YOLO Introduction

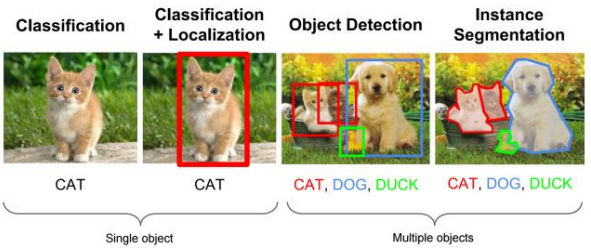
- You only look once, real time object detection deep learning network -

2020 - 2021

Ando Ki, Ph.D.
adki@future-ds.com

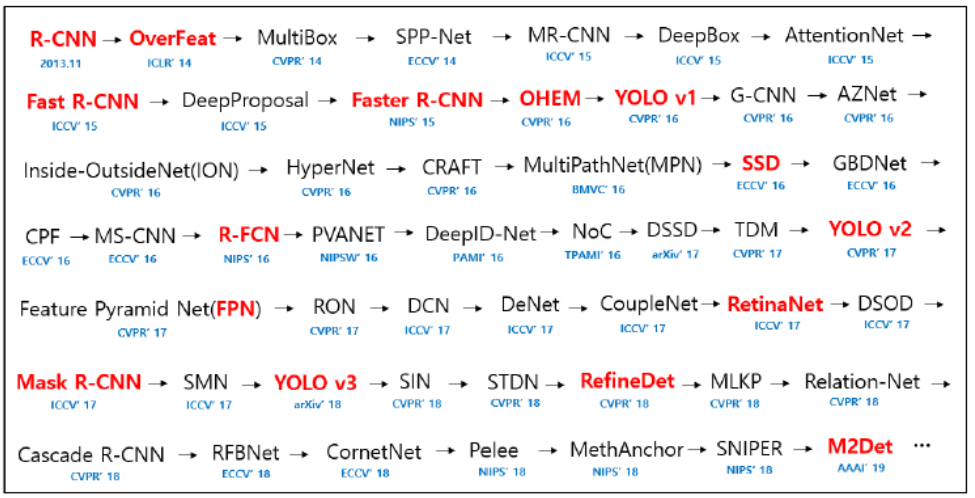
Table of contents

Object recognition / detection



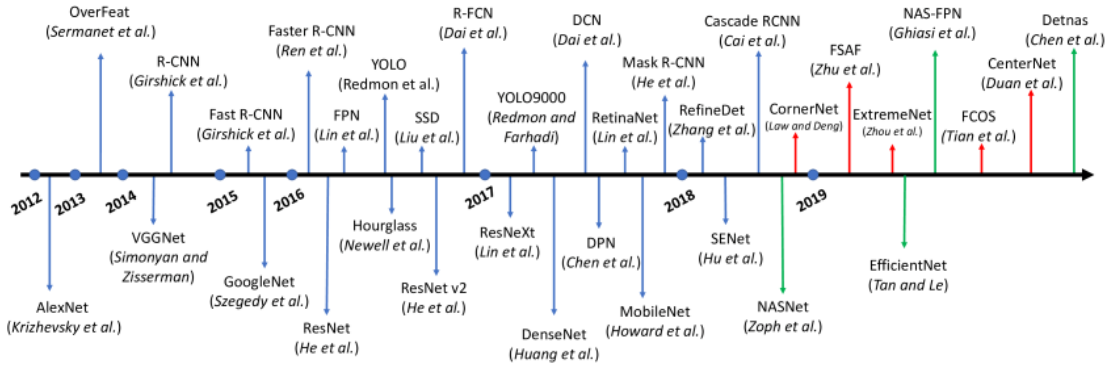
- Image classification
  - to figure out which category is in the picture
- Object localization
  - to figure out where the object locates
  - object localization + classification: for one object
- Object detection
  - to find all the objects in the image and draw bounding boxes
    - dealing with multiple objects in the picture
    - draw bounding box
- Instance segmentation (semantic segmentation)
  - to find exact boundaries of objects

