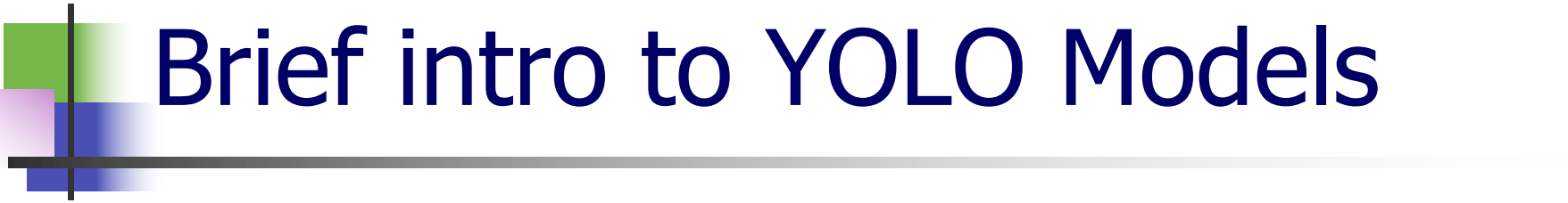
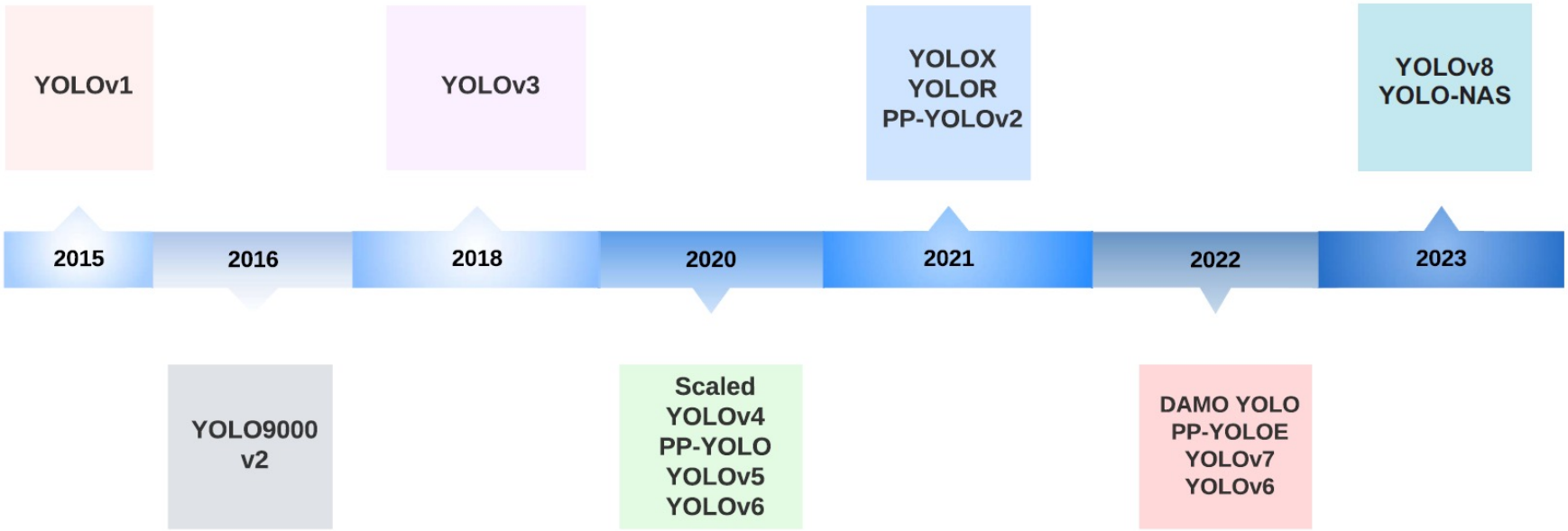


