YOLO

YOLO9000 : Better, Faster, Stronger (2017)

2023.01.11

Introduction

YOLO9000 :

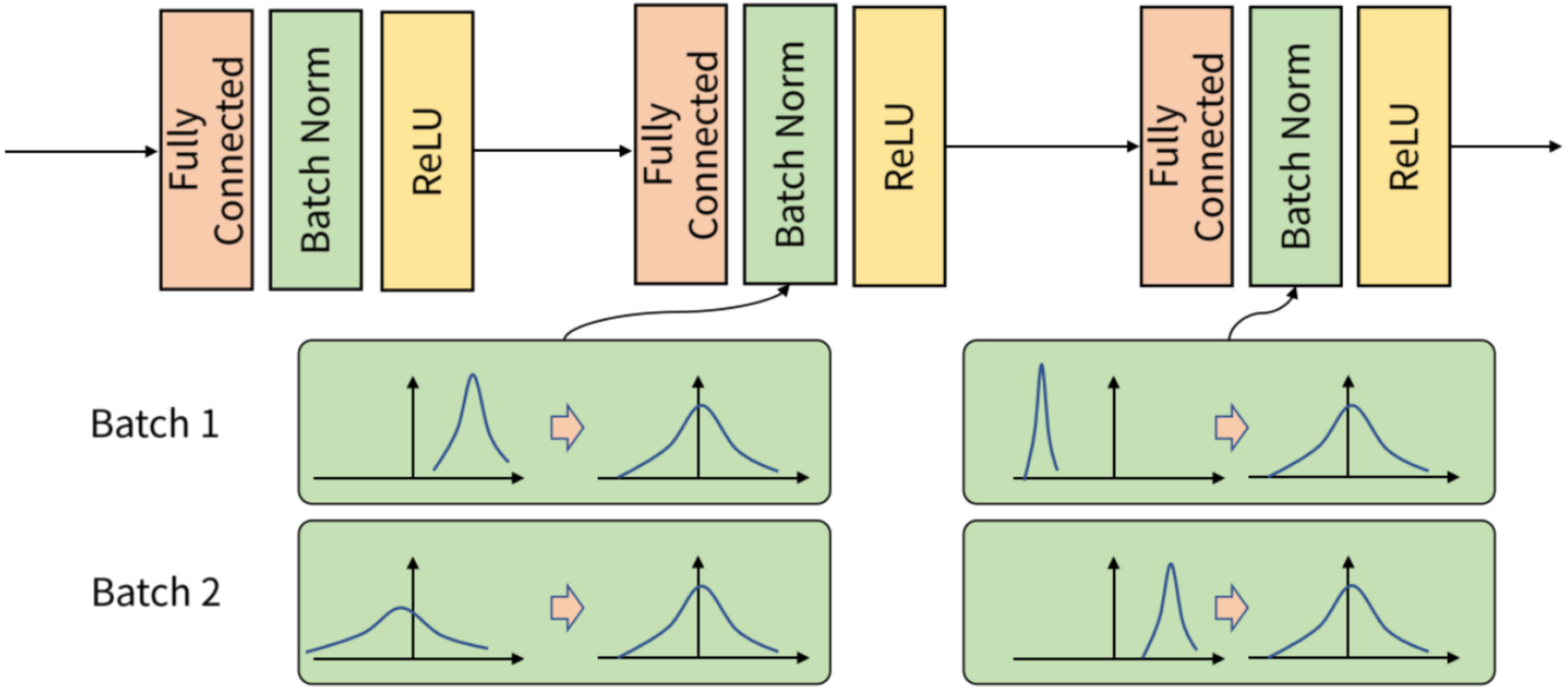
Better, Faster, Stronger

Accruacy, mAP 측면의 개선사항 속도 개선 더 많은, 다양한 클래스 예측

➡ YOLO v2

Better : Batch Normalization

- 학습 과정에서 각 배치 단위 별로 데이터가 다양한 분포를 가지더라도 각 배치별로 평균과 분산을 이용해 정규화
- 기존 YOLOv1 모델의 모든 convolution layer에 Batch Normalization 추가 -> mAP를 2%정도
- overfitting 없이 dropout 제거 가능



Better : High resolution classifier

- YOLO ➡ 224x224 해상도의 입력으로 classifier network를 학습 ➡ 448x448 해상도의 입력으로 detection network를 학습
- YOLOv2 ➡ 448x448 해상도의 입력으로 10epoch fine-tune을 진행 ➡448x448 해상도의 입력으로 detection network 학습을 진행
- 결과 ➡ mAP가 약 4% 정도 증가

