

YOLO9000: Better, Faster, Stronger

(Joseph Redmon, Ali Farhadi, 2017, CVPR)

IVPG Lab Seminar 2022.04.06

세종대학교 지능기전공학부

18학번 장윤정

YOLO9000: Better, Faster, Stronger

YOLOv2

〈Better〉

- 아쉬웠던 YOLO의 성능을 개선해보자!
- 특히 low recall과 localization errors

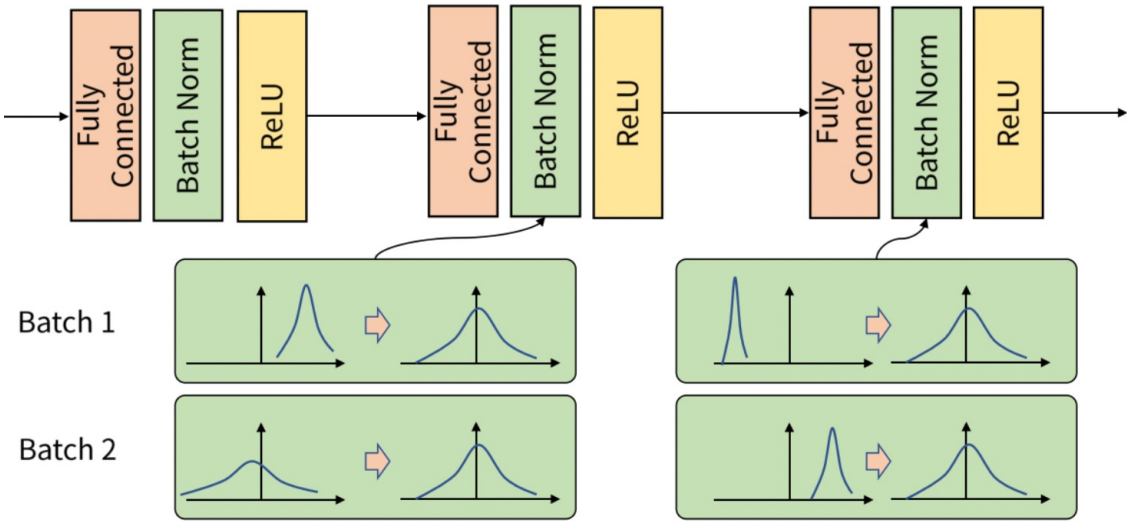
〈Faster〉

- YOLO의 장점이었던 속도를 더 개선해보자!

Better : Batch Normalization

Better
Batch Normalization
High Resolution Classifier
Convolutional With Anchor Boxes
Dimension Clusters
Direct location prediction
Fine-Grained Features
Multi-Scale Training

Faster
Darknet-19



Batch Normalization. Batch normalization leads to significant improvements in convergence while eliminating the need for other forms of regularization [7]. By adding batch normalization on all of the convolutional layers in YOLO we get more than 2% improvement in mAP. Batch normalization also helps regularize the model. With batch normalization we can remove dropout from the model without overfitting.

Batch Normalization

- 데이터를 배치 단위로 학습 할 때, 계층별로 데이터의 분포가 달라지는 현상을 방지하기 위해 사용
- 각 배치별로 평균과 분산을 이용해 정규화
- 학습 속도 증가, local minimum 방지

In the YOLOv2,

- 모든 컨볼루션 레이어에 Batch Normalization 추가 (mAP 약 2% 향상)
- Dropout 사용하지 않고 overfitting 방지 → Batch Normalization과 dropout을 같이 쓰는 경우보다 성능이 저하되었을까?