Object detection: state of the art progress



https://deeplearning.mit.edu

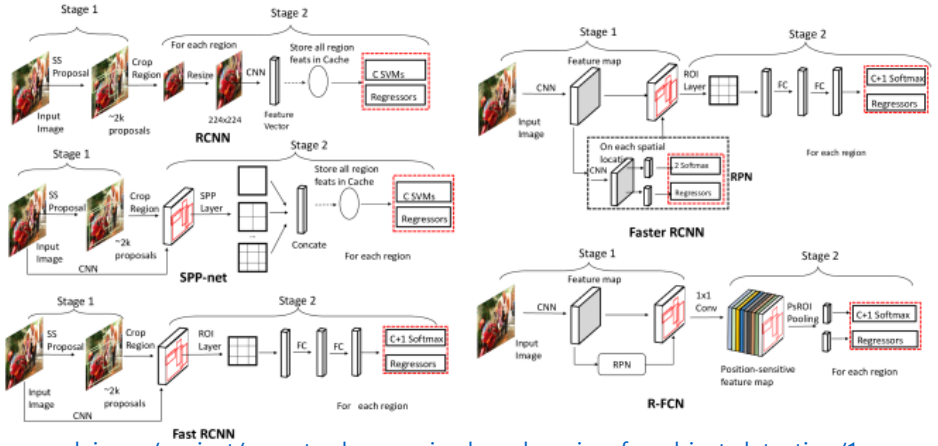
Object detection: state of the art progress



https://www.groundai.com/project/recent-advances-in-deep-learning-for-object-detection/1

Object detections: R-CNN (Region-based CNN)

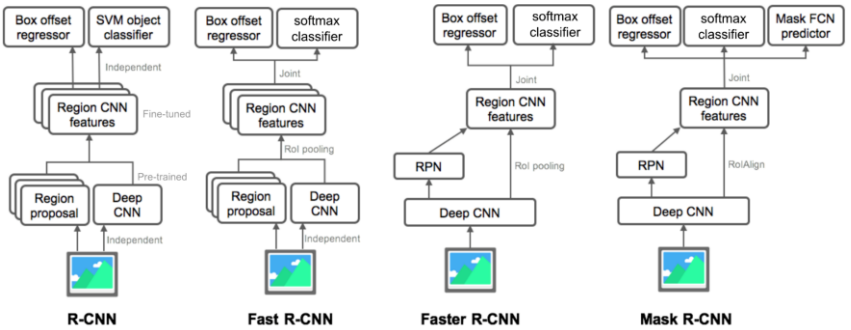
- Two-stage detectors: proposal generation and region classification



https://www.groundai.com/project/recent-advances-in-deep-learning-for-object-detection/1

Object detections: R-CNN (Region-based CNN)

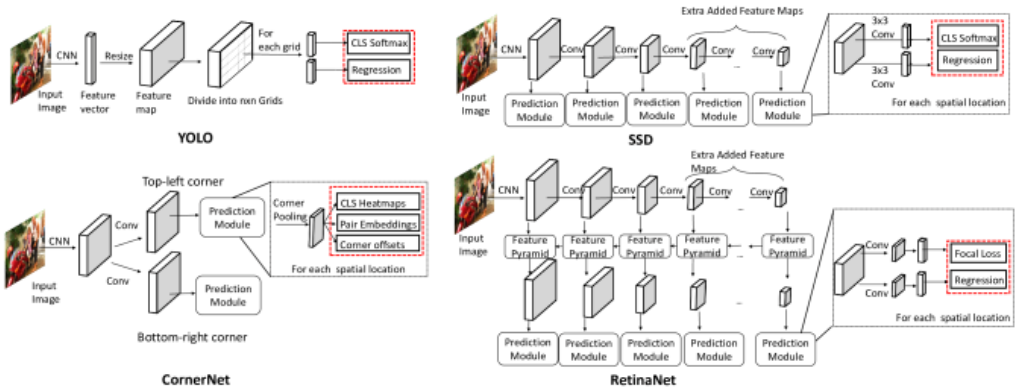
- Two-stage detectors: proposal generation and region classification
  - 1. First, the model proposes a set of regions of interests by select search or regional proposal network and then
  - 2. A classifier only processes the region candidates



https://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html

Object detections: YOLO

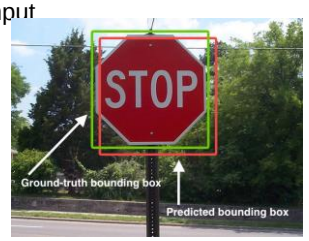
- One-stage detectors (unified detectors)



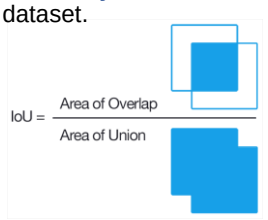
https://www.groundai.com/project/recent-advances-in-deep-learning-for-object-detection/1

Terminologies

- GT: Ground Truth box (i.e., hand labeled box)
  - the hand labeled bounding boxes from the training/testing set that specify *where* in the image our object is
  - represents the desired output (ideal output) of an algorithm on an input
- PB: Predicted box
  - calculated box



- IoU (Intersection over Union)
  - an evaluation metric used to measure the **accuracy of an object detector** on a particular dataset.