Brief intro to YOLO Models

Sang Yup Lee

YOLO (You Only Look Once) models

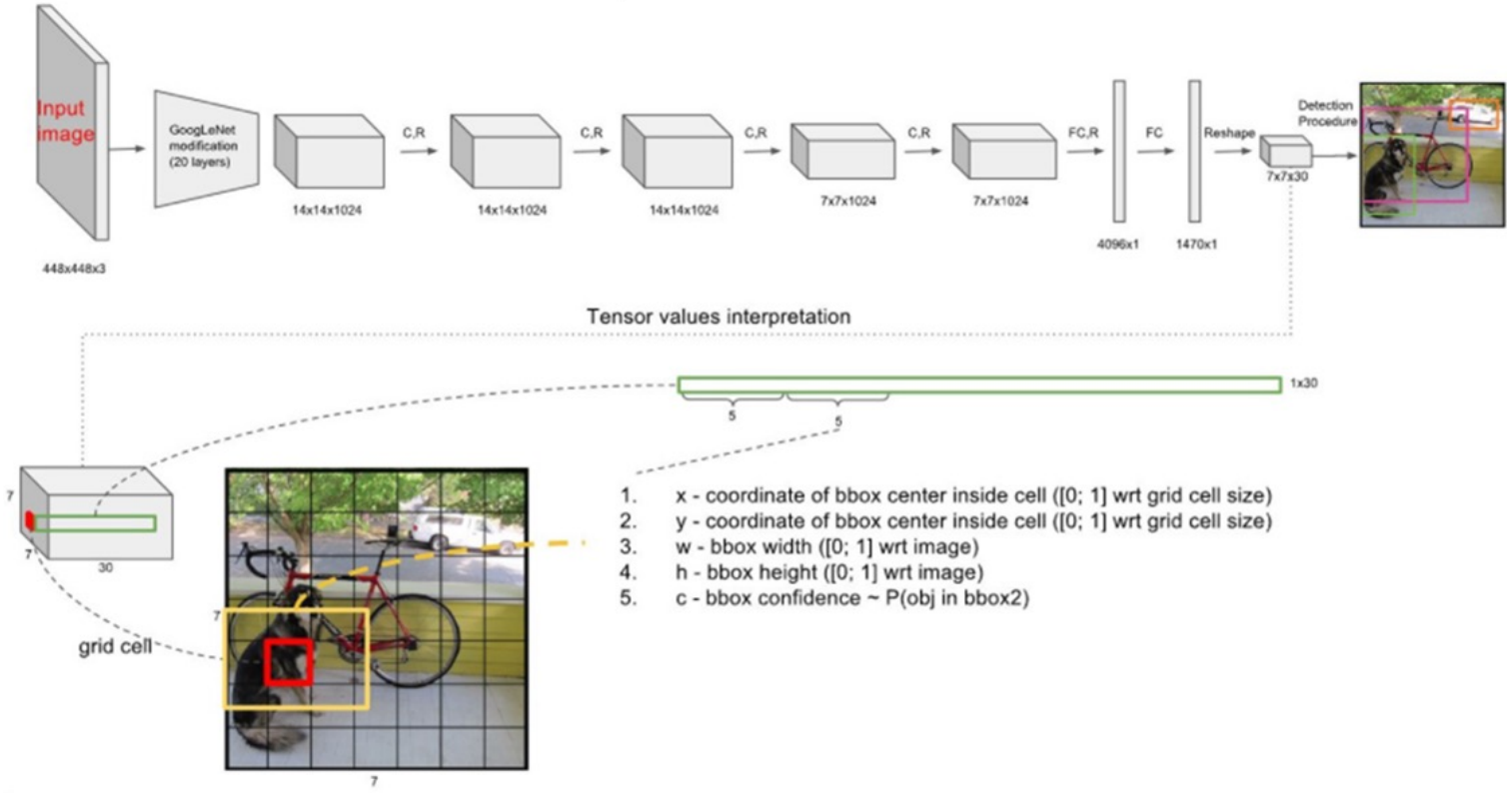
- Timeline



<source: Terven, J., & Cordova-Esparza, D. A comprehensive review of YOLO: From YOLOv1 and beyond. arXiv 2023. *arXiv preprint arXiv:2304.00501*.>

