YOLOv3: An Incremental Improvement

Joseph Redmon
Ali Farhadi

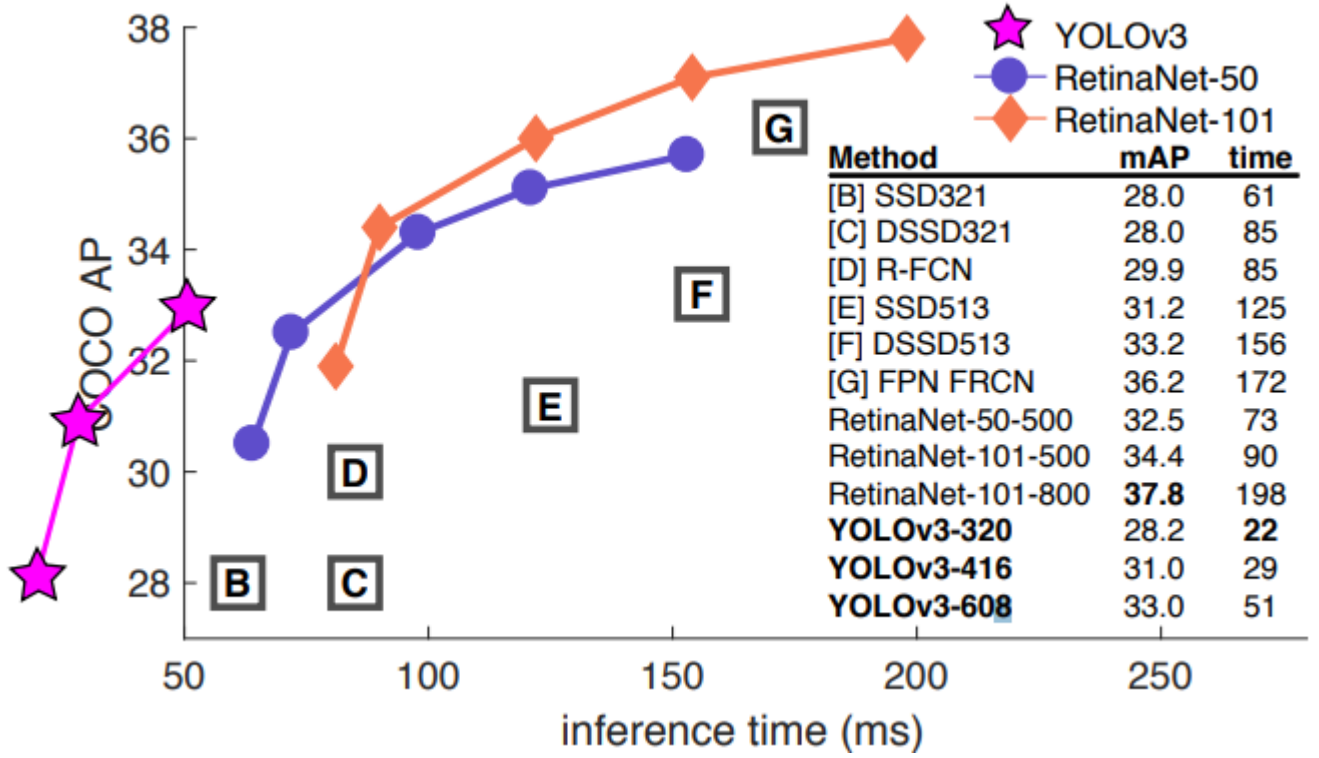
Ⅰ. Paper보다 Tech Report

Ⅱ. YOLOv2에서 약간의 변형.

Ⅲ. 속도에 강점을 둠.

Ⅳ. Metric 방식에 대한 비판.