Labeled data가 있으므로 계산이 가능

Terminologies

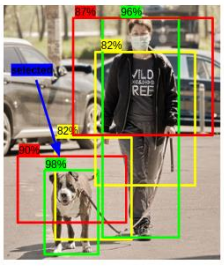
- Confidence score
  - how certain it is that the predicted bounding box actually encloses some object.
    - This score doesn't say anything about what kind of object is in the box, just if the shape of the box is any good.
    - 0 means no object

Confidence Score: Pr(Object)∗IOU(pred, truth)

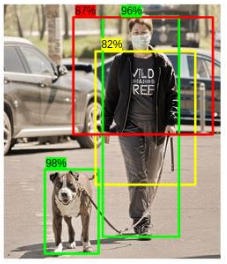
Labeled data가 있으므로 계산이 가능

Terminologies

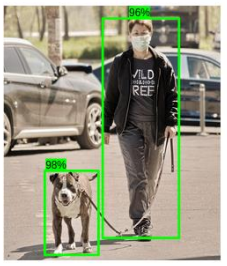
- Non-max suppression (NMS)
  - Removes bounding boxes (ROI: region of interest) with low confidence score, since most of bounding boxes will not contain an object.



Step 1: Selecting Bounding box with highest score



Step 3: Delete Bounding box with high overlap

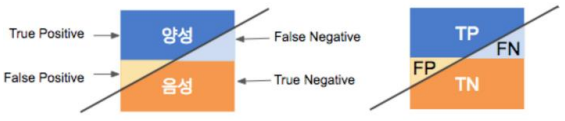


Step 5: Final Output

- Step 1: Select the box with highest objectiveness score
- Step 2: Then, compare the overlap (intersection over union) of this box with other boxes
- Step 3: Remove the bounding boxes with overlap (intersection over union) >50%
- Step 4: Then, move to the next highest objectiveness score
- Step 5: Finally, repeat steps 2-4

Terminologies

- Confusion matrix
  - P = TP + FN
  - N = FP + TN



- Accuracy = (TP+TN)/(P+N)
  - 전체 중 제대로 예측한 비 (모델의 정확도)
- Error rate = (FN+FP)/(P+N)
  - 전체 중 잘 못 분류한 비

https://bcho.tistory.com/m/1206 https://sumniya.tistory.com/26

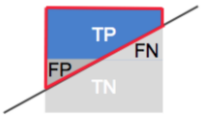
- Confusion matrix
  - True Positive(TP) : 실제 True인 정답을 True라고 예측 (정답)
  - False Positive(FP) : 실제 False인 정답을 True라고 예측 (오답)
  - False Negative(FN) : 실제 True인 정답을 False라고 예측 (오답)
  - True Negative(TN) : 실제 False인 정답을 False라고 예측 (정답)

		실제 정답	
		True	False
분류 결과	True	True Positive	False Positive
	False	False Negative	True Negative

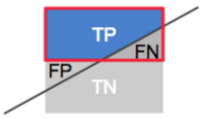
<Fig1. Confusion matrix>

Terminologies

- Precision = TP/(TP+FP)
  - 정밀도, 정확도
  - 맞다고 예측한 것 중 실제로 옳은 것의 비



- Sensitivity (Recall) = TP/(TP+FN)
  - 민감도, 재현율, 검출율
  - 옳다고 예측 한 것 (사선 위) 중 정말 옳은 것(TP)이 전체 옳은 것(P)에 대한 비