Better : High Resolution Classifier

Better
Batch Normalization
High Resolution Classifier
Convolutional With Anchor Boxes
Dimension Clusters
Direct location prediction
Fine-Grained Features
Multi-Scale Training

Faster
Darknet-19

In the YOLO,

- Classifier는 224x224(이미지넷 데이터셋) 학습 → Detection을 위해서는 448x448 사용
- 갑자기 커진 Input resolution이 성능에 영향을 미친다고 생각



In the YOLOv2,

- Classifier network를 10 epochs 동안 448x448로 fine-tuning → Detection을 위해서는 416x416 사용 (뒤에서 설명)
- Network에서 filter를 조정하여 high resolution input에서 잘 동작하도록 학습
- mAP 약 4% 향상

Better : Convolutional With Anchor Boxes

Better
Batch Normalization
High Resolution Classifier
Convolutional With Anchor Boxes
Dimension Clusters
Direct location prediction
Fine-Grained Features
Multi-Scale Training

Faster
Darknet-19

In the YOLO,

- Network 마지막에 Fully connected layer를 이용해서 bbox 직접 예측

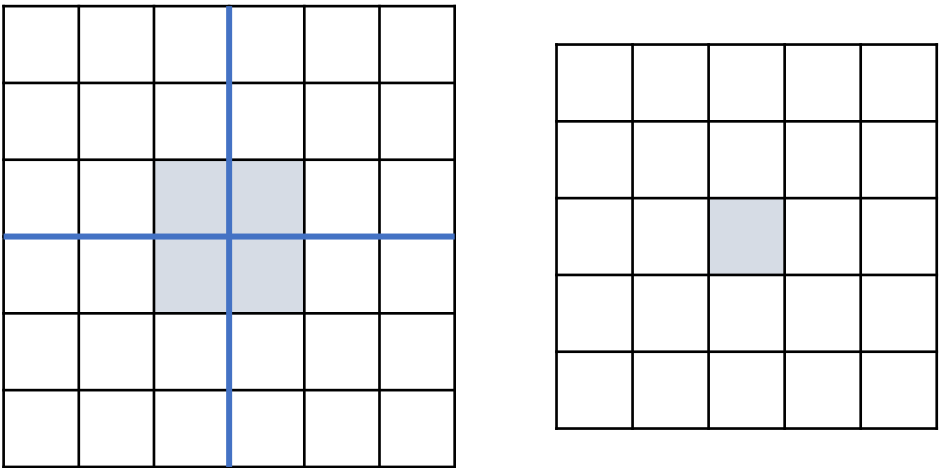
In the Faster R-CNN,

- 미리 aspect ratio를 정의해 놓은 hand-picked priors(anchor box) 9개를 어떻게 조정할지 학습

In the YOLOv2,

- Fully connected layer 제거하고, Convolutional layer만 사용 + anchor box 개념 도입
- Input image 416x416 사용 이유
 - network가 pooling(downsampling)을 5번해서 이미지가 1/32로 줄어드는데, 최종 feature map의 크기가 홀수가 되도록!
 - $448 / 32 = 14$, $416 / 32 = 13 \rightarrow$ 최종 feature map은 13x13
 - 큰 object들이 가운데 위치한 경우가 많아서 정가운데에 위치한 그리드 셀은 하나인 것이 좋음

e.g.)




Better : Convolutional With Anchor Boxes

Better
Batch Normalization
High Resolution Classifier
Convolutional With Anchor Boxes
Dimension Clusters
Direct location prediction
Fine-Grained Features
Multi-Scale Training

In the YOLO,

- Network 마지막에 Fully connected layer를 이용해서 bbox 직접 예측
- 각 그리드 셀에서 선택된 하나의 bbox만 class 예측