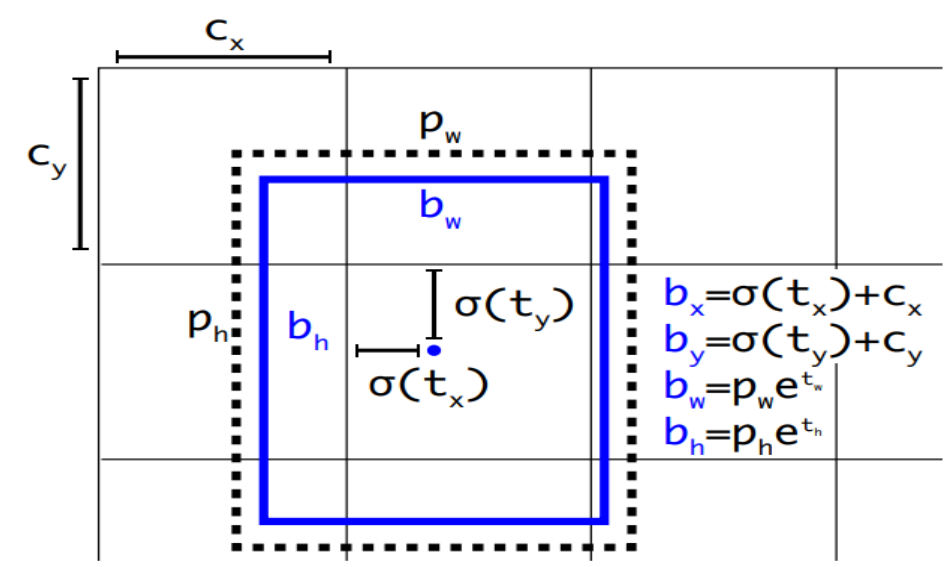
Ⅴ. 기술 윤리에 대한 의견.

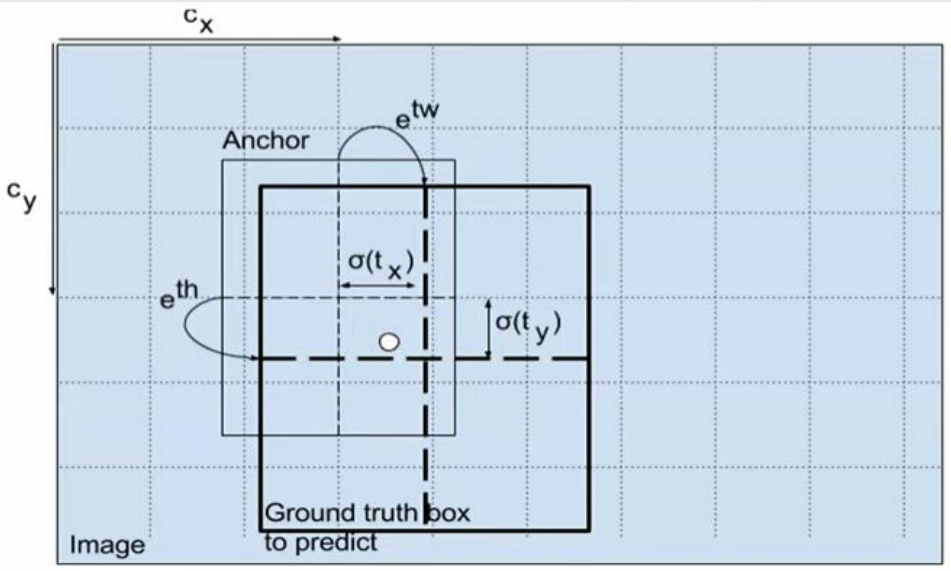


Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.2	**22**
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

- COCO Metric으로는 320x320 22ms, 28.2 mAP로 SSD보다 세배 빠름.
- .5 IOU mAP(AP_{50})으로는 57.9로 57.5, 198ms의 RetinaNet보다 더 정확하며 3.8배 더 빠름.
- 공식 코드: https://pjreddie.com/darknet/yolo/

Bounding Box Prediction





- 전체적으로 YOLO9000의 시스템의 Bounding Box prediction 방법을 따름(Dimension clusters as Anchor boxes).

- Anchor 박스의 중심점이 (C_x, C_y)로 처음에 주어 졌을 때, (C_x, C_y)에서 1칸 내에서 중심점이 이동함.
-> (t_x, t_y)를 구해서 시그모이드를 취하면 0과 1 사이가 됨.

