You Only Look Once: Unified, Real-Time Object Detection

빅 데 이 터 경 영 통 계 전 공 2 0 1 9 2 7 8 4 윤 경 서

CONTENTS

CHAPTER 01

주제 및 YOLO의 장단점

본 논문 주제 및 개요

—

본 논문은 기존 DPM, R-CNN 방식의 object detecion방법의 단점이 실시간 object detection에는 적합하지 않다고 판단하여 YOLO라는 모델을 제안하고자 한다.

DPM : 이미지 전체를 거쳐 슬라이딩 윈도(sliding window) 방식으로 객체 검출을 하는 모델
R-CNN : region proposal 방식을 사용해 먼저 이미지에서 잠재적 bounding box 생성
-> bounding box 내에서 분류기를 실행하여 분류
-> 다른 물체를 기준으로 bounding box를 세분화하고 중복검출을 제거하면서 box 재평가

위 모델들운 복잡한 파이프라인을 가지고, 단계를 개별적으로 학습시키기 때문에 최적화도 어려우며, 느리다는 단점이 있음

-> 이미지 픽셀에서 bounding bod의 좌표와 class 확률로 객체 탐지를 위한 단일 회귀 문제로 재구성하여 진행되어 이미지를 한 번만 보고 어떤 물체가 있고, 어디에 있는지 예측할 수 있다는 의미의 You Only Look Once, YOLO를 제안

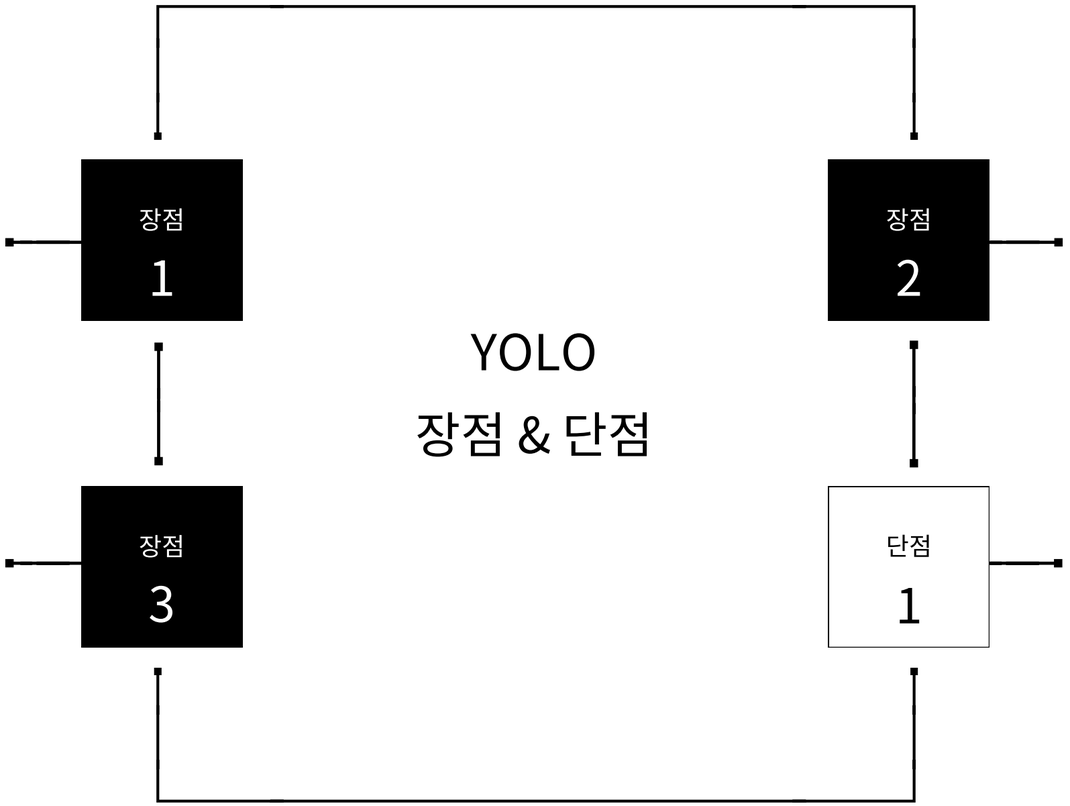
YOLO의 장단점

단순한 파이프라인으로 속도가 빠르다.