In the YOLOv2,

- Fully connected layer 제거하고, Convolution layer만 사용 + anchor box 개념 도입
- Input image 416x416 사용 이유
 - network가 pooling(downsampling)을 5번해서 이미지가 1/32로 줄어드는데, 최종 feature map의 크기가 홀수가 되도록!
 - $448 / 32 = 14$, $416 / 32 = 13 \rightarrow$ 최종 feature map은 13x13
 - 큰 object들이 가운데 위치한 경우가 많아서 정가운데에 위치한 그리드 셀은 하나인 것이 좋음
- 모든 anchor box마다 class와 objectness 예측 (anchor box : 13x13x5= 845)
- mAP는 소폭 감소(69.5→69.2), recall은 많이 개선(81%→88%) \longrightarrow 이 경우에, precision은 많이 안좋아졌다는 의미는 아닐까? 기존 YOLO의 장점이었던 적은 background errors가 유지 되었을까?
 - Precision = 옳게 검출한 box / 예측한 모든 box
 - Recall = 옳게 검출한 box / 모든 GT box
 - AP = precision-recall 그래프의 아래쪽 면적
 - mAP = class당 AP를 모두 합해서 class의 개수로 나눠준 것

Faster
Darknet-19

Better : Dimension Clusters

Better
Batch Normalization
High Resolution Classifier
Convolutional With Anchor Boxes
Dimension Clusters
Direct location prediction
Fine-Grained Features
Multi-Scale Training
Faster
Darknet-19

- Anchor box 사용으로 인한 두가지 문제,
1. hand-picked priors(anchor box) ✓
 2. 초반 iterations 동안의 위치 예측 불안정성(instability)

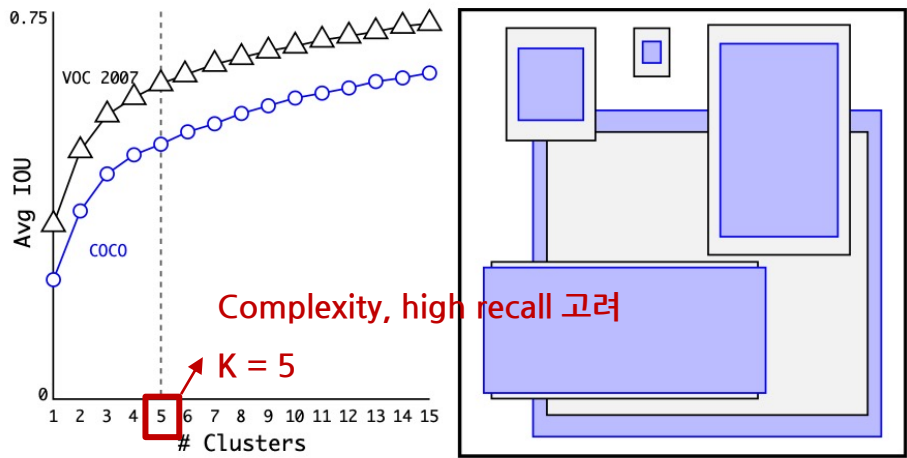


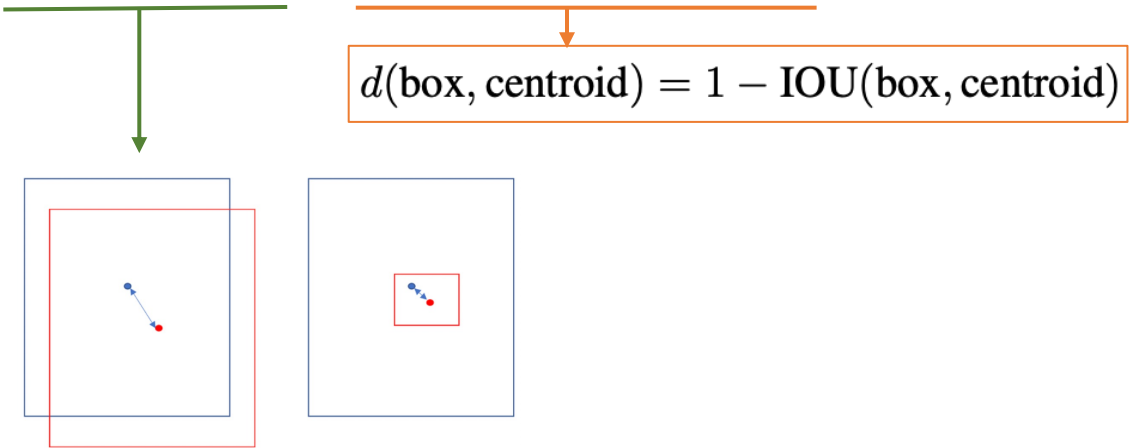
Figure 2: Clustering box dimensions on VOC and COCO. We

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

Table 1: Average IOU of boxes to closest priors on VOC 2007.

