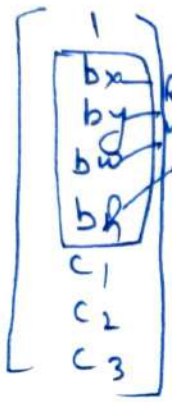
Yolo A fresh approach to object detection

Yolo (1)

- Not classification as detection
- New idea.
  - Evaluated rigrously by the authors
  - The fact that new idea is better is validated by rigrous evaluation.
- Specifying spatially separated bounding boxes and associated class

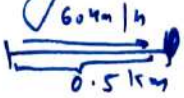
OD as as regression problem



- Single neural network
- It can be optimized end to end directly on detection problem
- ~~tapad~~ Unified architecture is extremly fast
  it outperforms

CVPR-2016 self driving car

~20ms

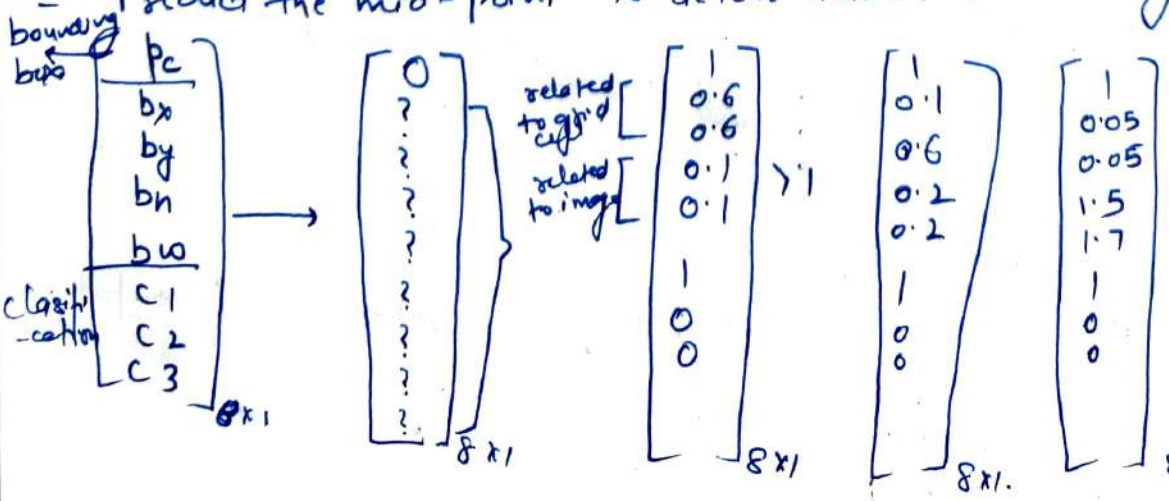


- Generalizes better
  Yolo learns generalizable representation of objects
  - Less likely to break down when applied to new domains
- Accuracy concern
  - The first version of the yolo algorithm lacks behind in accuracy

R-CNN, fast RCNN :- 7-8%.

Preparing the training data

- 8x8 size grid.
- Predict the mid-point to determine the bounding box



$$\begin{bmatrix} 0.05 \\ 0.05 \\ 1.5 \\ 1.7 \\ 1 \\ 8 \end{bmatrix}$$

Intuition for object detection.
- Yolo reasons globally about the image
- Yolo sees the entire image during training and test time
- Unlike sliding or R CNN
- It encodes contextual information about classes as well as their appearance

3 × 3 × 8

IOU Intersection over union

- objection detection is working well or not.

IoU : Intersection: Common part

Union:

$$IoU = \frac{\text{area of Intersection}}{\text{area of Union}}$$

measure to determine the bounding box similarity

$IoU \geq 0.5$ → 1

Understanding the output tensor

- Output tensor size

$S \times S \times (B * 5 + C)$

$S \times S$: size of grid

B : # bounding box

C : # of classes

why multiple bounding box?

1000 images

car

pedestrian

motorcycle

- Multiple grid cell may claim an object

Test output :- only one object

$B = 2, C = 3$ → bicycle, car, dog

$13 \times 13 \times (2*5+3)$

$= 13 \times 13 \times (13)$

$13 \times 13 \times 2 = 169 \times 2 = 338$

Non max Suppression

Each grid cell predicts.

- $P_c = Pr(\text{object}) * IoU$ (Probability of confidence)
- & conditional class probabilities
- $Pr(\text{class}_i | \text{object}) * Pr(\text{object}) * IoU$

→ Pr(object)

→ class1 class2

→ Discard $p_c \leq 0.6$

→ for remaining boxes highest p_c → output as prediction

$$\begin{bmatrix} p_c \\ b_x \\ b_y \\ b_w \\ b_h \end{bmatrix} \rightarrow$$

discard $p_c \leq 0.6$ Yolo(2)

— Discard any remaining boxes IoU ≥ 0.5

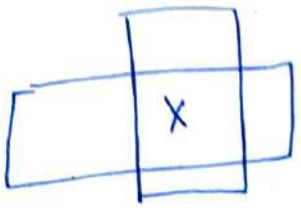
— Independently carry out non max supression for each class

Anchor boxes Limitation of Yolo:- one object per grid cell

- Anchor boxes solve it up to some extent

- can be also reduced by increasing s

Overlapping objects



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

$$\begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \end{bmatrix}$$ anchor box1

$$\begin{bmatrix} p_c \\ b_x \\ b_y \\ b_h \\ b_w \end{bmatrix}$$ anchor box2

$$\begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

each object → 1 grid cell

mid point.

each object → mid point → 1 grid cell + anchor box for the grid cell with highest IoU

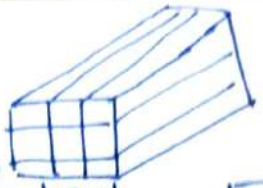
Anchor boxes:- help in specialization

— Parts of the network specialize in detection.

Yolo:- Training with anchor boxes

y is 3 × 3 × 2 × 8 → p_c + no. of bounding box

5+3 classes

Image → Trained yolo architecture → 

$P_c \leq 0.6$ = Discarded

Non max suppressed output

Limitation of YOLO

Improving object detection

Anchor box 1

Anchor box 2

classification

classification + localization

object Detection

You only look once

Instance segmentation

- why should it work?
  - Authors had an idea / intuition.
  - That alone in not sufficient.
  - Prepare for evaluation
    - This requires laborious task of data set preparation for training
  - so you better have confidence in your idea / intuition
- The fact that new idea is better is validated by rigorous evaluation.
  - Train / debug / improvise
  - Performance on test set
  - Generalize well.
- The above steps are true for any new deep learning algorithm
  - Repeat them for AlexNet (but of course for classification