YOLOv3:
An Incremental Improvement

Joseph Redmon

박태우

Introduction

- Tech report

- 작은 변화만을 적용시켜 성능을 조금 향상시킴.

- 솔직히 다른 연구에 매진하느라 YOLO에 대해 작은 향상만 이뤘다고 함.

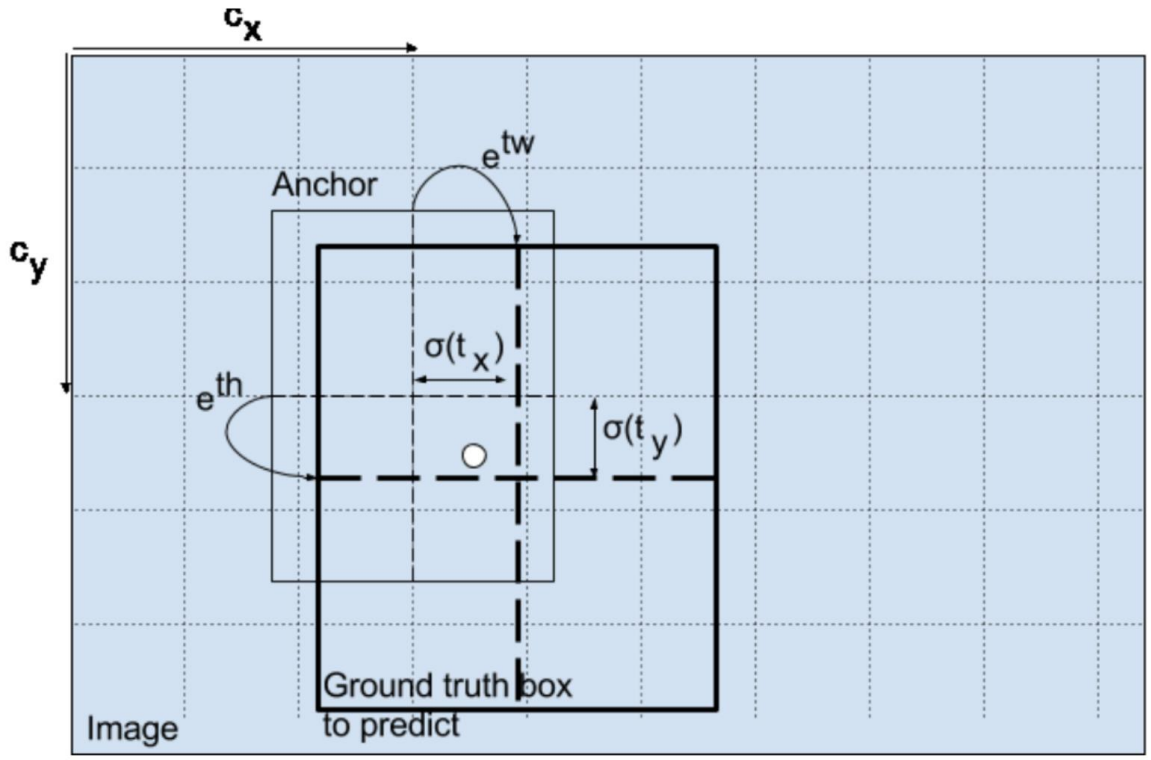
Bounding Box Prediction

- 기존과 같이 dimension cluster를 통해 anchor box를 이용.

- 네트워크는 bounding box의 x, y, w, h를 예측함.

- 다음과 같은 Squared error loss를 이용.

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



- 이는 기존 anchor box가 속한 grid cell에서 벗어나지 않도록 한다.

(tx, ty 값에 sigmoid function을 사용하기 때문에 변화 값이 0~1 사이의 값이 나옴)

Bounding Box Prediction

- Objectness score(confidence)를 기존과 달리 logistic regression을 이용함.

  Ground truth와 가장 overlap이 많이 되는 bounding box가 1의 값을 가짐.

- Ground truth 1개당 1개의 bounding box가 할당되도록 함.

  다른 detection의 경우 ground truth당 여러 개의 bounding box가 할당됨.

Class Prediction

- YOLO v3는 기존과 달리 multilabel classification을 적용시킴.

- 하지만 softmax는 많은 class 중 오직 1개만을 선택하는 문제가 있음.

- 그래서 softmax 대신에 logistic regression을 이용함.

- 즉 (person, woman)과 같이 multi classification을 가능하게 하기 위해 각 class에 대해 logistic regression 적용으로 0 ~ 1의 값을 부여하고 threshold를 사용함.

