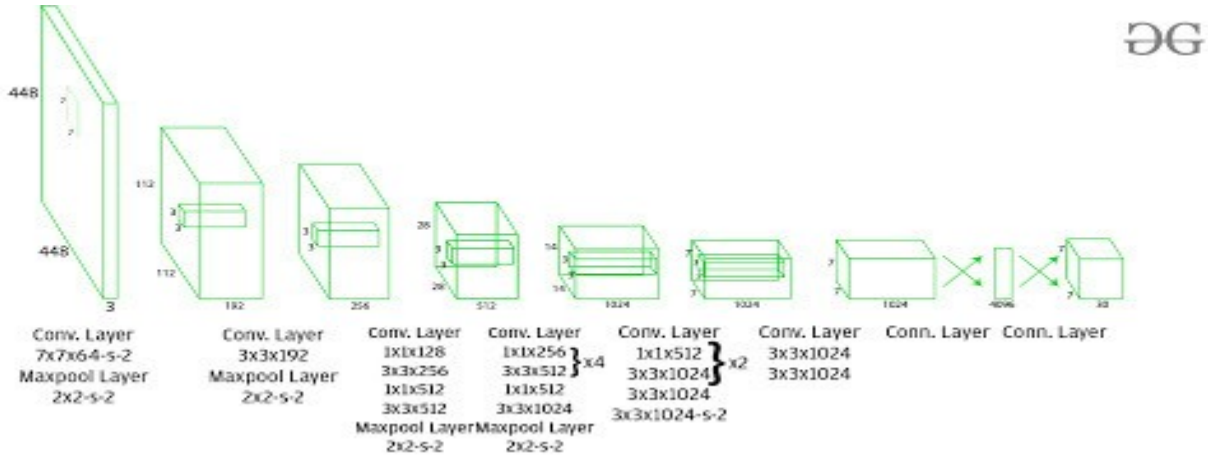
Name:Panneerselvam N
Date : 01/06/2021

You Only Look Once(YOLO v1)

Yolo is an object detection algorithm which has high Frame Per Second(FPS) with decent mAP. It predicts the bounding in Regression Way.

Architecture :



Explanation :

Yolo v1 has 24 convalutional layers and followed by two fully connected layers. First 20 convalutional layers are pre-trained with ImageNet Dataset which has 1000 clases.Next , 4 convalutional layers are trained with input labeled data.

Final output will be 3 Dimensional vectors (7x7x30). Here, 7 x7 is a number of grids and 30 is (20 classes + 2 x (1 object + 4 Coordinates)) .

There are 1x1 (Point wise Convalutional) and 3x3 convalutional are used to control number of Learning Parameter and number of operations in Convalutional Layers. This method is inspired from Inception Net.

Input Image Size :

- Model is trained with size of (224 x 224)
- Model is tested with size of (448 x 448).

Activation Functions :

- Leaky Relu in all Conve Layers
- Linear activation function in ouput layer

Operation steps:

1. First, Input image split into SxS grid cells. Each cell is responsible for predict B bounding boxes. Here S may be anything like (7x7) or (19x19) , B is number of bounding boxes.

2. Each cell's bounding box generates vectors in size (5 + numbers of class).so Total output will be (SxS + Bx5 + C)

Ex:

y =	pc
	bx
	by
	bh
	bw
	c1
	c2
	c3

In this example,
Pc (object) = 1 (if object present) or 0
Bx , By = coordinates of center of an object
Bh , bw = height and width
C1, c2, c3 = three clases

3. Bonding boxes vectors are predicted with consider of center of an object. So , there is chance to many bounding boxes for one object. This problem is solved by Non Max Suppression(NMS).

4. Simply NMS removes less confidence score bounding boxes with respect to one class. It repeat the opeartion for every classes.

5. Final output has vector of object class and corresponding coordinates with high confidence score.

Loss Function :
Loss is calculated for back propagation of network to optimize learning parameters.

There are three type loss is calculated:

- Classification Loss
- Localization Loss
- Confidence Loss

Regression loss

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$
$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

Confidence loss

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$
$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

Classification loss

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2$$

In above figure,
I means total grids (SxS)
j means number of bounding boxes in each grid cell

Performance :

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [31]	2007	16.0	100
30Hz DPM [31]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [38]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[28]	2007+2012	73.2	7
Faster R-CNN ZF [28]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

Advantages of Yolo v1:

- Good speed compared to other algorithms.
- Less background mistakes
- Unified Network Architecture

Limitations :

- Faces difficulties while predicting small objects
- More localization error with compare to Faster RCNN
- If two object has same center point , Yolo v1 faces difficulties for predicting two classes.