Anchor box 몇 개 사용할지 k-means clustering 사용

- Clustering(군집화) : 데이터 안에서 패턴과 구조를 발견하는 비지도 학습 방법
- K-means clustering 이란?
 - K : 데이터 세트에서 찾을 것으로 예상되는 그룹 수
 - Means : 각 데이터로부터 그 데이터가 속한 클러스터의 중심까지의 평균 거리 (이 값을 최소화 하는 것이 목표)
- Euclidean distance 대신 IOU를 이용한 새로운 거리 측정법 적용



Better : Dimension Clusters

Better
Batch Normalization
High Resolution Classifier
Convolutional With Anchor Boxes
Dimension Clusters
Direct location prediction
Fine-Grained Features
Multi-Scale Training
Faster
Darknet-19

- Anchor box 사용으로 인한 두가지 문제,
1. hand-picked priors(anchor box) ✓
 2. 초반 iterations 동안의 위치 예측 불안정성(instability)

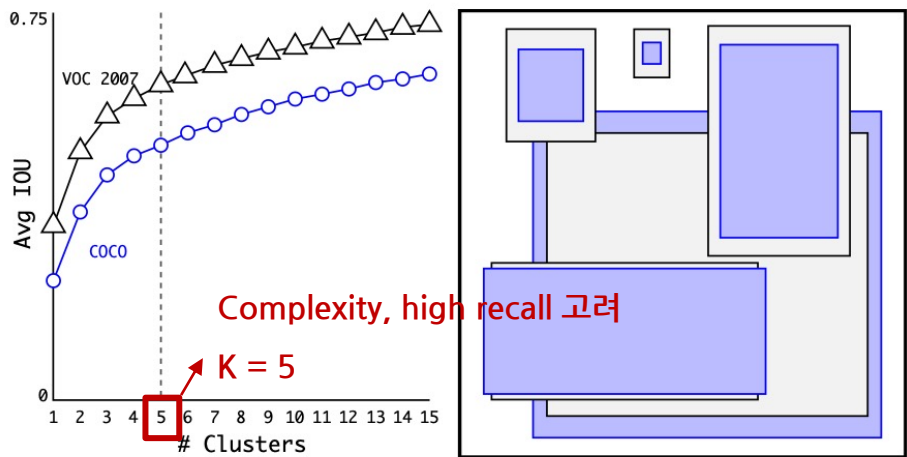


Figure 2: Clustering box dimensions on VOC and COCO. We

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

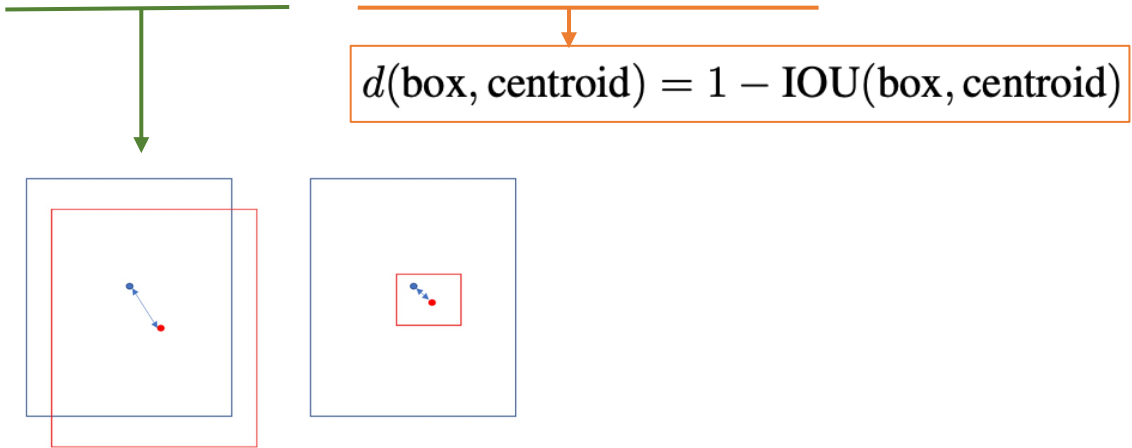
Table 1: Average IOU of boxes to closest priors on VOC 2007.