Better : Convolutional with Anchor boxes

- YOLO ➡ 마지막 Fully-connected layer를 통해 bounding box들의 좌표를 직접 예측

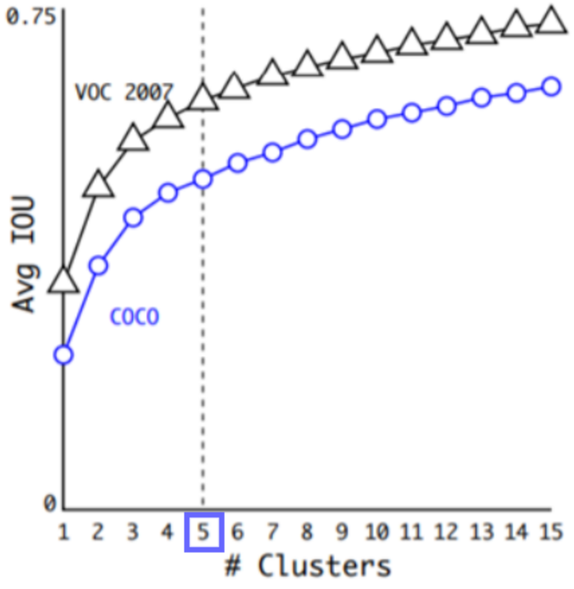
- YOLOv2 ➡ YOLO의 fully-connected layer를 제거 & bounding box 예측을 위해 anchor box를 이용
  - ➡ 입력의 해상도를 416x416으로 축소
  - ➡ why? 큰 객체가 보통 이미지의 중앙을 차지하는 경우가 많아서 가운데 cell이 하나인 feature map이 예측을 더 잘 수행
  - ➡ mAP (69.5% -> 69.2%) / recall (81% -> 88%)

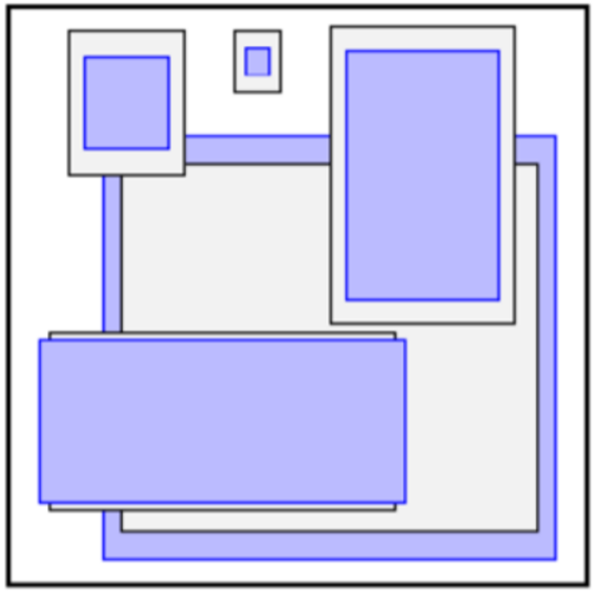
Better : Dimension clusters

- Anchor Box의 2가지 문제점

1) Anchor box의 개수와 비율은 직접 지정해야함 ➡ K-means Clustering ➡ IOU값을 고려한 distance metric

2) 학습 초기의 모델의 불안정성





$$d(\text{box}, \text{centroid}) = 1 - \text{IOU}(\text{box}, \text{centroid})$$

Box Generation	#	Avg IOU
Cluster SSE	5	58.7
Cluster IOU	5	61.0
Anchor Boxes [15]	9	60.9
Cluster IOU	9	67.2

Better : Dimension clusters

- Anchor Box의 2가지 문제점

1) Anchor box의 개수와 비율은 직접 지정해야함 ➡ K-means Clustering ➡ IOU값을 고려한 distance metric

2) 학습 초기 모델의 불안정성 ➡ anchor box 외부의 좌표로 예측하는 경우가 발생 ➡ bounding box마다 5개의 값을 예측 ➡ mAP 5% 증가