- Anchor 박스의 원래 크기가 (P_w, P_h)로 주어졌을 때, Ground Truth 박스의 크기에 맞게 Anchor의 크기 변경.
-> (t_w, t_h)를 구해서 Exponential를 취해서 Anchor 박스의 너비와 높이에 곱해서 구함.

- 훈련 간 Box prediction과 관련된 손실은 GT와 Predicted box간의 제곱 오차의 합으로 계산함.
-> b와 관련된 식을 거꾸로 전개하면 t값들을 계산할 수 있는데 GT와 Predicted box의 t값들을 계산함.

Bounding Box Prediction

- Bounding box가 얼마나 객체를 포함하고 있는가를 나타내는 Objectness score를 각 Bounding box마다 Logistic regression으로 계산함.
- 어떤 박스가 다른 박스보다 가장 Ground Truth 박스와 많이 겹친다면 Objectness score가 1이 됨.
- 어떤 박스가 Ground Truth와 가장 많이 겹치지는 않지만 Threshold보다 많이 겹칠 때, 이 박스는 무시됨(저자들은 0.5 threshold 값 적용).
- 각 Ground Truth마다 1개의 Anchor 박스만 할당함. 그래서 어떤 박스가 Ground Truth에 할당되지 않는다면 Objectness 손실만 계산하고 Coordinate나 Class와 관련된 손실은 계산하지 않음.

Class Prediction

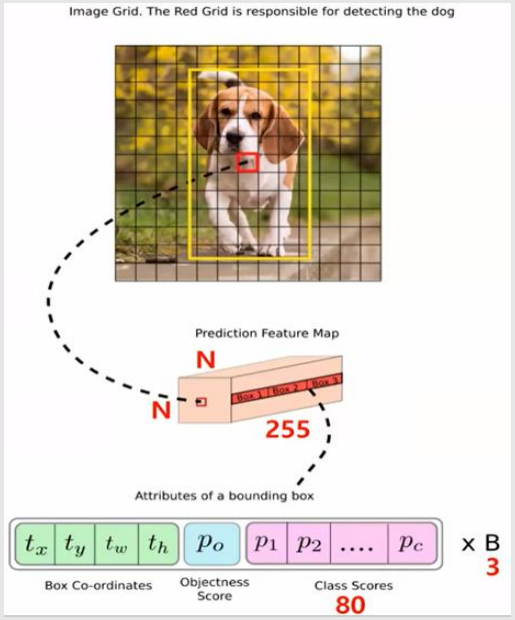
- 각 box마다 Multilabel classification을 수행 시에 Softmax대신에 각 클래스마다 Logistic regression 수행.

=> 데이터 마다 클래스가 동시에 존재하는 경우가 있는데(Woman and Person) Softmax는 각 객체가 정확히 1개의 클래스만 가진다는 것을 전제하므로 맞지 않음. 따라서 각 클래스마다 Logistic regression을 수행해서 여러 개의 클래스를 가질 수 있도록 함.

- Class 예측과 관련된 손실은 Binary cross-entropy로 계산함.