해당 데이터셋만을 위한 성능 향상 방법이 연구적으로 인정받는 방법일까?

Anchor box 몇 개 사용할지 k-means clustering 사용

- Clustering(군집화) : 데이터 안에서 패턴과 구조를 발견하는 비지도 학습 방법
- K-means clustering 이란?
 - K : 데이터 세트에서 찾을 것으로 예상되는 그룹 수
 - Means : 각 데이터로부터 그 데이터가 속한 클러스터의 중심까지의 평균 거리 (이 값을 최소화 하는 것이 목표)
- Euclidean distance 대신 IOU를 이용한 새로운 거리 측정법 적용



Better : Direct location prediction

Better
Batch Normalization
High Resolution Classifier
Convolutional With Anchor Boxes
Dimension Clusters
Direct location prediction
Fine-Grained Features
Multi-Scale Training

Anchor box 사용으로 인한 두가지 문제,

1. hand-picked priors(anchor box)
2. 초반 iterations 동안의 위치 예측 불안정성(instability) ✓

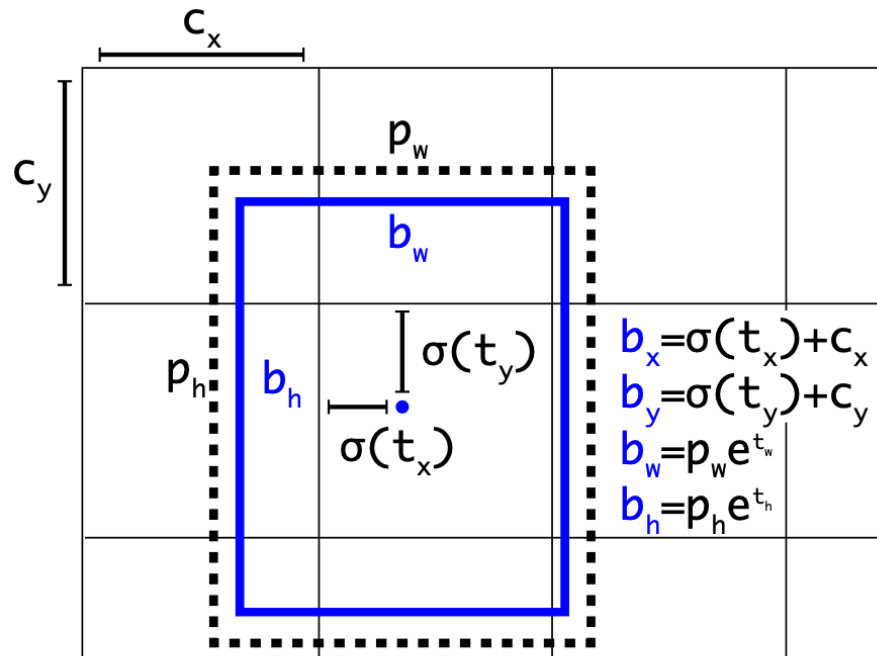


Figure 3: Bounding boxes with dimension priors and location

- C_x, C_y : 그리드 셀의 좌상단 끝 offset
- P_w, P_h : prior(anchor box)의 너비와 높이
- t_x, t_y, t_w, t_h : 예측해야 할 값들
- b_x, b_y, b_w, b_h : GT와의 IOU를 계산 할 최종 bbox의 offset