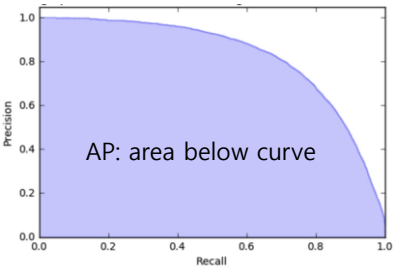


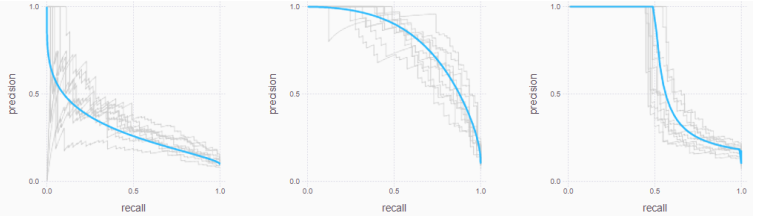
https://bcho.tistory.com/m/1206
https://sumniya.tistory.com/26

- Precision (1 – false_alarm_rate)
  - **PPV(Positive Predictive Value, Positive 정답률)**
    - 날씨 예측 모델이 맑다로 예측했는데, 실제 날씨가 맑았는지를 살펴보는 지표
      - 30일 중 20일이 맑았는데, 확실한 2일만 맑다고 예측했고 2일이 맑았다면, precision은 100%이지만 무슨 의미?
        - FP를 줄여서 TP를 극대화 함.
- Recall (1 – miss_rate)
  - Sensitivity or hit rate
    - 실제 날씨가 맑은 날 중에서 모델이 맑다고 예측한 비율을 나타낸 지표
    - 30일 중 25일이 맑다고 하였고, 이중 20일이 정말 맑았다면, recall은 100%이지만 무슨 의미?
      - TP와 FP를 최대화하여 TP를 극대화함
- Precision과 Recall이 모두 높아야 좋은 모델

Terminologies

- Average Precision
  - 한 prediction에서 여러 class에 대한 민감도(recall)와 정밀도(precision)을 2차원으로 표현하고 (precision-recall curve)
  - 이 것의 적분으로 object detection의 정도를 측정 ➔ AP (0 ~ 1)
    - higher value is better
- mAP (Mean Average Precision)
  - 모든 class에 대해 평균을 낸 것➔ mAP





Terminologies

- Let define true positive when IoU>0.5

True positive
IoU of predicted BB (yellow) and GT BB (blue) >= 0.5 with correct prediction

True negative



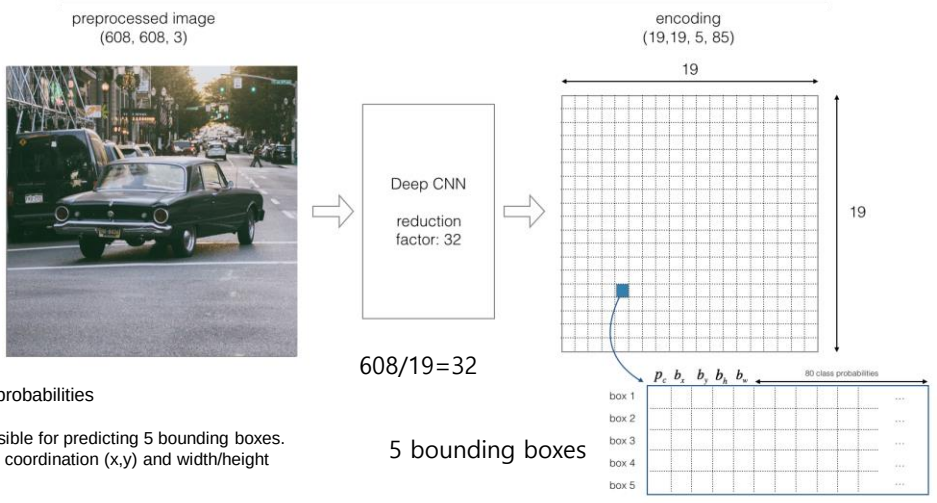
False positive
IoU<0.5
Duplicated BB


False negative
no detection at all
wrong prediction even IoU>=0.5
→ horse not person

https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52

- Let's IoU be 0.5
  - TP: True Positive, if IoU>=0.5.
  - NP: False Positive, if IoU<0.5 including detection for the object that has no Ground Truth.
  - TN: False Negative, if no detection for the object that has Ground Truth.
  - FN: True Negative, nothing to do with object detection.
    - ➲ TN is every part of the image for the result of no-prediction.

