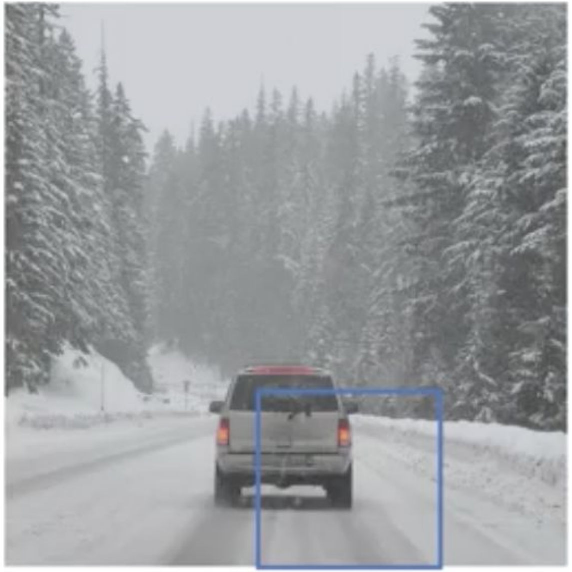
One shot, and gives same answer as sliding window.





One shot, and gives same answer as sliding window.
But what about the bounding box?





Sliding window → depending on window-size and stride, it may happen that the car does not fully fit in any window.

Also, bounding box may be rectangle and not square.

How to get accurate bounding box?

YOLO algorithm

You Only Look Once (YOLO)

Place a grid on the image. Here 3x3

Apply classification+localisation.

For 3 classes, we have 8 numbers as output for 1 grid.

So, output dimension is 3x3x8.

Object assignment to grid-cell is based on object mid-point.

Bounding box can span >1 grid-cells.

Intersection over union

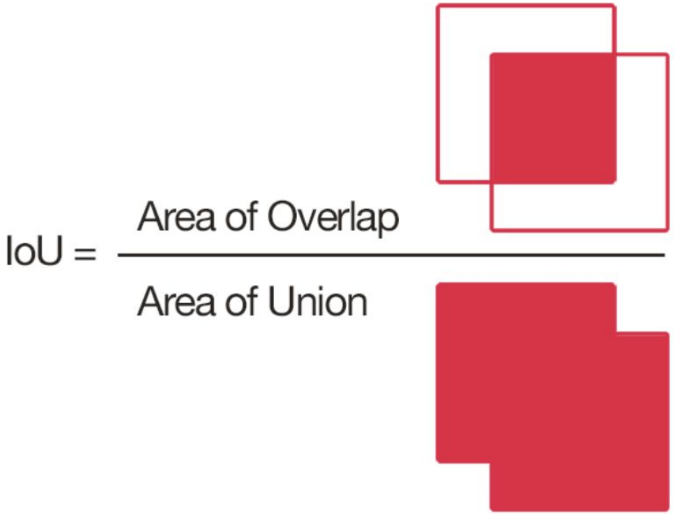
How to evaluate the performance of the object localizer?

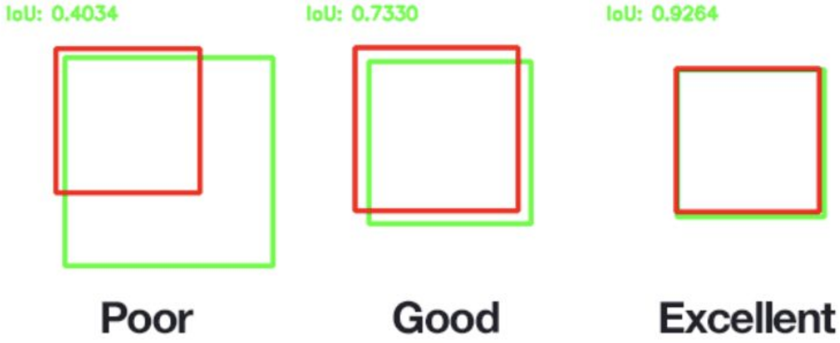
$$IOU = \frac{Intersection\ area}{Union\ area}$$