In Region proposal networks (e.g. Faster R-CNN),

- 학습 초기에 anchor box의 위치가 해당 그리드셀을 벗어나는 등 변동되어 학습이 불안정하게 될 가능성

In the YOLOv2,

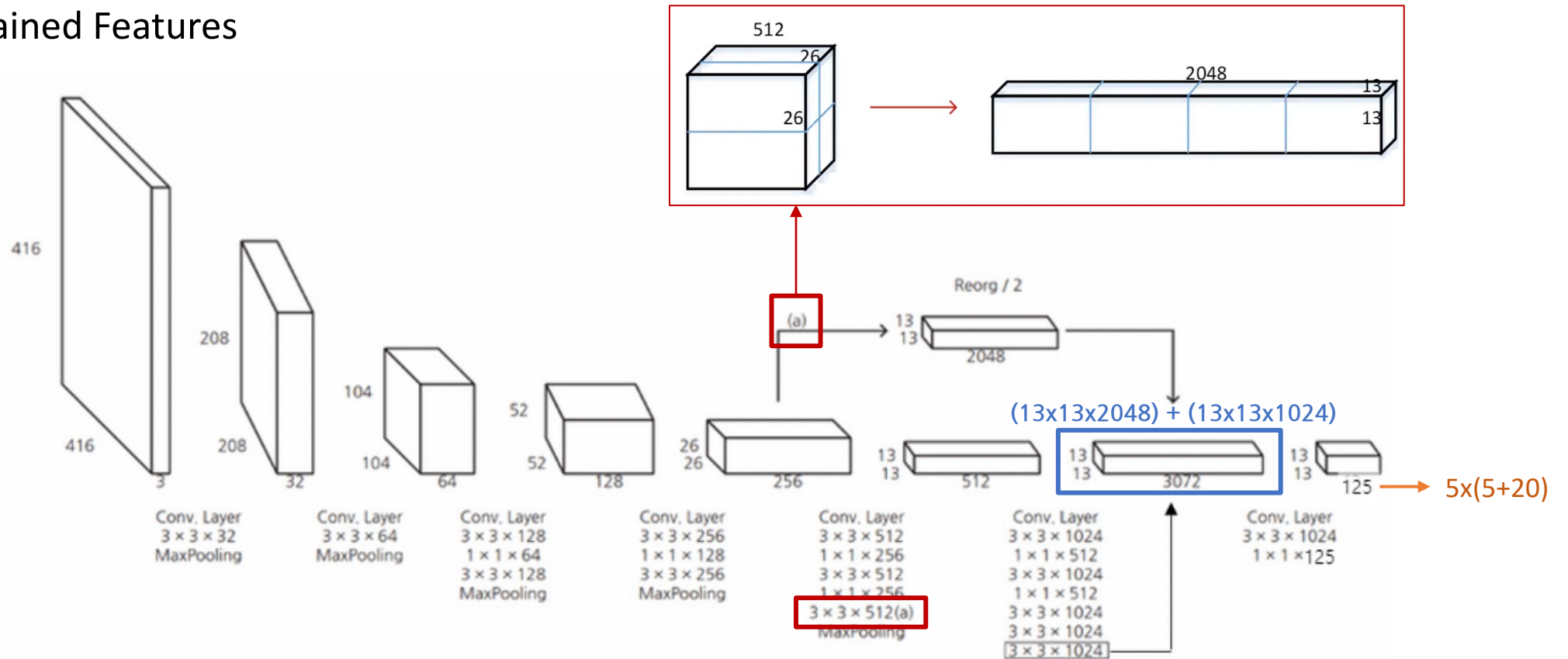
- 시그모이드를 사용해서, 해당 그리드셀에서 만든 bounding box는 중심이 항상 그리드셀 내부에 있도록 설정
- 즉, 위치(x, y)는 YOLO에서처럼 그리드셀 내부로 설정
- 종횡비(w,h)는 anchor box의 비율에서 시작할 수 있도록 설정
- 위치 제한은 네트워크를 안정화 (K-means clustering과 함께 약 mAP 5% 향상)