단순히 테스트 단계에서 새로운 이미지를
YOLO 신경망에 넣어주기만 하면 쉽게
객체 검출을 할 수 있다. 동영상을
실시간으로 처리할 수 있을만큼
속도가 빠르고, 높은 정확성을 갖는다.

객체의 일반적인 부분 학습

객체의 일반화 가능한 표현을 학습하기
때문에 훈련단계에서 보지 못한 새로운
이미지에 다른 모델들보다 더 강건하다.
즉, 검출 정확도가 더 높다.



예측할 때 전체 이미지를 본다.

전체 이미지를 보기 때문에
클래스 모양에 대한 정보뿐만 아니라
주변 정보까지 학습하여 처리하기 때문에
백그라운드 오류 수를 줄인다.

정확성이 떨어진다.

속도와 정확성은 trade-off 관계로
빠르게 객체를 검출하기 때문에
다른 모델에 비해 정확도가 떨어진다.

CHAPTER 02

YOLO

Unified Detection

객체 검출의 개별 요소를 단일 신경망으로 통합한 모델

각각의 bounding box를 예측하기 위해 이미지 전체의 특징을 활용하기 때문에 높은 정확성을 유지, 실시간 객체 검출 가능

01 input이미지를 S*S grid로 나누어 만약 어떤 객체의 중심이 특정 그리드 셀 안에 위치하면 그 그리드 셀이 해당 객체를 검출

02 각각의 그리드 셀은 B개의 bounding box와 그 bounding box에 대한 (confidence score, conditional class probabilities) 예측

confidence score : bounding box가 그 객체를 포함한다는 것을 얼마나 믿을만한지, 예측한 bounding box가 얼마나 정확한지 나타냄
conditional class probabilities : 그리드 셀안에 객체가 있다는 조건 하에 그 객체가 어떤 class에 해당하는지에 대한 조건부 확률

03 테스트 단계에서 bounding box에 특정 class의 객체가 나타날 확률과
예측된 bounding box가 그 class 객체에 얼마나 잘 들어맞는지를 나타내는 class-specific confidence score 계산

class-specific confidence score : class probability * 개별 bounding box의 confidence score

Network Design

하나의 CNN 구조로 디자인 되어있음

Convolution layer 24개	+	Fully connected layer 2개
이미지 특성 추출		class 확률 bounding box 좌표