YOLO v1

- 모형의 구조



Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

YOLO v1

- 주요 특성
  - InceptionNet을 이용해서 7x7x1024 형태의 feature map 생성 (7x7 셀 존재)
  - 하나의 셀이 하나의 객체 탐지
  - 하나의 셀이 두 개의 후보 bounding box를 예측
    - 이는 anchor box는 아님
    - 각 bounding box에 대해 좌표와 confidence score를 계산
    - 이 중 confidence score가 높은 상자를 책임 상자로 간주
  - 하나의 셀에 대해서 물체가 각 클래스에 속할 확률 계산

YOLO v1

- 학습
  - PASCAL VOC 데이터셋 사용
- 비용함수

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

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

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

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

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

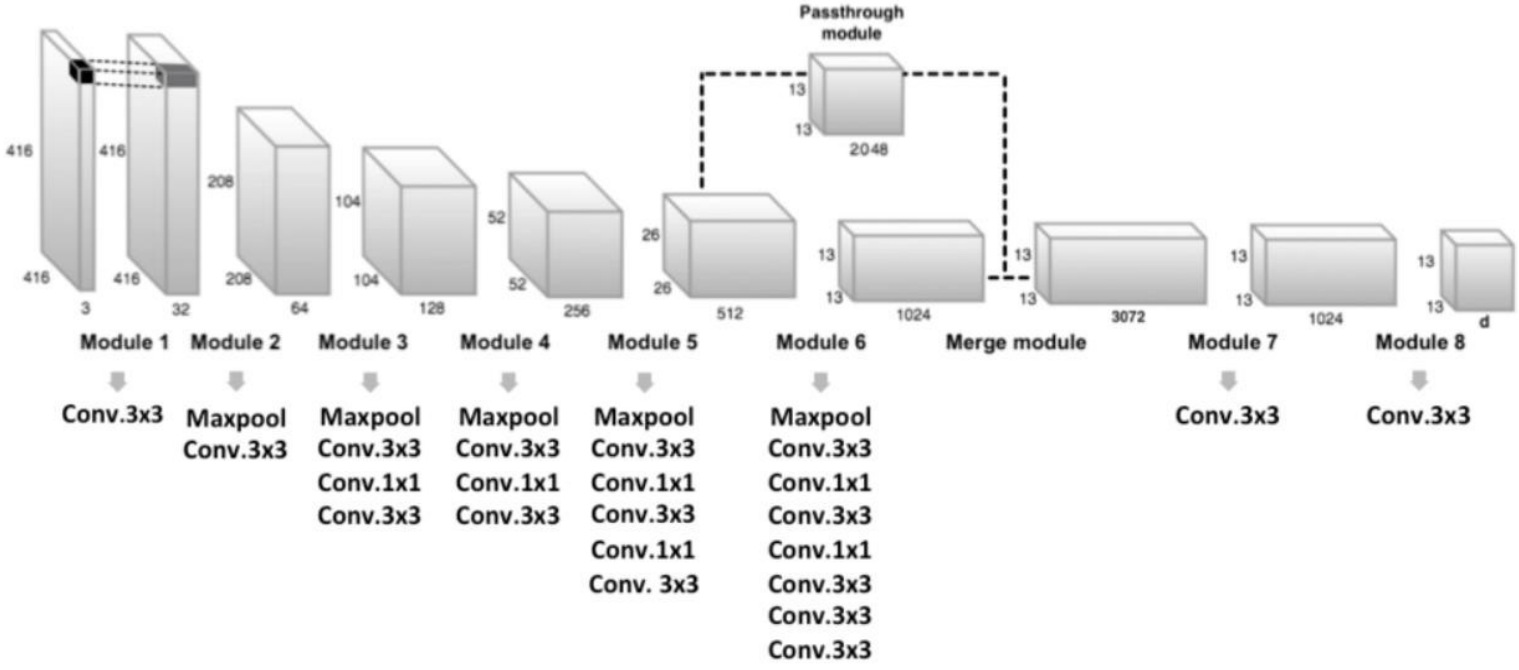