Predictions Across Scales

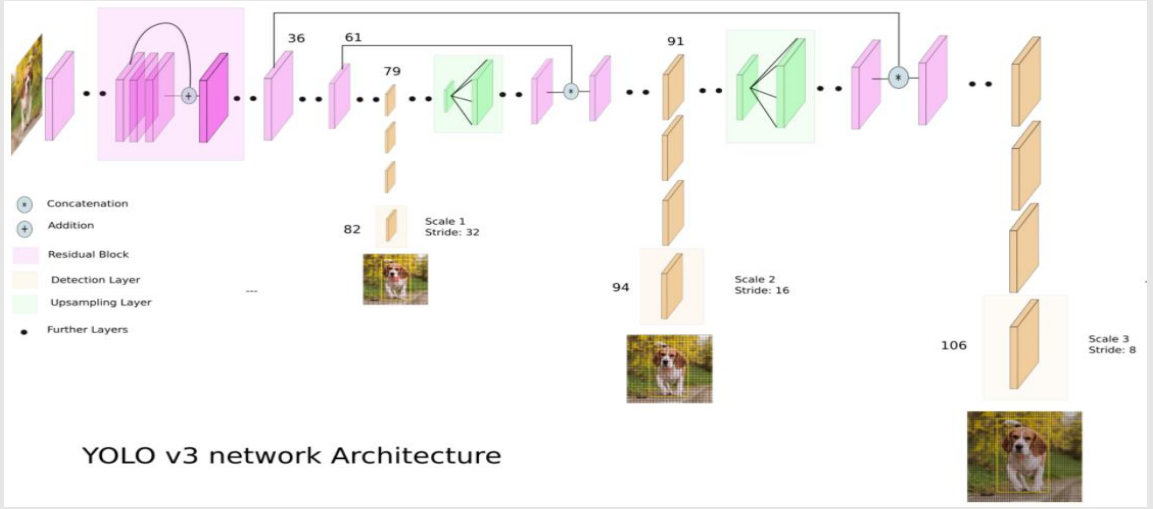
3 Boxes with 3 Scales



• 3가지 Scale별로 3개의 Anchor 박스가 할당됨.

• Grid의 각 Cell마다 3개의 박스에 대한 Coordinates, Objectness score, Class scores를 예측하므로
N x N x (3 x (4 + 1 + # of Classes))

• 크기가 다른 두 Feature map을 Concatenate하기 위해서 Up sampling 수행.
=> Meaningful semantic information + Finer-grained information

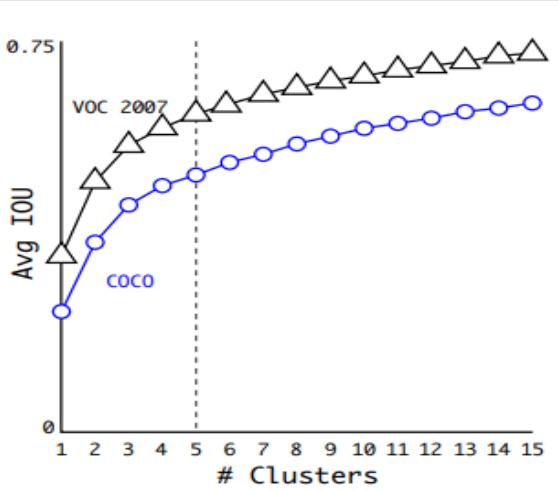


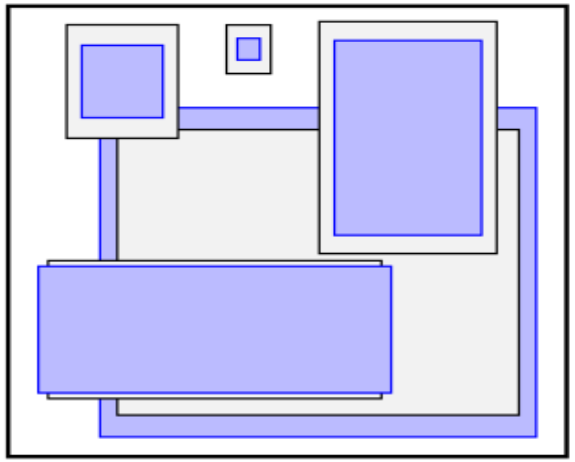
Anchor boxes

• 도메인 데이터의 Ground Truth 박스 정보에 대해서 K-means clustering을 수행하여 9개의 Anchor 박스를 추출해냄.

• 9개의 박스를 3개의 Scale에 균등하게 할당.
(10×13),(16×30),(33×23),
(30×61),(62×45),(59×119),
(116x90),(156 × 198),(373 × 326).

• # of Bounding Boxes
1. YOLOv1 98 boxes(7x7 grid cells, 2 boxes per cell, 448 x 448)
2. YOLOv2 845 boxes(13x13 grid cells, 5 anchor boxes)
3. YOLOv3 10,647 boxes(416 x 416)





Feature Extractor

Backbone	Top-1	Top-5	Bn Ops	BFLOP/s	FPS
Darknet-19 [15]	74.1	91.8	7.29	1246	**171**
ResNet-101[5]	77.1	93.7	19.7	1039	53
ResNet-152 [5]	**77.6**	**93.8**	29.4	1090	37
Darknet-53	77.2	**93.8**	18.7	**1457**	78

• 저자들이 새로 제안한 Darknet-53은 ResNet-152와 정확도는 비슷하면서 속도는 2배 이상 빠름.