7 x 7 x 30의 예측 텐서로 최종 output을 갖는 모델

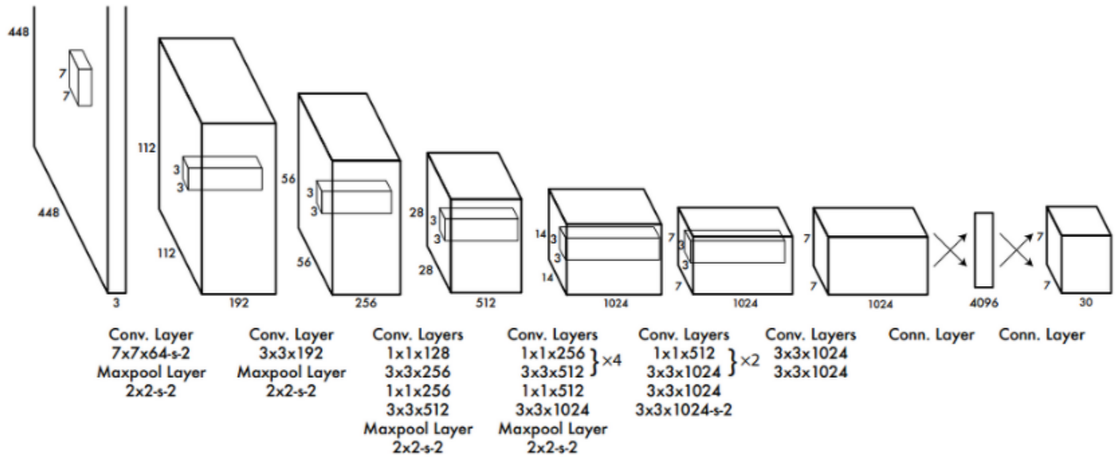


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

Training

데이터 셋 : 1000개의 클래스를 갖는 ImageNet 데이터셋

- 위 데이터 셋을 사용하여 YOLO의 앞단 20개 convolution layer를 pretrain 시킴
    -> ImageNet 2012 검증 데이터셋 88% 정확도 기록

- ImageNet은 분류를 위한 데이터셋이기 때문에 사전 훈련된 분류 모델을 객체 검출 모델로 바꿔주어야 함
    -> 20개의 convolution layer 뒤에 4개의 convolution layer와 2개의 Fully connected layer 추가

- YOLO 신경망의 마지막 계층에는 선형 활성화 함수, 나머지 모든 계층에는 leaky ReLU를 적용

※ 구조상의 문제해결

1) YOLO의 loss : SSE(sum squared error)

- SSE를 최적화 시키는 방법으로 localization loss와 classification loss 가중치를 동일하게 취급하는데
  이 두 loss의 가중치를 동일하게 두고 학습시키는 것은 좋지 않음

  localization loss : bounding box의 위치를 얼마나 잘 예측했는지

  classification loss : 클래스를 얼마나 잘 예측했는지

-> localization loss와 classification loss 중 localization loss의 가중치 증가

2) SSE는 큰 bounding box와 작은 bounding box에 대해 모두 동일한 가중치로 loss를 계산

- 작은 객체는 조금만 움직여도 bounding box를 벗어나게 됨

-> bounding box의 widht와 hegith에 square root를 취해준 값을 loss function으로 사용

3) 이미지 내에 배경 영역이 더 많기 때문에 대부분의 그리드 셀에 객체가 존재하지 않음

- 그리드 셀에 객체가 없기 때문에 대부분의 그리드 셀의 confidence score=0이 되고 이는 모델의 불균형을 초래

-> 객체가 없는 그리드 셀보다 객체가 존재하는 그리드 셀의 confidence loss의 가중치를 증가

연구진

파스칼 VOC 2007, 2012 훈련 및 검증 데이터 셋을 활용하여 135 epochs로 YOLO 모델 훈련

batch size=64, momentum=0.9, decay=0.0005로 설정

학습률(learning rate)을 0.001에서 0.01로 천천히 상승

처음부터 높은 learning rate로 훈련시키면 gradient explosion이 발생하기 때문에 처음에는 작은 값부터 시작

이후 75 epoch 동안 0.01, 30 epoch 동안 0.001, 마지막 30 epoch 동안 0.0001로 learning rate 설정

과적합을 막기 위해 dropout과 data augmentation을 적용

drop out 비율 : 0.5

data augmentation : 원본 이미지의 20%까지 random scaling과 random translation을 적용

추론

- OLO는 하나의 신경망 계산만 필요하기 때문에 테스트 단계에서는 굉장히 빠름
- 파스칼 VOC 데이터 셋에 대해서 YOLO는 한 이미지 당 98개의 bounding box를 예측해주고,
  그 bounding box마다 클래스 확률을 구해줌