Faster
Darknet-19

Better : Fine-Grained Features

Better
Batch Normalization
High Resolution Classifier
Convolutional With Anchor Boxes
Dimension Clusters
Direct location prediction
Fine-Grained Features
Multi-Scale Training

Faster
Darknet-19



- Network 구조에서 앞쪽에서는 작은 물체를 찾고, 뒤쪽에서는 큰 물체를 찾는 방법을 택함 (e.g. SSD)
- YOLOv2는 passthrough layer 방법 사용
- 마지막 pooling 전 feature map인 26x26x256을 4등분해서 13x13x2048로 만듦 (higher resolution features)
- 뒤에 13x13x1024 feature map(low resolution features)과 concatenation해서 13x13x3072 feature map을 만듦
- mAP 약 1% 향상

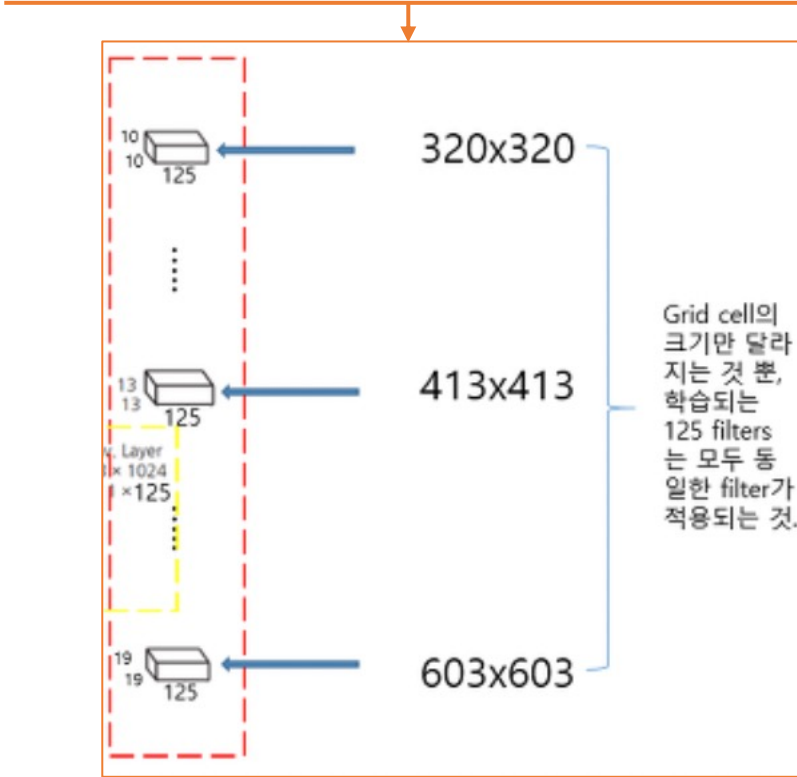
Better : Multi-Scale Training

Better
Batch Normalization
High Resolution Classifier
Convolutional With Anchor Boxes
Dimension Clusters
Direct location prediction
Fine-Grained Features
Multi-Scale Training
Faster
Darknet-19

In the YOLOv2,

- fully connected layer가 없고 전부 convolutional layer이기 때문에 input size 변화 가능 (마지막이 1x1x125 → 125 유지)
- 10 batchs 마다 새로운 크기의 이미지 학습
- Network가 1/32만큼 downsampling 하기 때문에, 이미지의 크기는 32의 배수인 {320x320, 352x352, ..., 608x608} 중에서 추출

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



- Low resolution : 굉장히 빠르지만 성능 낮아짐
- High resolution : SOTA급 성능 + 낮지만 real-time 가능한 FPS

Table 3: Detection frameworks on PASCAL VOC 2007.