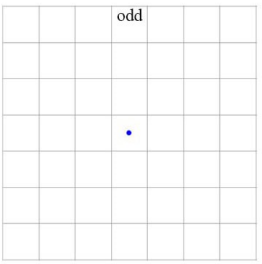
Terminologies: grid



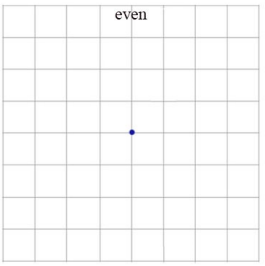
- Grid, bounding box, class probabilities
  - 19x19 grid.
  - Each grid cell is responsible for predicting 5 bounding boxes.
  - Each bounding box has coordination (x,y) and width/height (w,h)
  - 80 classes

5 bounding boxes

Why odd number of grid



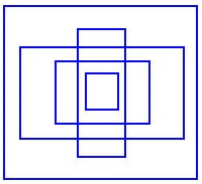
7x7 grid



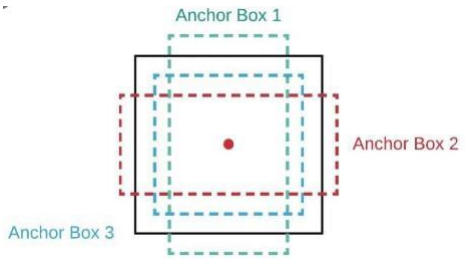
8x8 grid

- The center of a picture is often occupied by a large object. With an odd number grid cell, it is more certain on where the object belongs.

Anchor boxes

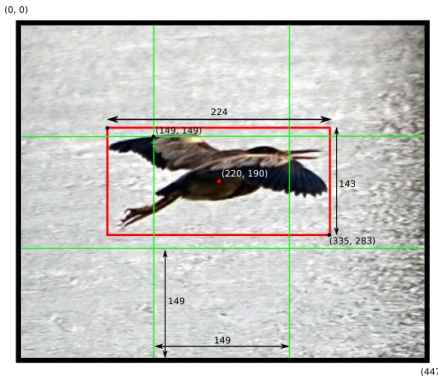


5 Anchor boxes



- **Anchor boxes** (also called default boxes) are a set of predefined box shapes selected to match ground truth bounding boxes, because most of objects in the training dataset or generally in the world (e.g. person, bicycle, etc.) have a typical height and width ratio.
- Instead of predicting 5 arbitrary boundary boxes, we predict offsets to each of the anchor boxes above. If we **constrain** the offset values, we can maintain the diversity of the predictions and have each prediction focuses on a specific shape. So the initial training will be more stable.

Terminologies: Anchor boxes



x = (220-149) / 149 = 0.48
y = (190-149) / 149 = 0.28
w = 224 / 448 = 0.50
h = 143 / 448 = 0.32

- Grid and bounding box example
  - Example of how to calculate box coordinates in a 448x448 image with S=3.
  - Note how the (x,y) coordinates are calculated relative to the center grid cell.
  - Note how the (w,h) ratio are calculated relative to the size of image.
- predict the box center (tx and ty in the figure 6) w.r.t the ***top left corner of its grid*** scaled by ***grid width and height .***
- Predict the width(tw) and height(th) of the box ***w.r.t an anchor box*** (pw and ph)

Terminologies: Anchor boxes

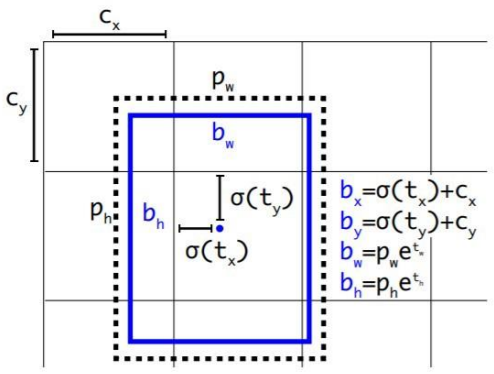


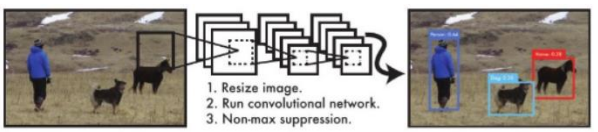
Figure 3: Bounding boxes with dimension priors and location prediction. We predict the width and height of the box as offsets from cluster centroids. We predict the center coordinates of the box relative to the location of filter application using a sigmoid function.