YOLO v1

- 성능

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

YOLO v2

- 모형의 구조

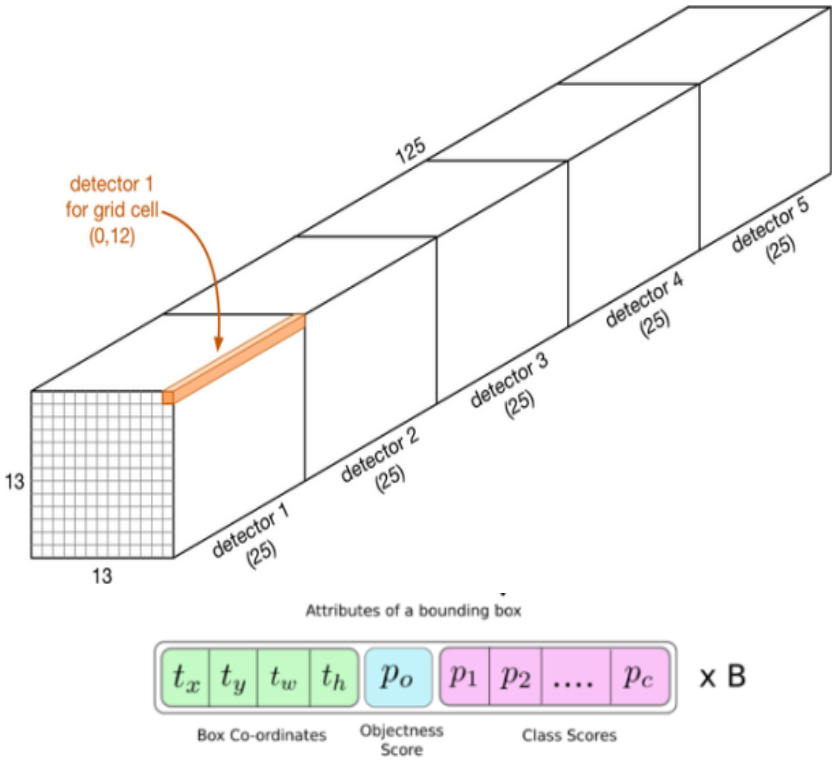


YOLO v2

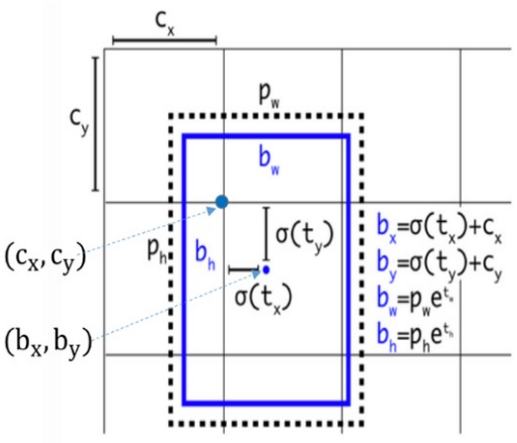
- 주요 특성
  - Anchor box 사용
    - 각 셀마다 5개 AB 사용
    - AB 형태를 결정하기 위해 군집화 방법 (K-Means) 사용
  - PassThrough module 사용
    - Skip connection 과 유사
    - 작은 물체를 더 잘 찾기 위해서
  - Feature extractor
    - Darknet-19 사용
  - Multi-Scale Training
    - 학습시 10 회 배치 마다 입력 이미지 크기를 320 부터 608 까지 동적으로 변경 (32 의 배수로 설정)

YOLO v2

- Anchor box 별 예측: AB 당 25개의 값 예측



$$b_x = \sigma(t_x) + c_x$$
$$b_y = \sigma(t_y) + c_y$$
$$b_w = p_w e^{t_w}$$
$$b_h = p_h e^{t_h}$$



- (pw,ph): anchor box size
- (tx,ty,tw,th): 모델 예측값
- (bx,by), (bw,bh): 예 측 BB 의 중심좌표와 너비 & 높이