Faster : Darknet-19

Better
Batch Normalization
High Resolution Classifier
Convolutional With Anchor Boxes
Dimension Clusters
Direct location prediction
Fine-Grained Features
Multi-Scale Training
Faster
Darknet-19

Type	Filters	Size/Stride	Output
Convolutional	32	3 × 3	224 × 224
Maxpool		2 × 2/2	112 × 112
Convolutional	64	3 × 3	112 × 112
Maxpool		2 × 2/2	56 × 56
Convolutional	128	3 × 3	56 × 56
Convolutional	64	1 × 1	56 × 56
Convolutional	128	3 × 3	56 × 56
Maxpool		2 × 2/2	28 × 28
Convolutional	256	3 × 3	28 × 28
Convolutional	128	1 × 1	28 × 28
Convolutional	256	3 × 3	28 × 28
Maxpool		2 × 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 × 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			

<VGG-16>
 : 224x224 resolution에서 306억 9천만 개의 부동소수점 연산 필요

<GoogLeNet>
 : 85억 2천만 연산으로 줄지만, 정확도 약간 낮음

- <Darknet-19>
- YOLOv2 개발자들이 직접 디자인해서 사용
 - VGG와 유사(3x3 filter 사용, pooling 후 채널 수 2배 등)
 - 마지막에 global average pooling 사용 (parameters 매우 감소)
 - 중간중간 1x1 컨볼루션으로 채널 수 줄임 (parameters 감소)
 - 하나의 이미지 처리에 55억 8천만 연산으로 줄어도 정확도 좋음

- In the YOLOv2,
- detection 수행 시 삭제
 - 뒤에 3x3 conv layer, passthrough layer, 1x1 conv layer 추가
- 최종 13x13x125의 feature map 생성 (125 = 5 x (5 + 20))



Conclusion

	YOLO								YOLOv2
batch norm?		✓	✓	✓	✓	✓	✓	✓	✓
hi-res classifier?			✓	✓	✓	✓	✓	✓	✓
convolutional?				✓	✓	✓	✓	✓	✓
anchor boxes?				✓	✓				
new network?					✓	✓	✓	✓	✓
dimension priors?						✓	✓	✓	✓
location prediction?						✓	✓	✓	✓
passthrough?							✓	✓	✓
multi-scale?								✓	✓
hi-res detector?									✓
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	78.6



- ✓ YOLOv2는 여러 방법을 적용하여 YOLO에 비해 15.2%의 mAP 향상
- ✓ 아쉬운점 : 최종적으로 YOLO의 단점이었던 localization errors는 얼마나 개선되었는지, 장점이었던 적은 background errors는 달라진 것이 없는지 등 세부적인 비교도 있었다면 더 좋았을 것 같음!

Reference

- <https://www.youtube.com/watch?v=6fdclSGgeio> (PR-023: YOLO9000: Better, Faster, Stronger)
- <https://www.youtube.com/watch?v=vLdrI8NCFMs> ([Paper Review] YOLO9000: Better, Faster, Stronger)
- <https://taeu.github.io/paper/deeplearning-paper-yolov2/> ([논문] YOLO9000: Better, Faster, Stronger 분석)
- <https://89douner.tistory.com/93> (10. YOLO V2)
- <https://gaussian37.github.io/dl-concept-batchnorm/> (배치 정규화(Batch Normalization))
- <https://hleecaster.com/ml-kmeans-clustering-concept/> (K-Means 클러스터링 쉽게 이해하기)

감사합니다

IVPG Lab Seminar 2022.04.06

세종대학교 지능기전공학부

18학번 장운정