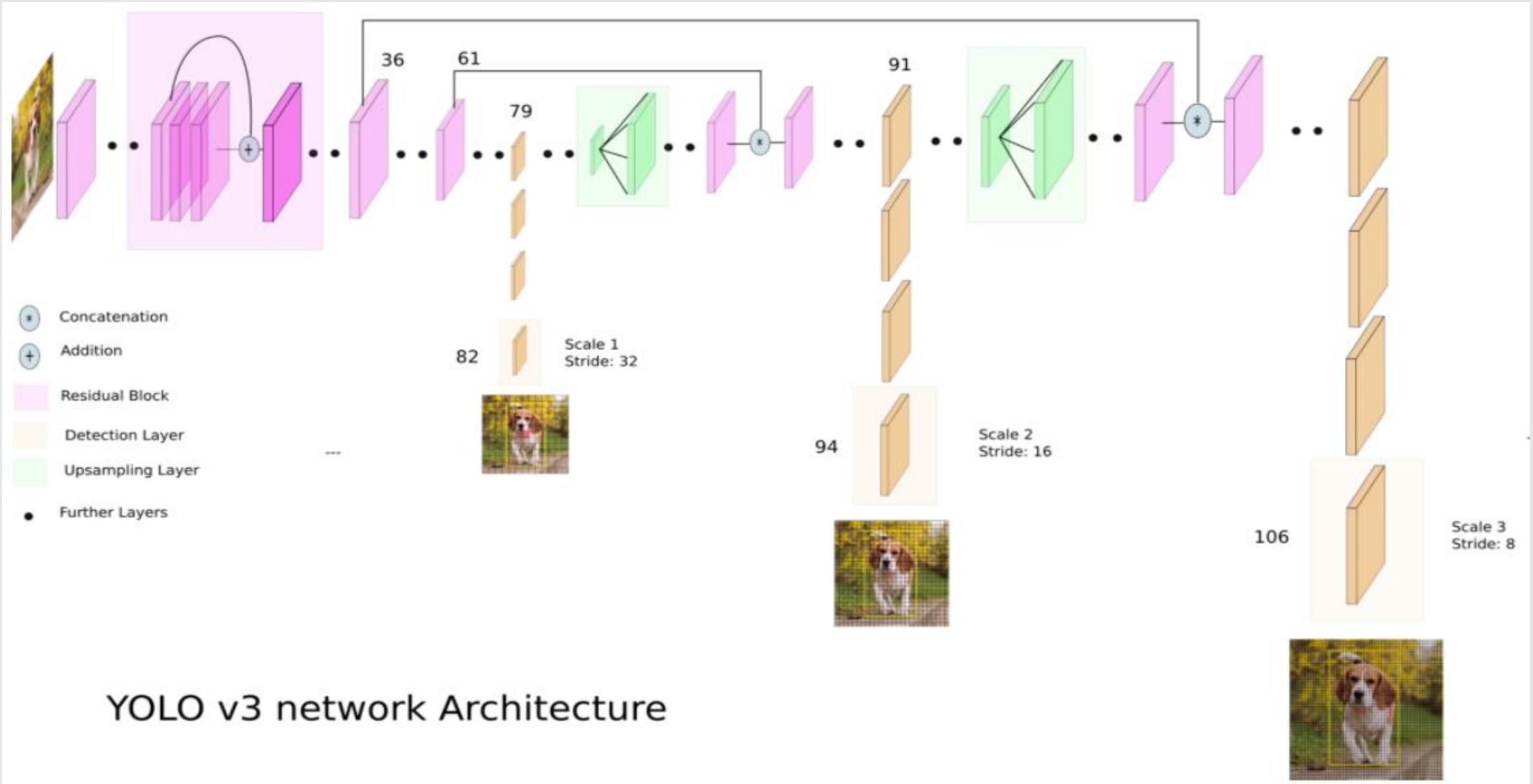
• ResNet-152보다 연산 량이 적으며 초당 연산 계산 량이 더 많음. 이는 Darknet-53이 GPU를 훨씬 더 잘 이용하도록 고안되었기 때문이라고 함.