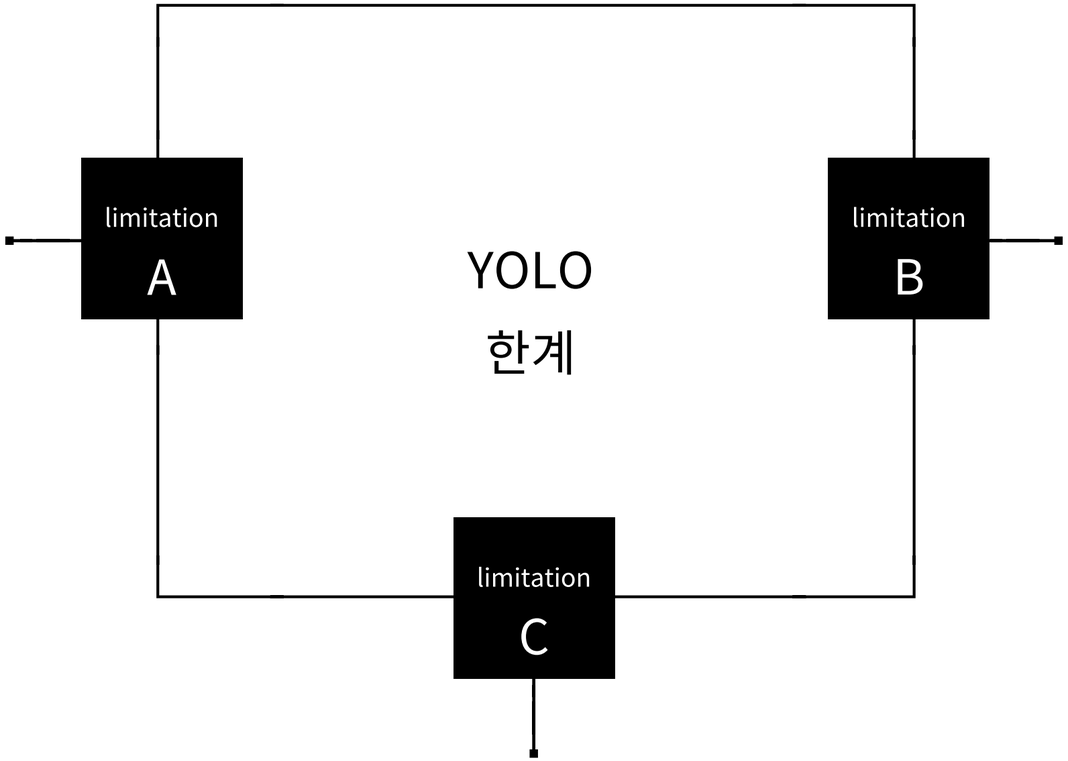
※ 이때 YOLO 그리드 디자인의 다중 검출 문제

- 하나의 객체를 여러 그리드 셀이 동시에 검출하는 경우가 있음.
- 하나의 객체가 정확히 하나의 그리드 셀에만 존재하는 경우에는 문제가 없지만 객체의 크기,
  객체의 위치에 따라 그 객체에 대한 bounding box가 여러 개 생기는 문제가 발생할 수 있음

비 최대 억제(non-maximal suppression)라는 방법을 통해 개선

-> YOLO는 비 최대 억제를 통해 mAP를 2~3%가량 향상

Limitations of YOLO



공간적 제약 문제 야기

하나의 그리드 셀은 오직 하나의 객체만
검출하므로 하나의 그리드 셀에 두 개
이상의 객체가 붙어있을 경우
이를 잘 검출하지 못하는 문제

새로운 종횡비를 만날 시 고전

YOLO 모델은 데이터로부터 bounding box를
예측하는 것을 학습하기 때문에 훈련 단계에서
학습하지 못했던 새로운 종횡비를 만날 시 고전함

부정확한 localization 문제

작은 bounding box가 위치 변화에 있어서
성능의 영향을 많이 받는데
큰 bounding box와 작은 bounding box의
loss에 대해 동일한 가중치를 둠

CHAPTER 03

다른 실시간 객체 검출 모델과 비교

Comparison to Other Real-Time Systems

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

Table 1: Real-Time Systems on PASCAL VOC 2007. Comparing the performance and speed of fast detectors. Fast YOLO is the fastest detector on record for PASCAL VOC detection and is still twice as accurate as any other real-time detector. YOLO is 10 mAP more accurate than the fast version while still well above real-time in speed.

