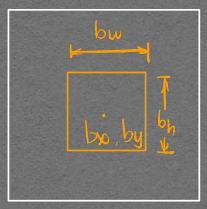
Object Localization

Classification with localization

Input Image → CNN → Labels: Cars, Pedestrian, Bike ...

Bounding Box (b_x, b_y, b_h, b_w)



$$y = \begin{bmatrix} P_c & \text{1 "is there an object?"} & 0 \\ b_x & & ? \\ b_y & \text{Bounding Box} & ? \\ b_h & & ? \\ b_w & & ? \\ c_1 & 0 & ? \\ c_2 & 1 \text{ Which Object it is?} & ? \\ c_3 & 0 & ? \end{bmatrix}$$

Don't Care If no object

$$\mathcal{L}(\hat{y}, y) = \begin{cases} (\hat{y}_1 - y_1)^2 + (y_2 - \hat{y}_2)^2 + \cdots (\hat{y}_8 - y_8)^2 & \text{if } y_1 = 1 \\ (\hat{y}_1 - y_1)^2 & \text{if } y_1 = 0 \end{cases}$$

Landmark Detection

Eye Corner as landmarks

Snapchat Emoji — Annotate nose, mouth for emotion detection

pose for pose detection

Object Detection (Multiple objects in an image)

Car Detection Example

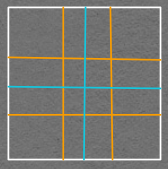
Training Set

Car Images

Non Car Images

→ CNN

Test Image



"切豆腐"

Sliding Windows Detection

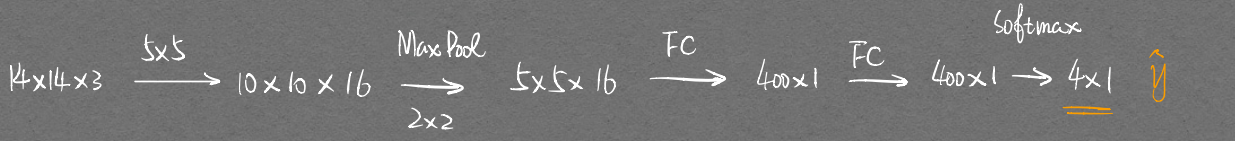
Crop 1, Crop 3, Crop 2, Crop 4 — Larger window size

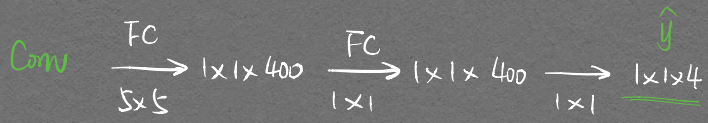
Expensive!

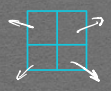
Smaller window size

Convolutional Implementation of Sliding Windows

Turn FC Layer into Convolutional Layers



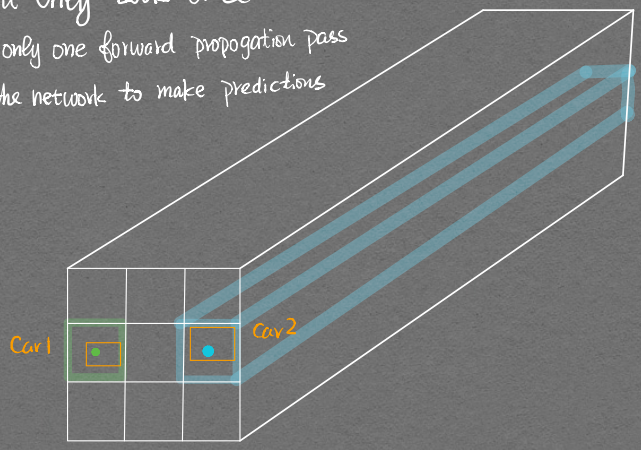


4 Sliding Windows  2x2x4

YOLO Algorithm (Get more accurate bounding boxs)

"You Only Look Once"

Requires only one forward propogation pass through the network to make predictions



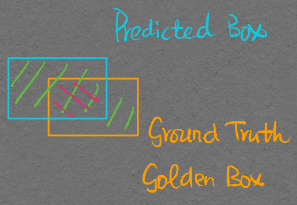
$$\begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

0.4, 0.3 } [0,1]

0.9, 0.5 } Could Be more than 1

Intersection over Union (IoU)

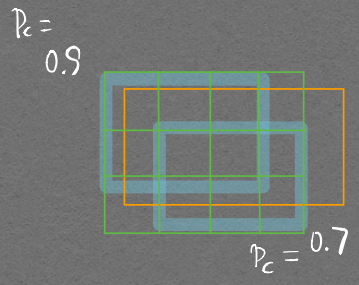
Evaluating Object Localization



"Correct" if $\frac{\text{size of}}{\text{size of}} \geq 0.5$

IoU : A measure of the overlap between 2 bounding boxes.

Non-max Suppression



Each output prediction is $\begin{bmatrix} P_c \\ b_x \\ b_y \\ b_h \\ b_w \end{bmatrix}$

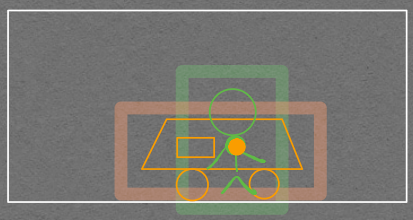
Discard all boxes with $P_c \leq 0.6$

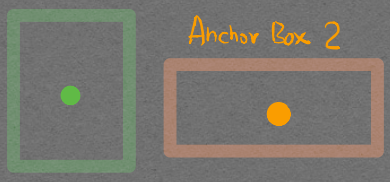
While there are any remaining boxes:

1. Pick the box with the largest P_c
   output that as a prediction
2. Discard any remaining box with IoU ≥ 0.5, with the box output in the previous step.

Darkening Overlapping Bounding Box

Overlapping Objects Anchor Boxes





Object Center points overlap
fall into the same grid

$$y = \begin{bmatrix} P_c & 1 \\ b_x & \\ b_y & \\ b_h & \\ b_w & \\ c_1 & 1 \\ c_2 & 0 \\ c_3 & 0 \\ P_c & 1 \\ b_x & \\ b_y & \\ b_h & \\ b_w & \\ c_1 & 0 \\ c_2 & 1 \\ c_3 & 0 \end{bmatrix}$$

Pedestrian

Car

Previously:

Object ⟶ Grid Cell that contains the middle point
(Grid Cell.)

With Two Anchor Boxes:

Object ⟶ Grid Cell that contains the middle point and Anchor Box with Highest IoU
(Grid Cell, Anchor Box)

Anchor Box helps specialization for object detection.

YOLO Algorithm (Putting all together)

y $3 \times 3 \times 2 \times 8$

- 3×3: Grid Cell Size
- 2: # Anchors
- 8:
  - 1 P_c
  - 4 Bounding Box Coordinates
  - 3 classes

Each grid cell → 2 Predicted Bounding Boxes

Get rid of low prob predictions → For each class, run non-max suppression → Final Outputs

Region Proposals (R-CNN)

Run CNN on only proposed Regions

Image —Segmentation Algorithm→ Blobs (of irregular shapes)

—CNN→ Label + Bounding Box

R-CNN → Fast R-CNN → Faster R-CNN

- R-CNN: Slow
- Fast R-CNN: Conv Sliding Windows
- Faster R-CNN: CNN to propose regions