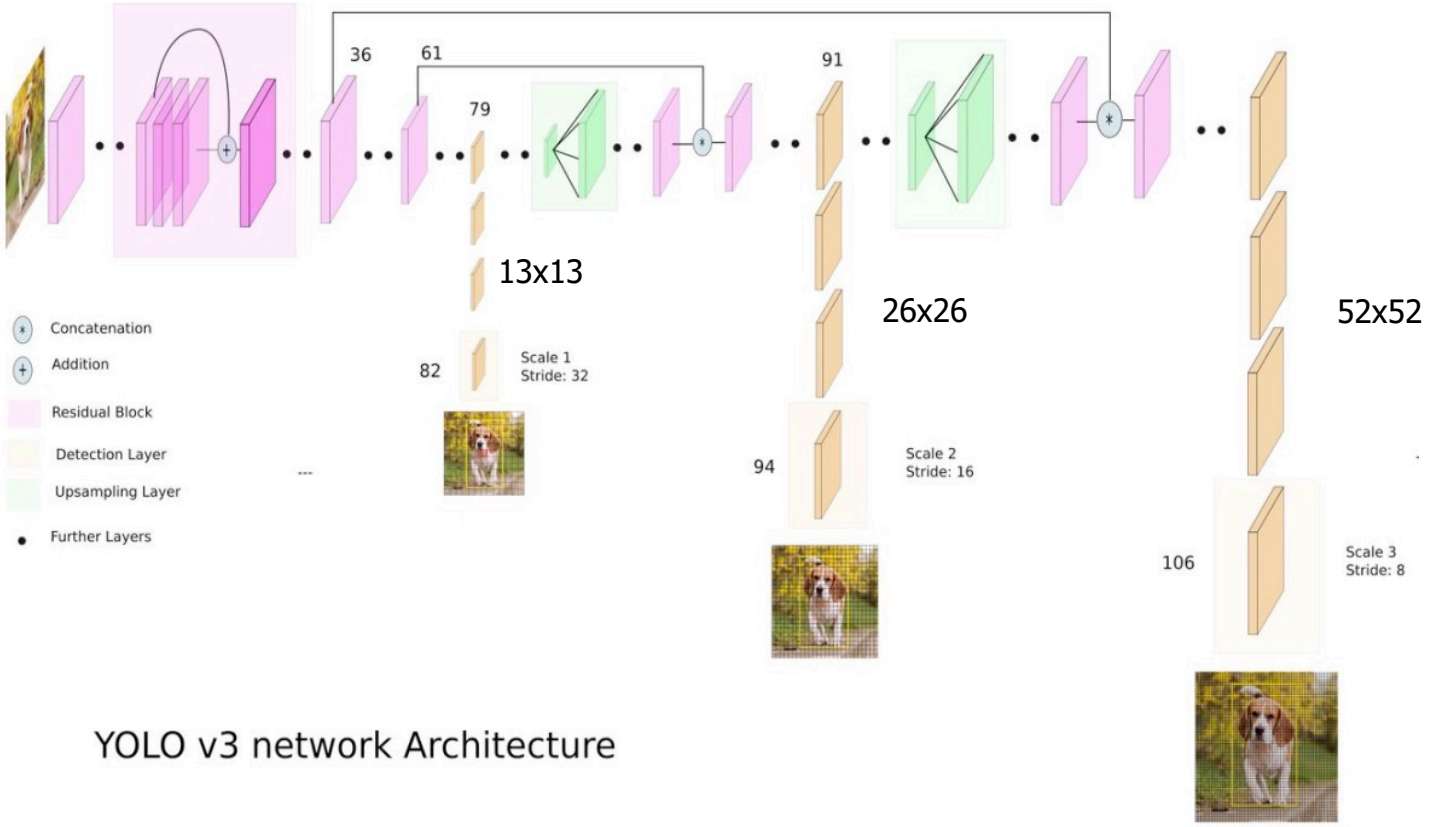
YOLO v2

- 성능

Detection Frameworks	Train	mAP	FPS
Fast R-CNN [5]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[15]	2007+2012	73.2	7
Faster R-CNN ResNet[6]	2007+2012	76.4	5
YOLO [14]	2007+2012	63.4	45
SSD300 [11]	2007+2012	74.3	46
SSD500 [11]	2007+2012	76.8	19
YOLOv2 288 × 288	2007+2012	69.0	91
YOLOv2 352 × 352	2007+2012	73.7	81
YOLOv2 416 × 416	2007+2012	76.8	67