좌측 표는 각종 객체 검출 모델 별 정확도(mAP)와 속도(FPS)를 보여줌

실시간 검출로 사용하기 위해서는 FPS가 30 이상이어야 함
(FPS = 30: 1초에 30 프레임의 영상을 처리)

정확도는 Fast R-CNN과 Faster R-CNN VGG-16이 가장 높지만
FPS는 너무 낮아 실시간 객체 검출 모델로 사용할 수는 없음

-> 정확도도 적당히 높고 속도도 빠른 모델은 YOLO 계열

VOC 2007 Error Analysis : 파스칼 VOC 2007 데이터 셋에 대해 YOLO와 Fast R-CNN의 성능을 비교

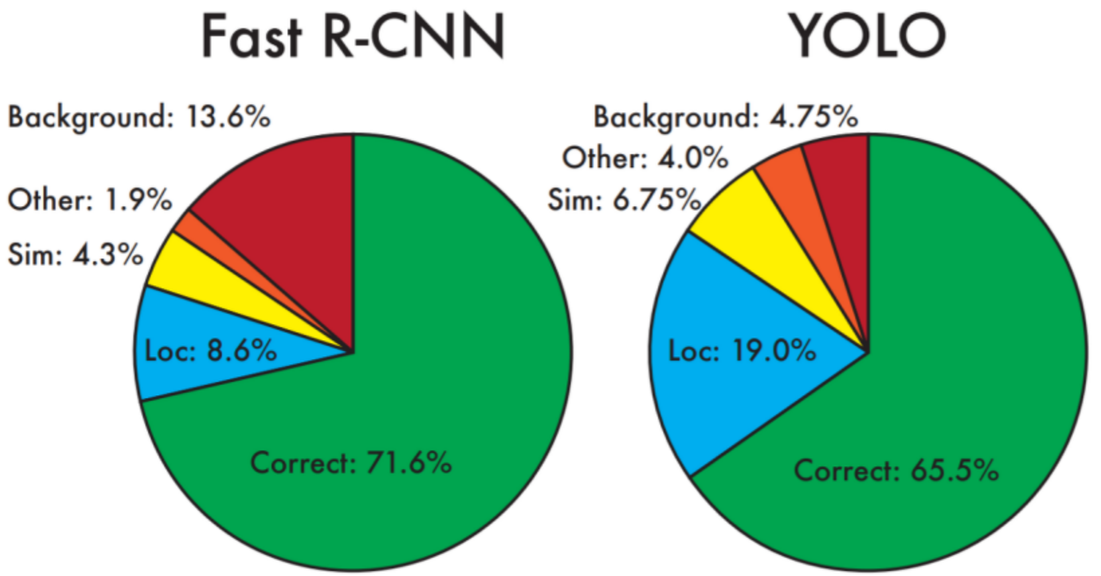


Figure 4: Error Analysis: Fast R-CNN vs. YOLO These charts show the percentage of localization and background errors in the top N detections for various categories (N = # objects in that category).

Correct : class가 정확하며 IOU > 0.5 인 경우

Localization : class가 정확하고, 0.1 < IOU < 0.5 인 경우

Similar : class가 유사하고 IOU > 0.1 인 경우

Other : class는 틀렸으나, IOU > 0.1 인 경우

Background : 어떤 Object라도 IOU < 0.1 인 경우

YOLO는 localization error 가 상대적으로 큼

Fast R-CNN은 background error가 상대적으로 큼

Combining Fast R-CNN and YOLO

	mAP	Combined	Gain
Fast R-CNN	71.8	-	-
Fast R-CNN (2007 data)	**66.9**	72.4	.6
Fast R-CNN (VGG-M)	59.2	72.4	.6
Fast R-CNN (CaffeNet)	57.1	72.1	.3
YOLO	63.4	**75.0**	**3.2**