Prediction Across Scales

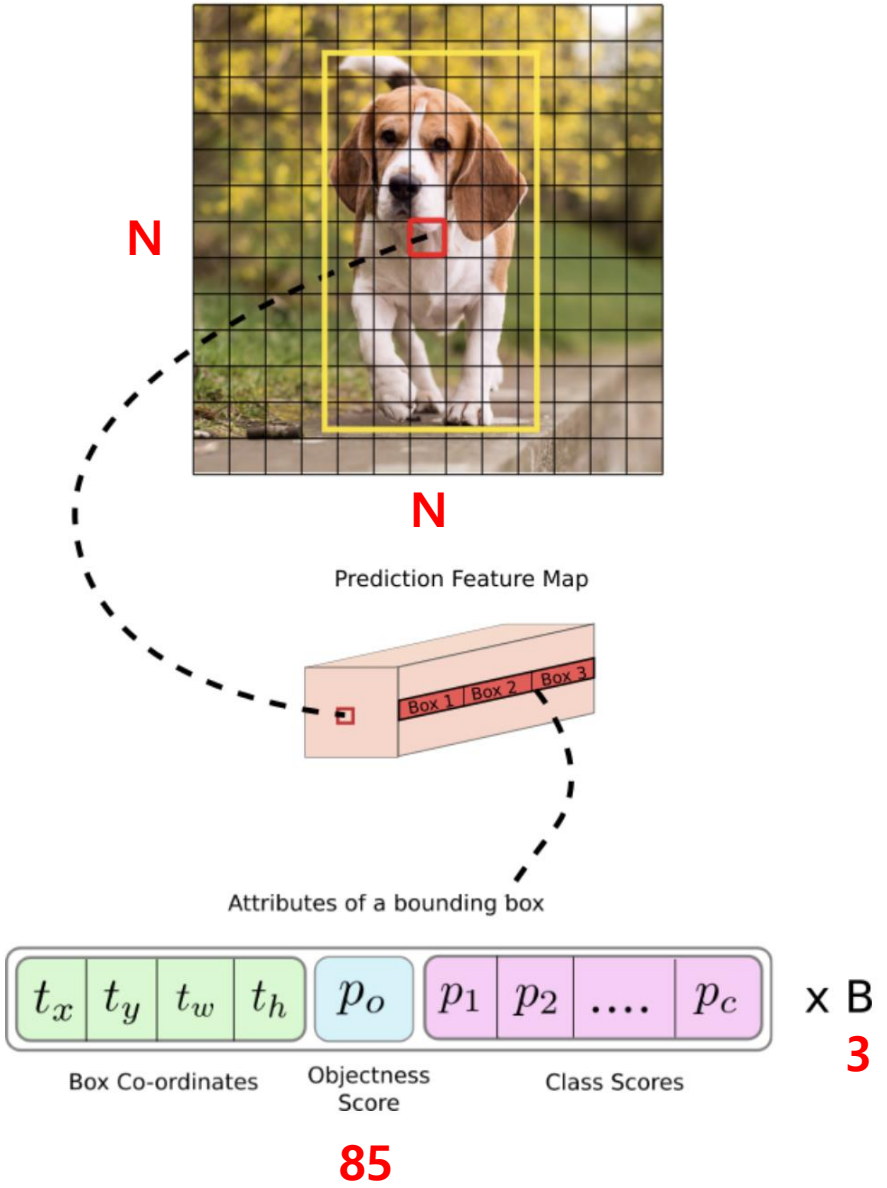
- YOLO v3는 3개의 스케일에 대해 bounding box들을 예측 (feature pyramid networks)

- 각 grid cell마다 3개의 bounding box를 예측함.

- 1개의 bounding box당 85개 값을 가짐

(4 bbox offset, 1 objectness pred, 80 class pred)

즉 1 scale에 N x N x [3 * (4+1+80)]의 값을 가짐.