YOLOv2 480 × 480	2007+2012	77.8	59
YOLOv2 544 × 544	2007+2012	**78.6**	40

YOLO v3

- 모형의 구조



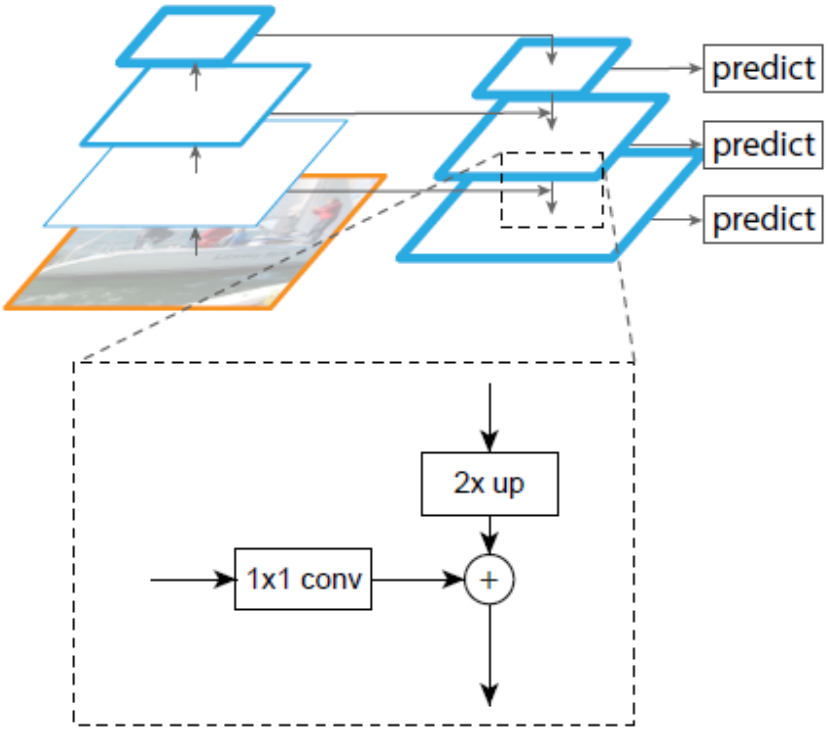
YOLO v3 network Architecture

YOLO v3

- 주요 특징
  - 3개의 feature map을 이용해서 detection 수행
    - Feature pyramid network 구조를 적용
    - 13x13, 26x26, 52x52 각 feature map에 대한 object detection 작업, 즉, loss 함수를 계산함
  - Darknet-53을 backbone으로 사용
    - ResNet을 개선한 방법

