YOLO

You Only Look Once :
Unified, Real-Time Object Detection (2016)

2023.01.04

01

Introduction

Introduction

YOLO의 장점

매우 빠르며, 탐지를 회귀 문제로 간주하기 때문에 복잡한 파이프라인이 불필요
다른 실시간 시스템의 평균 정밀도(precision)의 두 배 이상을 달성
sliding window나 region proposal 기반 방법과 달리, 이미지 전체를 보기에 백그라운드 오류 감소
객체의 특정적 표현이 아닌 일반화된 표현을 학습하기 때문에 일반화 능력 탁월

YOLO의 단점

최첨단 탐지 방법에 비해 비교적 낮은 정확도
이미지에서 객체를 빠르게 식별 가능하나, 일부 객체 특히 작은 객체의 위치를 정확하게 파악 어려움

02

Unified Detection

Unified Detection

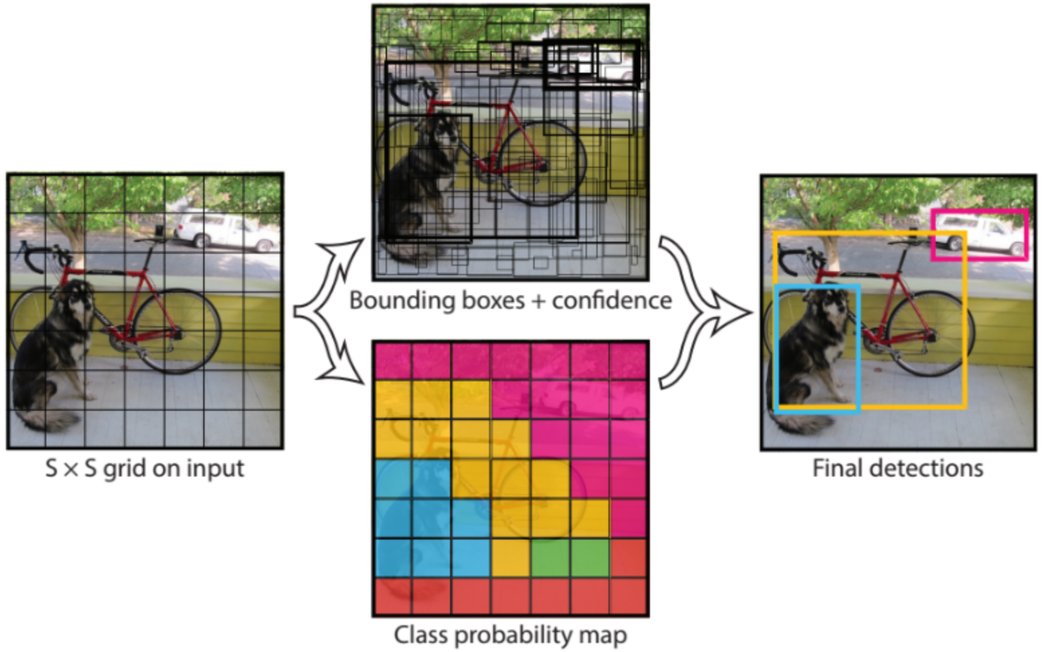

Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

1) 입력 이미지(input images)를 S x S 그리드(S x S grid)로 나눈다.

2) 각각의 그리드 셀(grid cell)은 B개의 bounding box와 그 bounding box에 대한 confidence score를 예측한다.

3) class-specific confidence score는 bounding box에 특정 클래스(class) 객체가 나타날 확률과 예측된 bounding box가 그 클래스(class) 객체에 얼마나 잘 들어맞는지를 나타낸다.

03

Network Design

Network Design

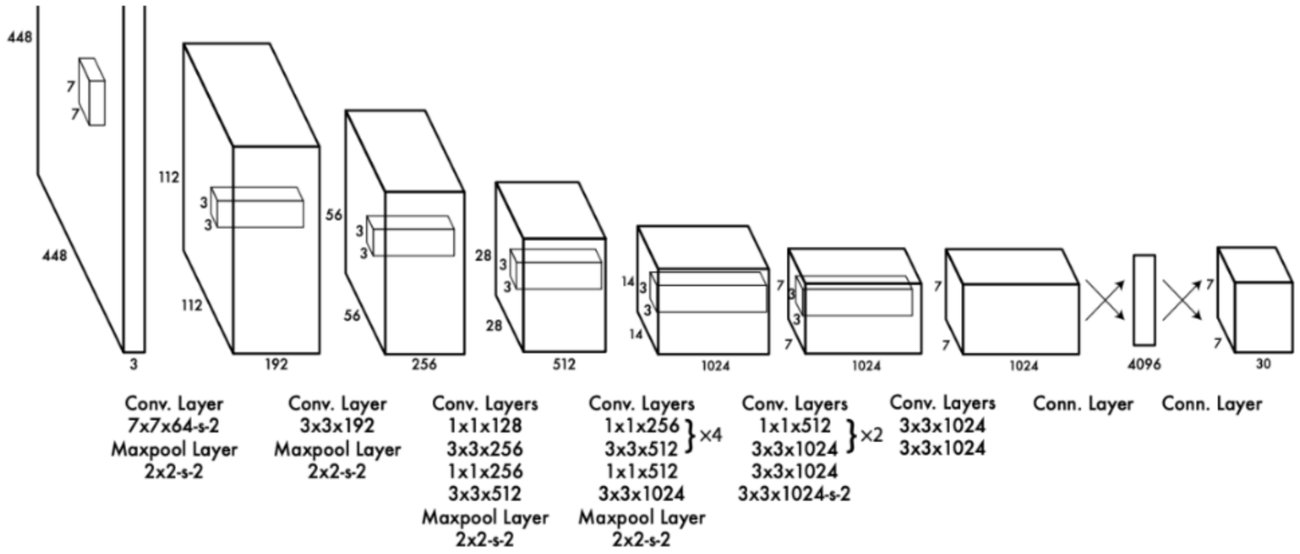

Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224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