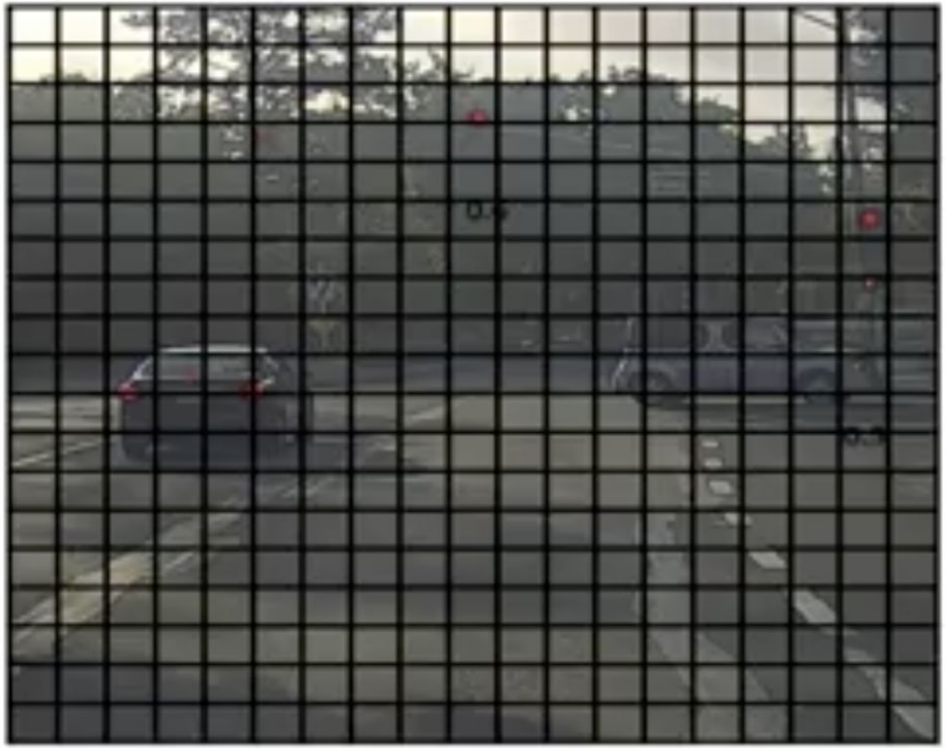
Assume correct if IOU > 0.5

May use higher threshold (0.6 or 0.7 or even higher) depending on problem.

IOU = 1 for a perfect bounding box.