• 저자들은 Crop, Warp를 적용하지 않은 전체 이미지로 훈련을 수행했고 Hard negative mining을 수행하지 적용하지 않음. (어떤 박스가 Ground Truth와 가장 많이 겹치지는 않지만 Threshold보다 많이 겹칠 때, 이 박스는 무시됨(저자들은 0.5 threshold 값 적용, 각 Ground Truth마다 1개의 Anchor 박스만 할당함 => Objectness score로 Background 부분을 버림.)

• Data augmentation, Batch normalization, Multi-scale training

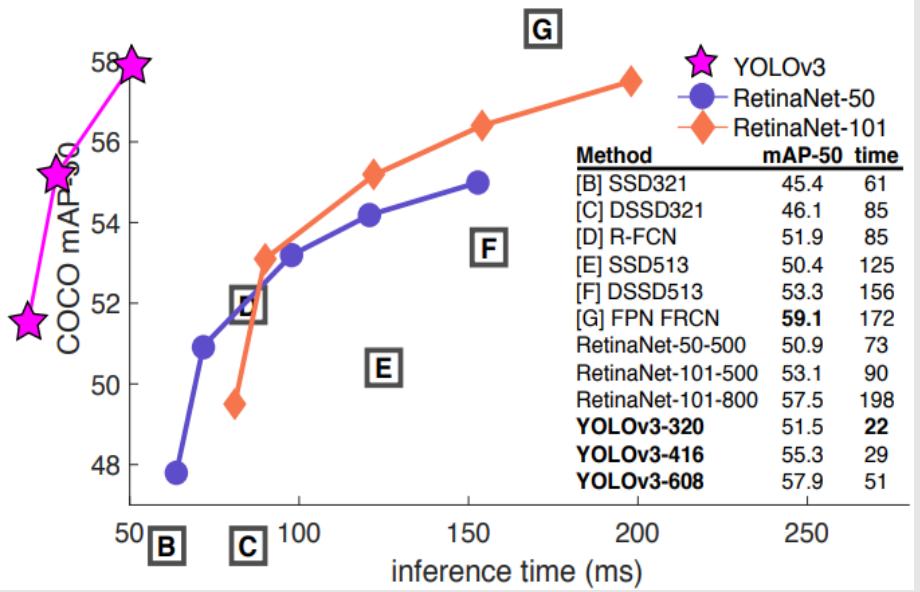
	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Feature Extractor



Experiments Result

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++ [5]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [8]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [6]	Inception-ResNet-v2 [21]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [20]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
One-stage methods							
YOLOv2 [15]	DarkNet-19 [15]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [11, 3]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [3]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet [9]	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet [9]	ResNeXt-101-FPN	**40.8**	**61.1**	**44.1**	**24.1**	**44.2**	51.2
YOLOv3 608 × 608	Darknet-53	33.0	57.9	34.4	18.3	35.4	41.9



- COCO Average mean AP metric으로 평가했을 때 정확도는 SSD 계열 모델과 비슷하지만 속도는 약 3배 더 빠름.

- COCO metric으로 평가하면 RetinaNet, Faster R-CNN 계열 모델들보다 정확도가 더 떨어짐.

- 그러나 COCO 이전의 Metric으로 평가하면(IOU= .5, AP_{50}) RetinaNet 모델과 정확도가 유사하면서 속도는 더 빠름. 저자들이 인정하길, YOLOv3는 완벽하게 GT 박스에 맞는 박스를 만들어 내는데는 어려움을 겪는다고 함. 그러나 Russakovsky 등의 연구를 인용하면서 사람조차 IOU of .3 ~ .5의 박스를 구별하는데 어려움을 겪는다고 하면서 .5씩 IOU 올려가며 모델 정확도를 측정하는 COCO Metric 방식에 의문을 제기함.