YOLO v3

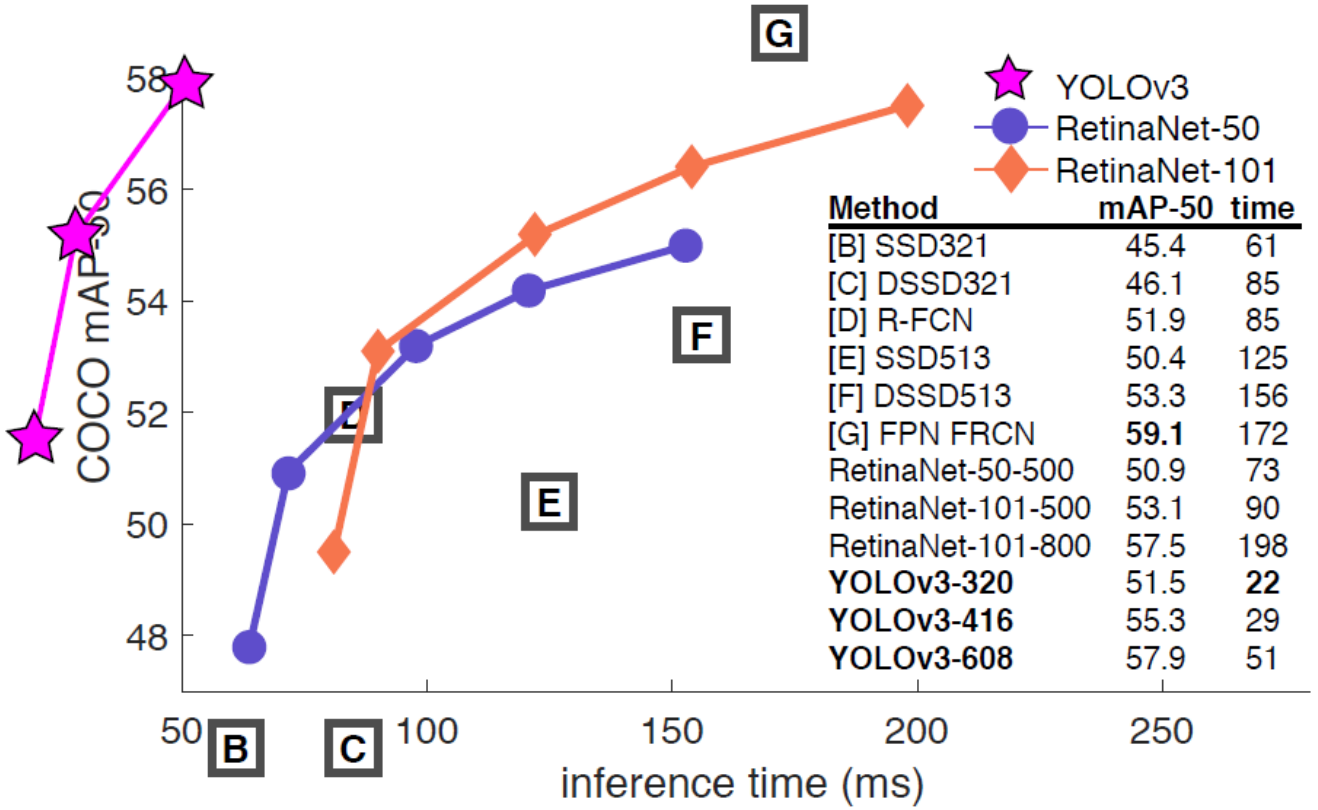
- Feature pyramid network



Lin, T. Y., Dollár, P., Girshick, R., He, K., Hariharan, B., & Belongie, S. (2017). Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2117-2125).

YOLO v3

- 성능



Method	mAP-50	time
[B] SSD321	45.4	61
[C] DSSD321	46.1	85
[D] R-FCN	51.9	85
[E] SSD513	50.4	125
[F] DSSD513	53.3	156
[G] FPN FRCN	**59.1**	172
RetinaNet-50-500	50.9	73
RetinaNet-101-500	53.1	90
RetinaNet-101-800	57.5	198
YOLOv3-320	51.5	**22**
YOLOv3-416	55.3	29
YOLOv3-608	57.9	51

Summary

항목	V1	V2	V3
원본 이미지 크기	446 X 446	416 X 416	416 X 416
Feature Extractor	Inception 변형	Darknet 19	Darknet 53
Grid당 Anchor Box 수	2개(anchor box는 아님)	5개	Output Feature Map당 3개 서로 다른 크기와 스케일로 총 9개
Anchor box 결정 방법		K-Means Clustering	K-Means Clustering
Output Feature Map 크기 (Depth 제외)	7 x 7	13 x 13	13 x13, 26 X 26, 52X52 3개의 Feature Map 사용
Feature Map Scaling 기법			FPN(Feature Pyramid Network)