IoU =
Area of Overlap
Area of Union

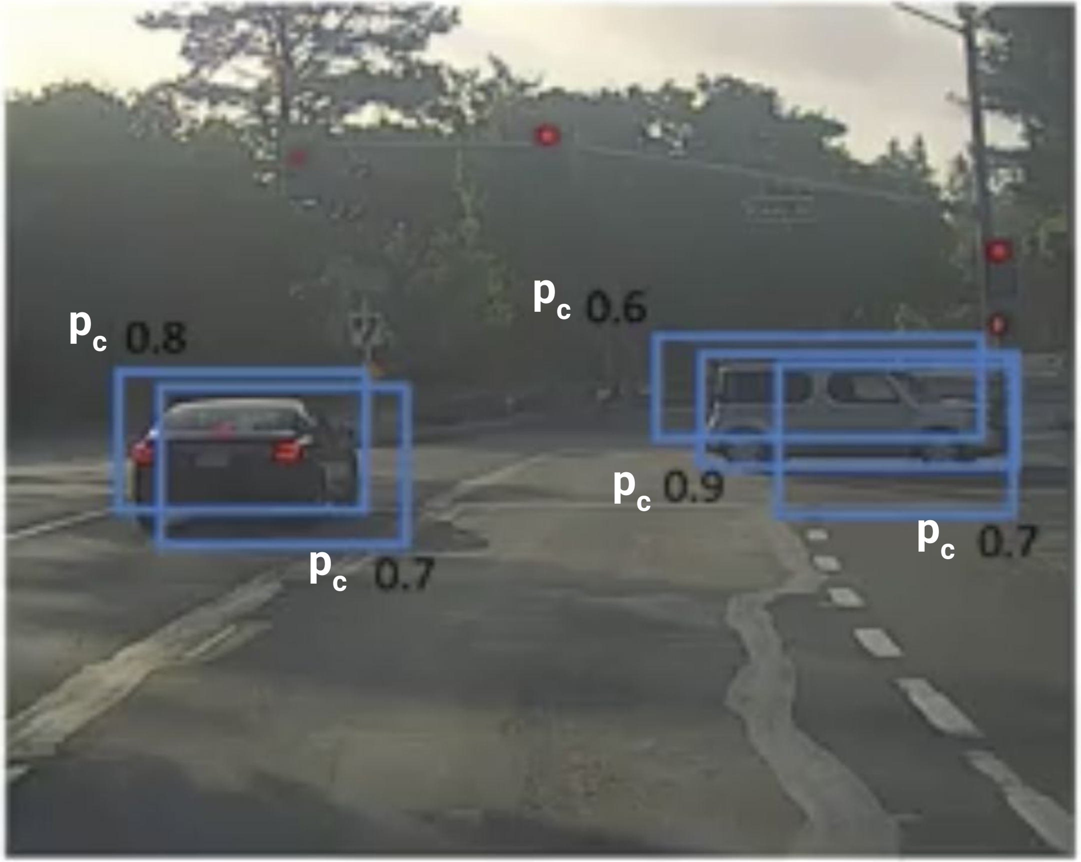

IoU: 0.4034
IoU: 0.7330
IoU: 0.9264
Poor
Good
Excellent



Object assignment to grid-cell is based on object mid-point.

But sometimes, multiple grids flag the same object.

Multiple detection of same object.
Way out? **Non-max suppression.**



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

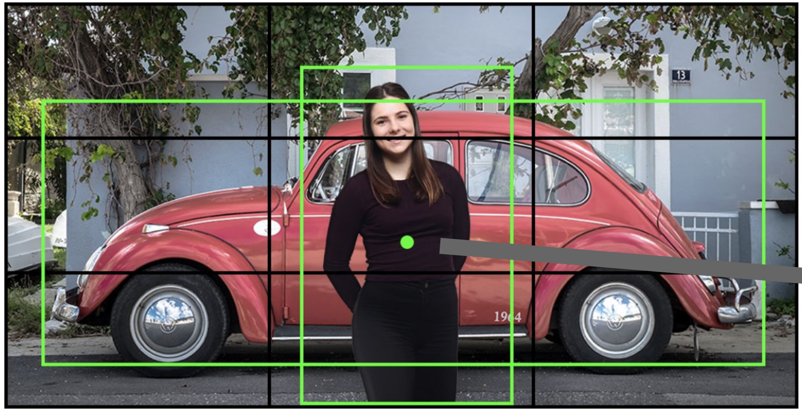
Select bounding box with highest p_c.

Find other bounding boxes that highly overlap with that.

Reject them.

Non-max suppression

Overlapping objects

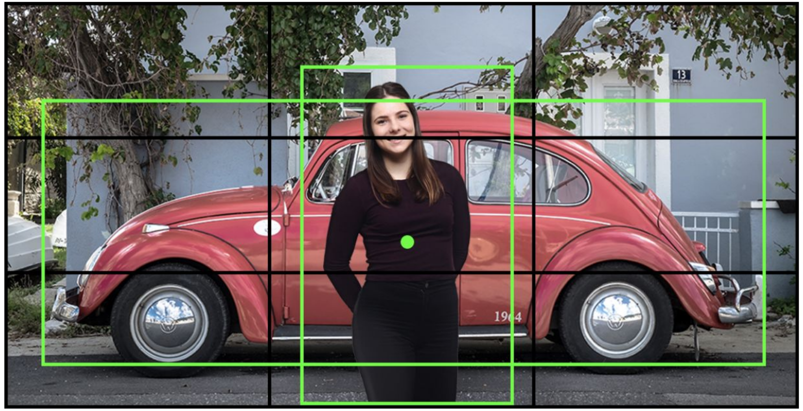


Say we have 3 classes,
Car, human and cycle.

Which bounding box and which class will it detect?

$$y = \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

Define a few anchor boxes with predefined shapes associated with different classes of objects that can occur in the same grid cell

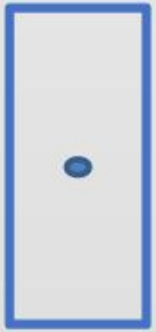


With two anchor boxes:

Each object in training image is assigned to grid cell that contains object's midpoint and anchor box for the grid cell with highest IoU.