- 기존 YOLO과 비교했을 때, 비교적 작은 물체에 대한 탐지 성능(AP_S)는 더 좋아졌지만 중간, 큰 물체에 대한 탐지 성능은 오히려 떨어졌다고 함(AP_M, AP_L). 이에 대한 추가적인 연구 필요.

Things They Tried But Didn't Work

$$\hat{G}_x = P_w d_x(P) + P_x \quad (1)$$
$$\hat{G}_y = P_h d_y(P) + P_y \quad (2)$$
$$\hat{G}_w = P_w \exp(d_w(P)) \quad (3)$$
$$\hat{G}_h = P_h \exp(d_h(P)). \quad (4)$$

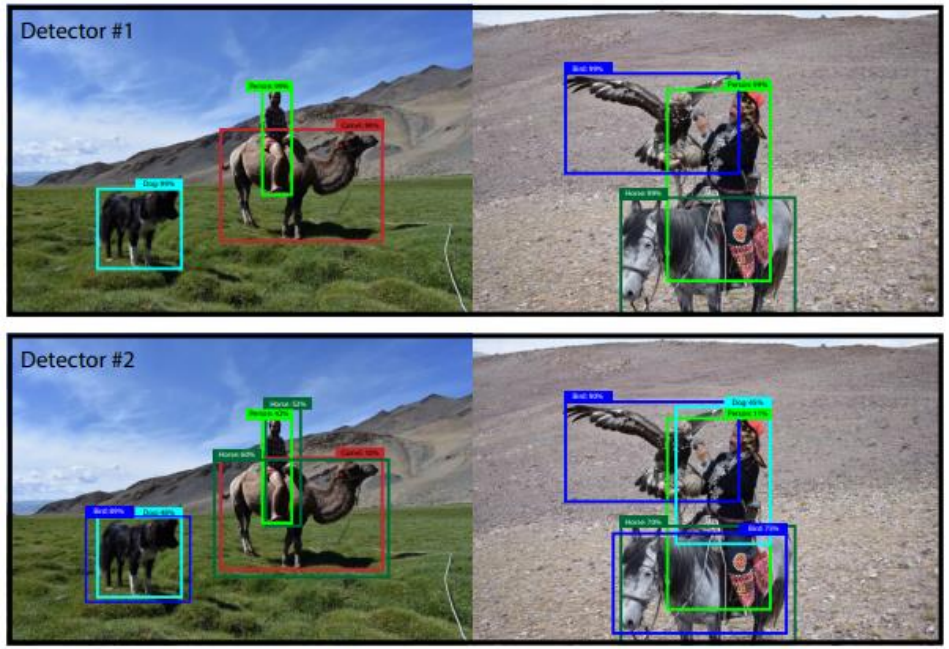
• Anchor box x, y offset predictions: 다른 알고리즘들과 같이 좌표에 대한 정보를 Prediction box의 Width와 Height의 비율로 구하는 방법을 적용.
=> 모델 성능의 안정성을 떨어트림.

• Linear x, y predictions instead of logistic: 선형 변환으로 좌표 값을 직접적으로 예측하는 방법을 적용.
⇒ 모델 mAP를 떨어트림.

• Focal loss: Focal loss를 적용.
⇒ 모델 mAP를 2 포인트 떨어트림.
⇒ 저자들이 추측하길 YOLOv3는 이미 Objectness prediction과 Conditional class prediction으로 Focal loss가 해결하려는 문제에 대해서 이미 Robust하기 때문이라고 함.

• Dual IOU thresholds and truth assignment: Faster R-CNN과 같이 두개의 IOU threshold를 적용하는 방법을 써봄. IOU가 .7이상이면 Positive, [.3 ~ .7]는 무시, .3 이하는 Negative.
⇒ 그다지 성능 개선이 일어나지 않음.

What This All Means



• YOLOv3가 Old metric으로는 성능이 좋다며, Russakovsky의 연구를 인용하며 COCO Metric에 대한 의문을 제기. 저자들에 의하면 COCO mAP로 평가했을 때 옆의 두 모델의 탐지 결과는 같다고 함.

• 기술 윤리적으로 기술의 진보가 군사 분야나 개인 정보 침해와 같이 악용되어서는 안된다고 함.