$$b_x = \sigma(t_x) + c_x$$

$$b_y = \sigma(t_y) + c_y$$

$$b_w = p_w e^{t_w}$$

$$b_h = p_h e^{t_h}$$

$$Pr(object) * IOU(b, object) = \sigma(t_o)$$

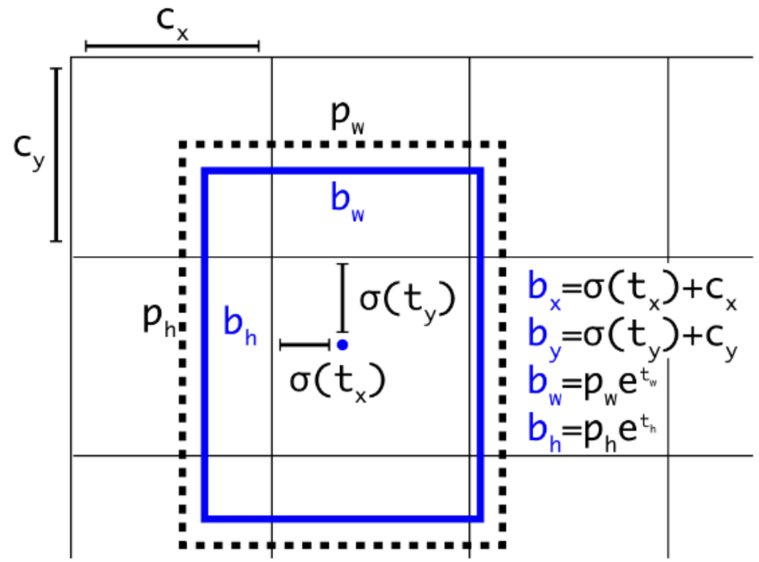
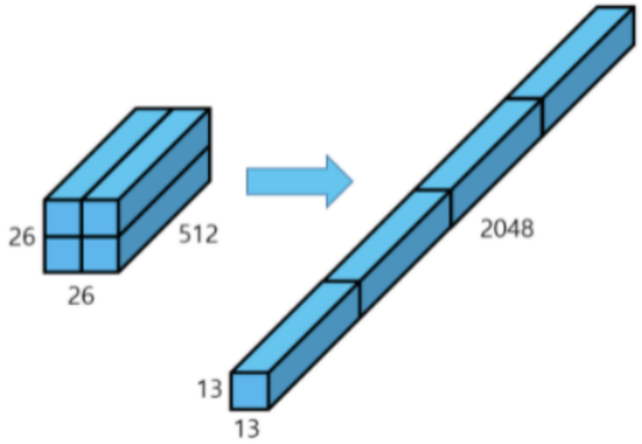


Figure 3: Bounding boxes with dimension priors and location prediction. We predict the width and height of the box as offsets from cluster centroids. We predict the center coordinates of the box relative to the location of filter application using a sigmoid function.

Better : Fine-Grained Features

- YOLO v2는 13x13 feature map은 큰 물체를 탐지하는데 충분할 수 있으나 작은 물체를 잘 탐지하지 못 할 수 있음
  - ➡ passthrough layer 사용
- 26x26x512 4등분, 13x13x2048로 만듦
  - ➡ 기존의 output인 13x13x1024 feature map과 concatenate를 수행
  - ➡ 13x13x3072 feature map 만듦
- mAP 대략 1% 증가



Better : Multi-Scale Training

- 전체 (fc layer 제거된) Convolutional Network로 이루어져있기 때문에 input image의 사이즈가 고정되지 않아도 된다.

  ➡ 10개 batch마다 입력 이미지의 크기 변경

- YOLO v2가 1/32배의 downsamplling을 진행하므로 학습시 32배수의 input size 이미지 사용
- high-resolution에서는 mAP가 높은 대신 FPS가 조금 떨어짐
- low-resolution에서는 mAP가 낮은 대신 FPS가 매우 빨라지게 된다.

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

Faster : Darknet-19

- Darknet-19라는 network를 새로 디자인하여 사용
- 총 19개의 Convolution layer와 5개의 maxpooling layer 구성
- vgg와 같은 3x3 filters 사용
- 마지막에 Global Average Pooling을 사용

  ➡ 학습 parameter수를 줄이기
- 3*3 convolution layer들 사이 중간중간에 1*1 convolution filter를 통해 channel을 줄이기
- 연산량이 줄어듦

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

Conclusion

- 기존 YOLO의 비해 mAP를 15.2% 끌어올림
- 기존 YOLO 모델에서 분류 정확도를 유지하면서 recall과 localization을 향상시키는 것에 집중
- 성능 향상을 위해 다양한 아이디어를 기존 모델에 적용

Reference

https://www.youtube.com/watch?v=6fdclSGgeio
(PR-023: YOLO9000: Better, Faster, Stronger)

https://www.youtube.com/watch?v=cNFpo7kDf-s&t=54s
(박경찬 - YOLO)

https://www.youtube.com/watch?v=vLdrI8NCFMs&t=1088s
([Paper Review] YOLO9000: Better, Faster, Stronger)

https://89douner.tistory.com/93
(10. YOLO V2)

감사합니다 :)

2023.01.11