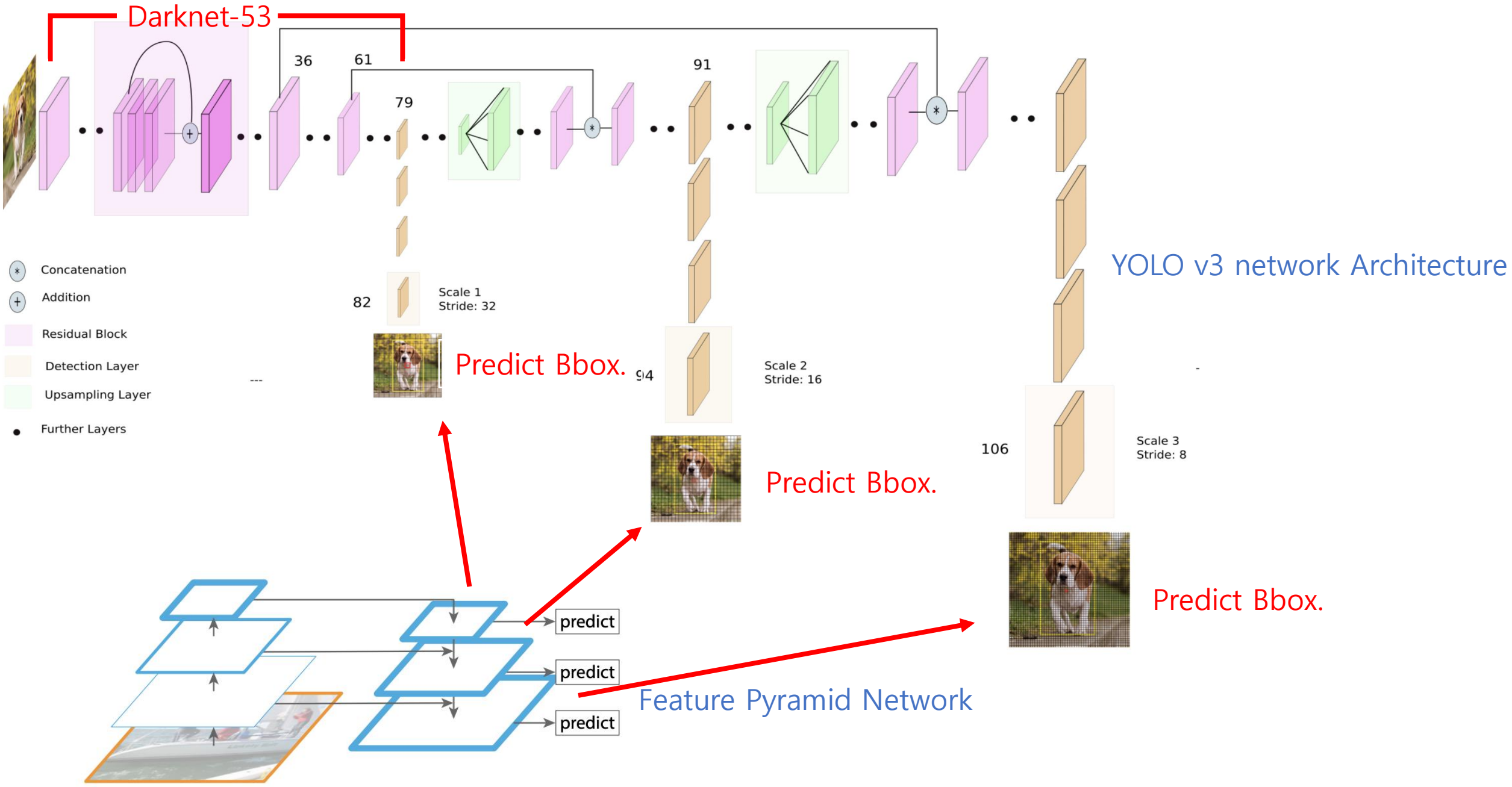
Prediction Across Scales

- 사용할 dataset을 clustering 해서 최적의 anchor box의 개수와 크기를 정함.

  (v3에서는 9개 선정)

- YOLO v1 : 7 x 7 grid cell x 2 bbox -> total 98 bounding box. (recall이 낮음)
- YOLO v2 : 13 x 13 grid cell x 5 bbox -> total 845 bounding box.
- YOLO v3 : ((52 x 52) + (26 x 26) + (13 x 13)) grid cell x 3 bbox -> 10647 bounding box.

Feature Extractor



Training

- Full images를 대상으로 트레이닝 했고 hard negative mining을 하지 않았음.
- 다른 detection의 경우 classification에 있어 background라는 class가 있음.
- Object가 있는 데이터보다 없는 데이터가 훨씬 많기 때문에 문제가 됨.
- 그래서 이런 문제를 해결하기 위해 hard negative를 사용.
- YOLO의 경우 objectness score를 thresholding함.

  (background class를 필요로 하지 않음)

How We Do

	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

정리

특징

- 본 논문은 v2 이후에 개선점을 통해 발표한 tech report.
- 성능적으로는 개선을 시켰지만, 속도는 오히려 조금 떨어짐.
- Bbox의 개수를 이전 버전들보다 상당히 많이 증가시켰고 multi scale prediction을 수행.

참고 링크

https://www.slideshare.net/JinwonLee9/pr207-yolov3-an-incremental-improvement

https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b