⑥ Object detection

↳ End goal: To predict 'bounding box' with object probability P_c and class probability 'c'

↳ Step 1: is to encode the image using CNN.

Ex: $608 \times 608 \times 3 \xrightarrow{CNN} \underbrace{19 \times 19}_{\text{Grid}} \times$ no of anchor boxes $\times$ no of class Probs

Ex: Say we have $608 \times 608 \times 3$ image, we have 5 anchor boxes, 80 classes

$\therefore 608 \times 608 \times 3 \xrightarrow{CNN} 19 \times 19 \times 5 \times 85$

here 80 → classes but each box has 5 params: b_x, b_y, b_h, b_w, P_c

$[\underbrace{P_c, b_x, b_y, b_w, b_h}_{5} \underbrace{C}_{80}] \to [85]$

↳ Now, $19 \times 19 \times 5 \times 85 \longrightarrow [19 \times 19 \times 5 \times 1, 19 \times 19 \times 4 \times 80]$

P_c — Probability that an object is present in each the box

$[19 \times 19 \times 5 \times 80]$ → each of the four boxes have the class probs now.

$[b_x, b_y, b_h, b_w]$

↳

Step 1 :- Compute box scores

↳ we have 19x19 grid and for each grid we have 5 boxes and for each box we have 80 class probability, object present probability ↳

(19x19x5x80) — each grid, each box, C ↳ A

(19x19x5x1) — each grid, each box, p_c ↳ B

⇒ box scores B_x = A x B = 19x19x5x80

Now B_x contains the probability that an object is present with class probability for each of the five boxes in each grid cell

→ For each grid we have 5 boxes and hence 5 box scores. Now for each boxes in each grid, find the maximum box score.

Ex:-

Say in the third grid we have 5 boxes

★ = [0.7 (1), 0.1 (2), -0.2 (3), 0.3 (4), 0.5 (5)]

max = 0.7 = box_1 = [b_x, b_y, b_h, b_w, p_c,

Now we check if the max box score in that grid is > threshold, if yes, we keep that box, if no we delete that box.

We repeat this step until we have kept/removed boxes from each grid cell.

[for each box there are 80 class (we find max value of class) ∴ box score will have 5 value (each for each box) in each grid cell]

↳ how do we know (b_x, b_y, b_h, b_w)?

Ex:- Say we are checking for boxes in grid 1 (out of 19x19 grid).

box 1 [P_c, b_x, b_y, b_h, b_w, c_1, c_2 . . . . . . c_{80}]

box 2 [P_{c_2}, b_{x_2}, b_{y_2}, ... c_{80}]

box 3 : [. . . . . .]

box 4 : [. . . . . .]

box 5 : [. . . . . .]

Now box_1 score $= P_c \cdot C_3 = 0.7$ (assume $P_c C_3$ is max)

box_2 score $= P_{c_2} C_{7_2} = 0.5$

box_3 score $= 0.3$, box_4 score $= 0.8$, box_5 score $= 0.5$

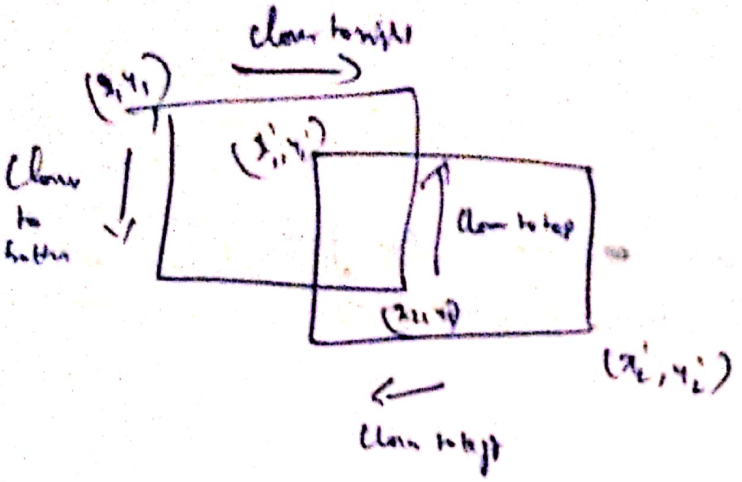
Now, say threshold is 0.6, therefore box-1, box-4 will be kept.

Therefore for grid 1 box 1, box 4 will be kept. Repeat for

↳ After removing / filtering boxes, apply non-max suppression to further remove more or overlapping boxes.

↳

IoU



↳ Take the best measure box, compute IoU with all other filtered boxes (filtered by above discussed previously).

Summary :-

↳ Say You have 100 images with three classes [Person, bat, ball] (Cricket image).

Step 1 :- Build a CNN to encode the image into $19 \times 19 \times 5 \times 8$

- (Bounding box, $(b_x, b_y, b_h, b_w, P_c$, Person, bat, ball)

↳ how?

Input image $600 \times 600 \times 8$ $\xrightarrow[\text{architecture}]{\text{CNN}}$ $19 \times 19 \times 5 \times 8$

You have the function to compute how different predicted boxes & actual boxes (drawn)

Step 2 : once CNN model is trained, (built, the model) in right can be used for other detection (on videos) we use this & then the encoded image is passed to Yolo ( ) .

Steps : Yolo uses the algorithm discussed before.