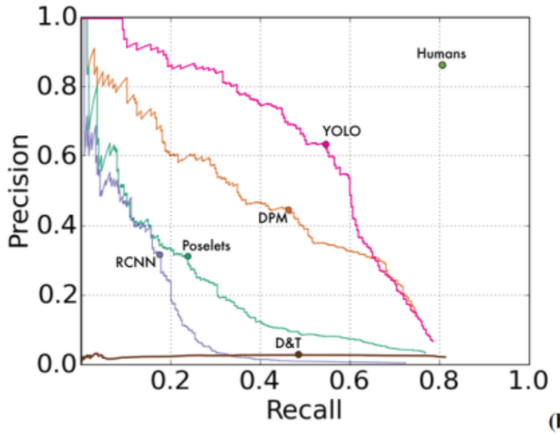
Table 2: Model combination experiments on VOC 2007. We examine the effect of combining various models with the best version of Fast R-CNN. Other versions of Fast R-CNN provide only a small benefit while YOLO provides a significant performance boost.

Fast R-CNN에 YOLO를 결합하여 background error를 줄인다면 굉장히 높은 성능을 낼 수 있을 것이며,
R-CNN이 예측한 bounding box와 YOLO가 예측한 bounding box가 유사하다면 두 bounding box가 겹치는 부분을 bounding box로 잡으면 더 정확한 bounding box를 찾을 수 있을 것임

- 파스칼 VOC 2007 데이터 셋에 대해 가장 성능이 좋은 Fast R-CNN 모델은 71.8%의 mAP를 기록.
- Fast R-CNN과 YOLO를 결합하면 mAP가 3.2% 올라 75.0%가 됨.
- Fast R-CNN과 다른 모델과도 앙상블을 해봤지만 mAP 향상은 0.3%, 0.6%로 미미

Generalizability: Person Detection in Artwork

- 객체 검출 연구를 위해 사용하는 데이터 셋은 훈련 데이터 셋과 테스트 데이터 셋이 동일한 분포를 지님
- 하지만 실제 이미지 데이터는 훈련 데이터 셋과 테스트 데이터 셋의 분포가 다를 수 있기 때문에
  훈련 데이터 셋에서 보지 못한 새로운 데이터 셋을 활용하여 테스트를 함
    -> 피카소 데이터 셋과 일반 예술 작품을 사용 (실제 이미지 학습, 예술 작품으로 테스트)



(a) Picasso Dataset precision-recall curves.

	VOC 2007	Picasso		People-Art
	AP	AP	Best F_1	AP
YOLO	**59.2**	**53.3**	**0.590**	**45**
R-CNN	54.2	10.4	0.226	26
DPM	43.2	37.8	0.458	32
Poselets [2]	36.5	17.8	0.271	
D&T [4]	-	1.9	0.051	

(b) Quantitative results on the VOC 2007, Picasso, and People-Art Datasets. The Picasso Dataset evaluates on both AP and best F_1 score.

Figure 5: Generalization results on Picasso and People-Art datasets.

- R-CNN은 VOC 2007에서는 높은 정확도를 보이지만
  예술작품에 대해서는 굉장히 낮은 정확도를 보임
- DPM은 예술 작품에 대해서도 정확도가 크게 떨어지지는
  않았지만 VOC 2007에서의 정확도도 그리 높은 편은 아니었음
- YOLO는 VOC 2007에서도 가장 높은 정확도를 보였고,
  예술 작품에 대해서도 정확도가 크게 떨어지지 않음

-> YOLO는 훈련 단계에서 접하지 못한
  새로운 이미지도 잘 검출

CHAPTER 04

결론 및 기여

결론 및 기여

- YOLO는 단순하면서도 빠르고 정확함
- YOLO는 훈련 단계에서 보지 못한 새로운 이미지에 대해서도 객체를 잘 검출함
- 새로운 이미지에 대해서도 강건함
- 보정하고자 하는 객체로 다양한 이미지가 들어올 수 있을 것이라 예상되는데
  이런 부분에 있어서 YOLO를 사용하는 것이 적합해보임
- 속도가 빨라 어플리케이션에 활용할만한 가치가 있어보임

THANK