Example:

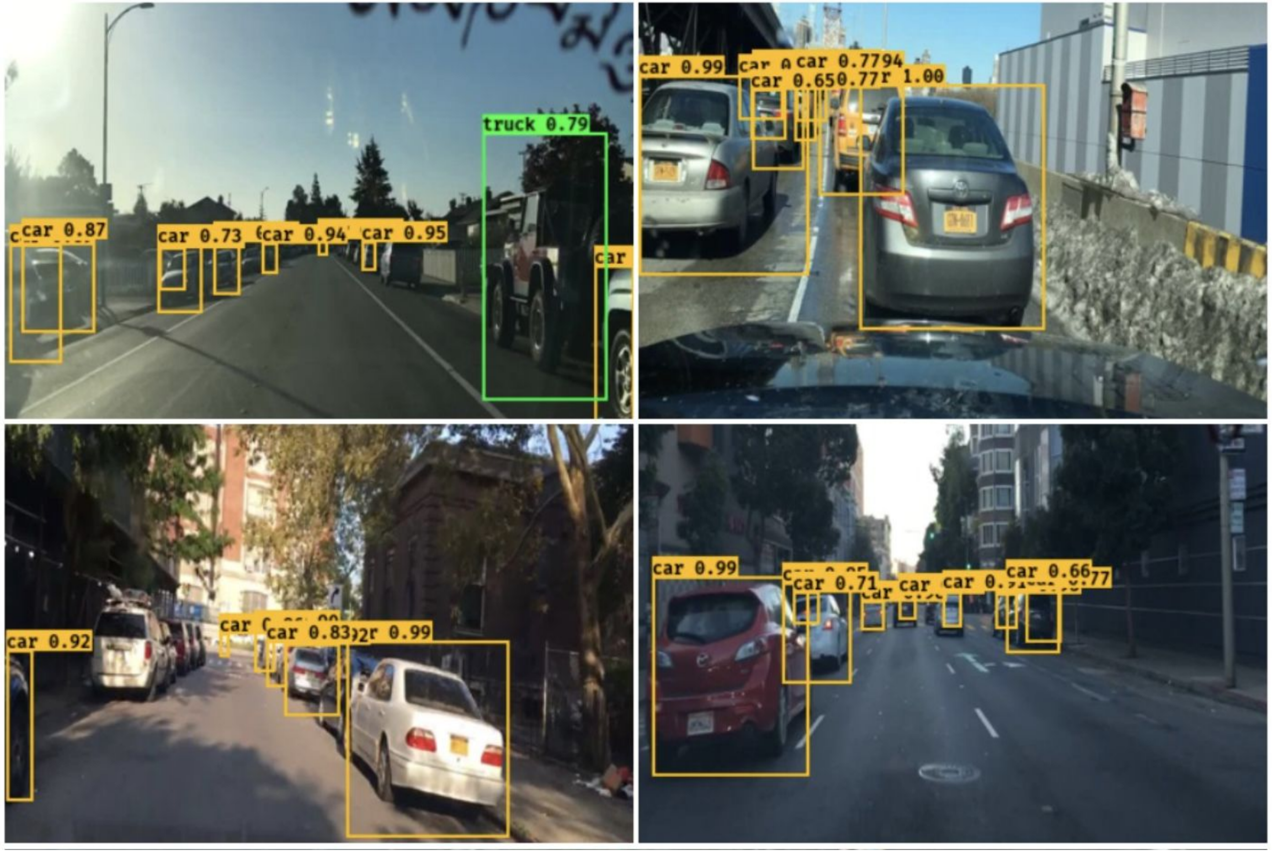
Anchor box 1 (A1):



Anchor box 2 (A2):