- Instead of predicting the absolute size of boxes w.r.t the entire image, Yolo introduces what is known as **Anchor Box**, a list of predefined boxes that best match the desired objects.
- The predicted box is scaled w.r.t the anchors.
- predict the box center (tx and ty in the figure 6) w.r.t the ***top left corner of its grid*** scaled by ***grid width and height .***
- Predict the width(tw) and height(th) of the box ***w.r.t an anchor box*** (pw and ph)

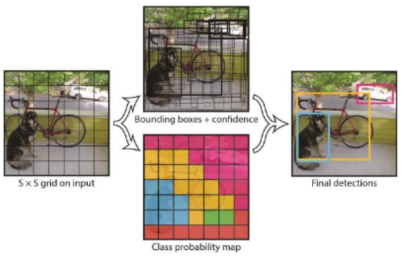
Loss function

- YOLO uses sum-squared error between the predictions and the ground truth to calculate loss. The loss function composes of:
  - the **classification loss**.
  - the **localization loss** (errors between the predicted boundary box and the ground truth).
  - the **confidence loss** (the objectness of the box).

YOLO (V1) detection system

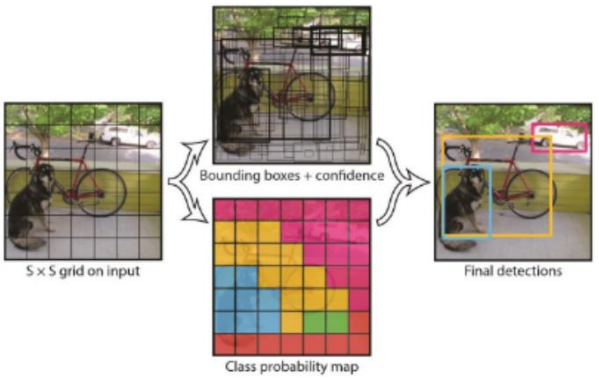


- (1) resize input image to 448x448
- (2) run a single convolution network: a regression
- (3) get result by confidence



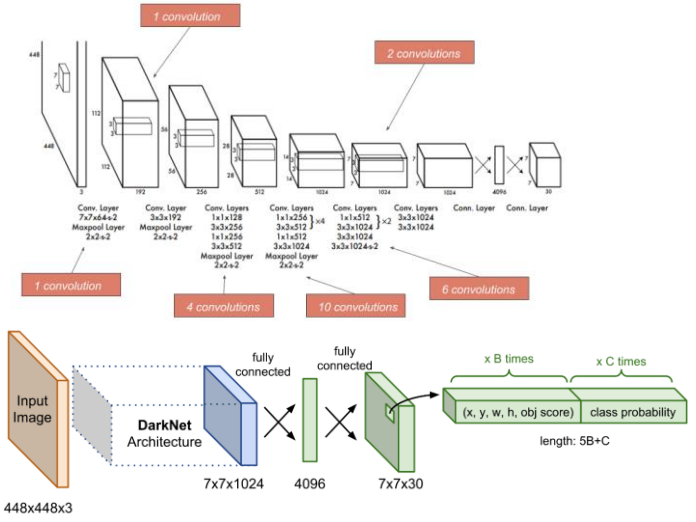
- (1) divides the image into an SxS (7x7) grid
- (2) predicts B (2) bounding boxes for each grid cell
  - only for bounding boxes those center fall in the grid
- (3) Get confidence for the boxes of C class probabilities

YOLO (V1) detection system

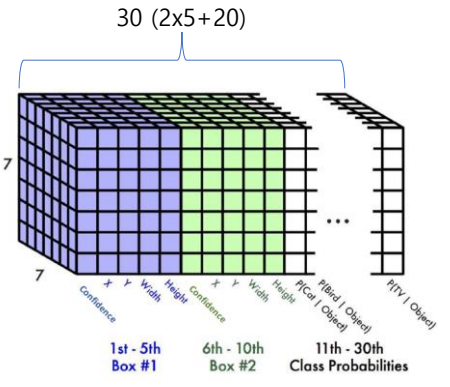


- Divide the input image into an **S × S** grid.
- Each grid cell predicts **B** bounding boxes.
- Each bounding box :
  - **Confidence** = $Pr(oggetto) * IOU_{pred}^{truth}$.
  - $\boldsymbol{x, y, w, h}$ = (x, y) bb center, w width, h height
- **C** class probabilities.
- Prediction = **S × S × (B * 5 + C)**