YOLO 네트워크 구조는 이미지 분류를 위한 GoogLeNet과 유사

네트워크에는 24개의 컨볼루션 레이어이 있고 그 뒤에 2개의 fully connected 레이어 존재

1 * 1의 reduction 레이어와 3 * 3 컨볼루션 레이어를 사용

네트워크의 최종 예측의 출력은 7*7*30 tensor

좀 더 빠른 객체 인식 속도를 위해 YOLO보다 더 적은 컨볼루션 계층(24개 대신 9개)과 필터를 사용 -> Fast YOLO

04

Training

Training

Dataset : 파스칼 VOC 2007, 2012

Epochs : 135

Batch size : 64

Momentum : 0.9

Decay : 0.0005

Learning rate : 0.001(1~3 epoch) -> 0.01(4~74 epoch) -> 0.001(74~104 epoch) -> 0.0001(105~135 epoch)

Drop out : 0.5

Data augmentation : 원본 이미지의 20%까지 random scaling, random translation

마지막 계층 > 선형 활성화 함수(linear activation function)를 적용

나머지 모든 계층 > leaky ReLU를 적용

05

Loss Function

Loss Function

YOLO에서는 SSE(Sum-Squared Error, 오차제곱합) 손실함수 사용

$$
\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \\
&+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
&+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2
\end{aligned}
$$

문제 1. Localization loss와 Classification loss의 가중치를 동일하게 취급

문제2. 이미지 내 대부분의 그리드 셀에는 객체가 없어 불균형 초래

문제3. 큰 bounding box와 작은 boudning box에 대해 모두 동일한 가중치로 loss를 계산

Loss Function

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \quad \text{1}$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right] \quad \text{2}$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \quad \text{3}$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \quad \text{4}$$

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2 \quad \text{5}$$

06

Inference

Inference

추론 단계에서도 테스트 이미지로부터 객체를 검출하는 데에는 하나의 신경망 계산

↓

But, YOLO의 그리드 디자인(grid design)은 한 가지 단점 존재

↓

객체의 크기가 크거나 객체가 그리드 셀 경계에 인접해 있는 경우, 그 객체에 대한 bounding box가 여러 개 생기는 다중 검출(multiple detections) 문제 발생

↓

비 최대 억제(non-maximal suppression)라는 방법을 통해 개선(mAP를 2~3%가량 향상)

07

Experiments

Experiments

(1)

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.

(2)

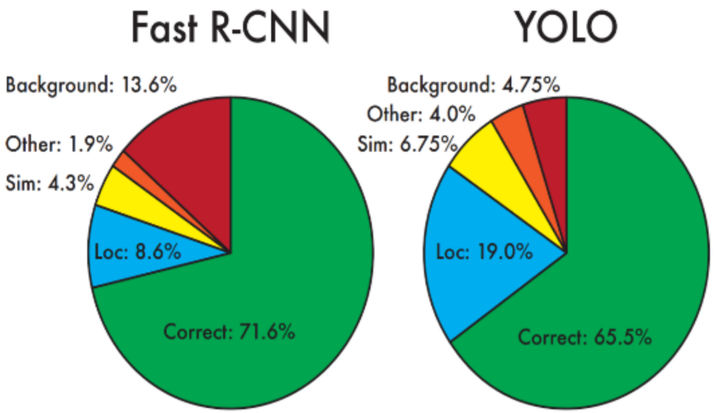

Figure 4: Error Analysis: Fast R-CNN vs. YOLO These charts show the percentage of localization and background errors in the top N detections for various categories (N = # objects in that category).

(3)

	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.

08

Conclusion

Conclusion

1) YOLO는 단순하면서도 빠르고 정확한 object detection 모델

2) training에서 보지 못한 새로운 이미지에 대해서도 object detection을 잘함

3) 여러 구역으로 나뉜 이미지를 보는 게 아니라 전체 이미지를 봄으로써
새로운 도메인에 대한 일반화 능력 우수

4) 새로운 이미지에 대해서도 성능이 좋기 때문에 Real-Time computer vision
application에서도 활용할 만한 가치가 있음

Reference

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

https://www.youtube.com/watch?v=8DjIJc7xH5U (십분딥러닝_14_YOLO(You Only Look Once)

https://www.youtube.com/watch?v=078V3kwBRBk ([Paper Review] You Only Look Once : Unified, Real-Time Object Detection)

https://bkshin.tistory.com/entry/%EB%85%BC%EB%AC%B8-%EB%A6%AC%EB%B7%B0-YOLOYou-Only-Look-Once (논문리뷰)

감사합니다 :)

2023.01.04