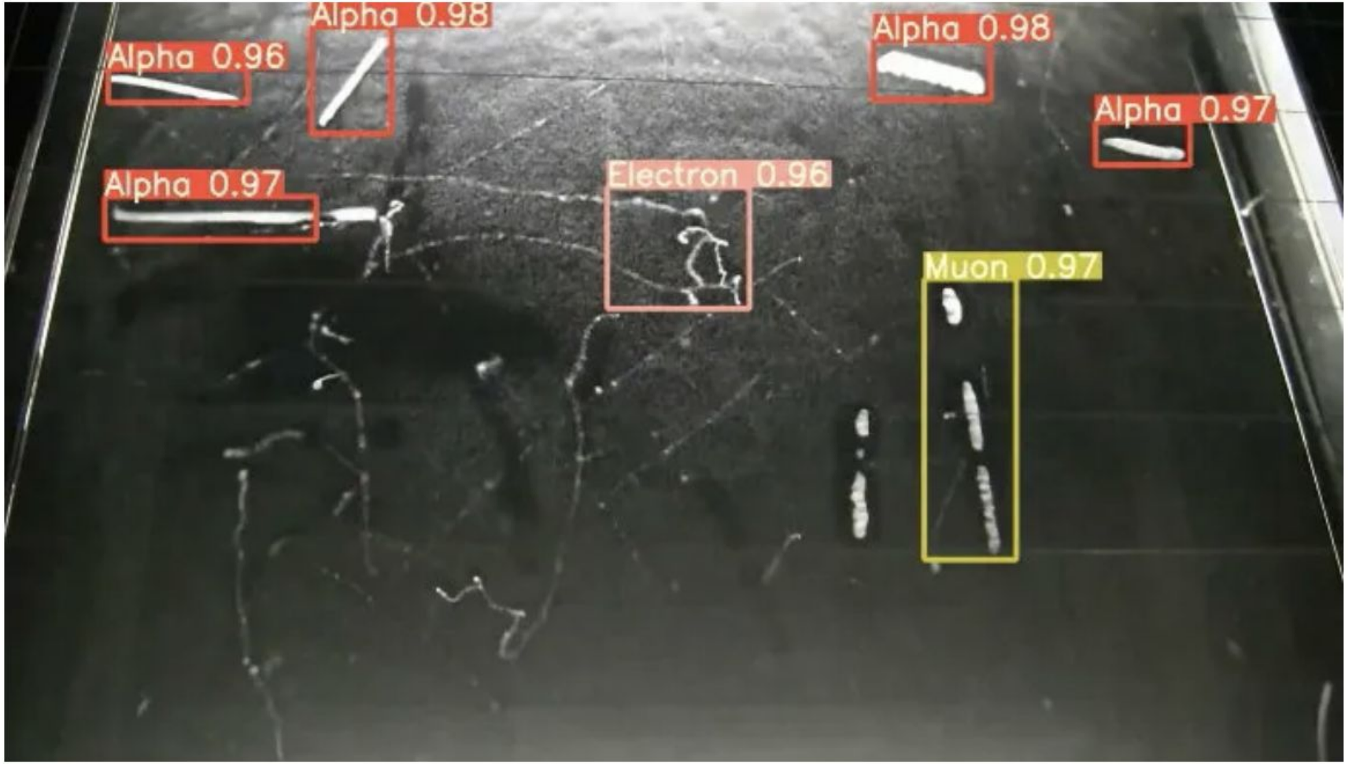
$$y = \begin{bmatrix} p_c^{A1} \\ b_x^{A1} \\ b_y^{A1} \\ b_h^{A1} \\ b_w^{A1} \\ c_1^{A1} \\ c_2^{A1} \\ \vdots \\ c_K^{A1} \\ p_c^{A2} \\ b_x^{A2} \\ b_y^{A2} \\ b_h^{A2} \\ b_w^{A2} \\ c_1^{A2} \\ c_2^{A2} \\ \vdots \\ c_K^{A2} \end{bmatrix}$$

Self-driving car

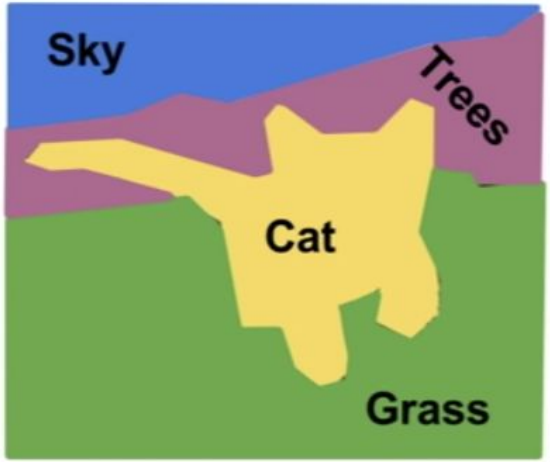


Images from Cloud Chambers and YOLO algorithm

Cloud Chambers → one of the oldest particle detectors.



Semantic segmentation



R-CNN

Region proposal.