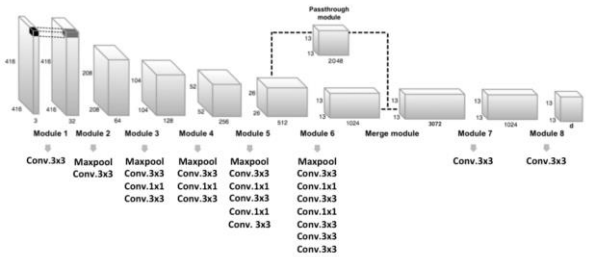
YOLO V1



- 24 convolution layers
- 2 fully connected layers



YOLO 2

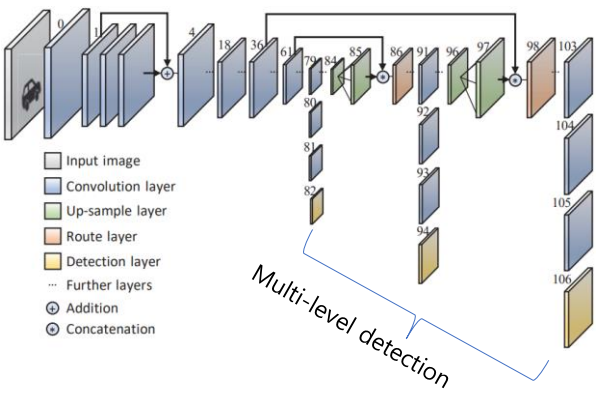


- Use darknet-19 architecture for feature extractor
  - While GoogleNet is used for YOLOv1.
  - 19 convolutional layer, 5 max-pooling layer,
- 30 layer architecture
- Batch normalization
- Anchor boxes
- High resolution input
  - 224x224 ➔ 448x448
- Fine-grained features
  - 13x13 ➔ 26x26
- No fully connection network at classifier layer

YOLO 9000

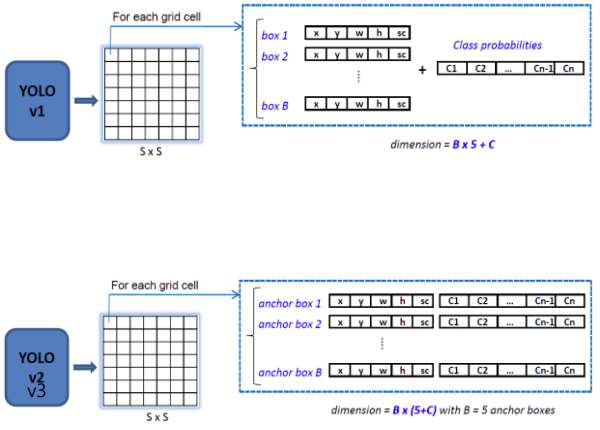
- YOLO9000
  - a real-time system that detects more than 9000 objects categories by combining COCO's detection dataset (80 classes) with ImageNet's classification dataset (~22K classes).
  - Use YOLO V2 that trained separately for classification and detection. ➔ Rich dataset training

YOLO V3



- Use darknet-53 architecture for feature extraction
- 106 layer architecture

YOLO outputs



- S: grid, B: bounding box, C: classes
- Yolo V1
  - (S, S, B x 5 + C): S=7, B=2, C=20 for VOC
  - 448x448 image
- Yolo V2
  - (S, S, B x (5 + C)): S=13, B=5, C=20 for VOC
  - 416x416 image
- YoloV3
  - (S, S, B x (5 + C)): S=13, B=3, C=80 for COCO
  - additionally, two more levels
    - (2S, 2S, B x (5 + C)) & (4S, 4S, B x (5 + C))
  - 416x416 image

Yolo weights

Parameters	Models		
	YOLO [10]	YOLOv2 [11]	YOLOv3 [12]
Number of layers	31	31	106
Multilevel prediction	-	-	3 levels
Anchor boxes	-	5	3
Input image size	448 × 448	416 × 416 544 × 544 608 × 608	320 × 320 416 × 416 608 × 608
Size of weight file	753 MB	258 MB	237 MB

References

- YOLO: Real-Time Object Detection
  - YOLO V3: https://pjreddie.com/darknet/yolo
  - YOLO V2: https://pjreddie.com/darknet/yolov2
  - YOLO V1: https://pjreddie.com/darknet/yolov1