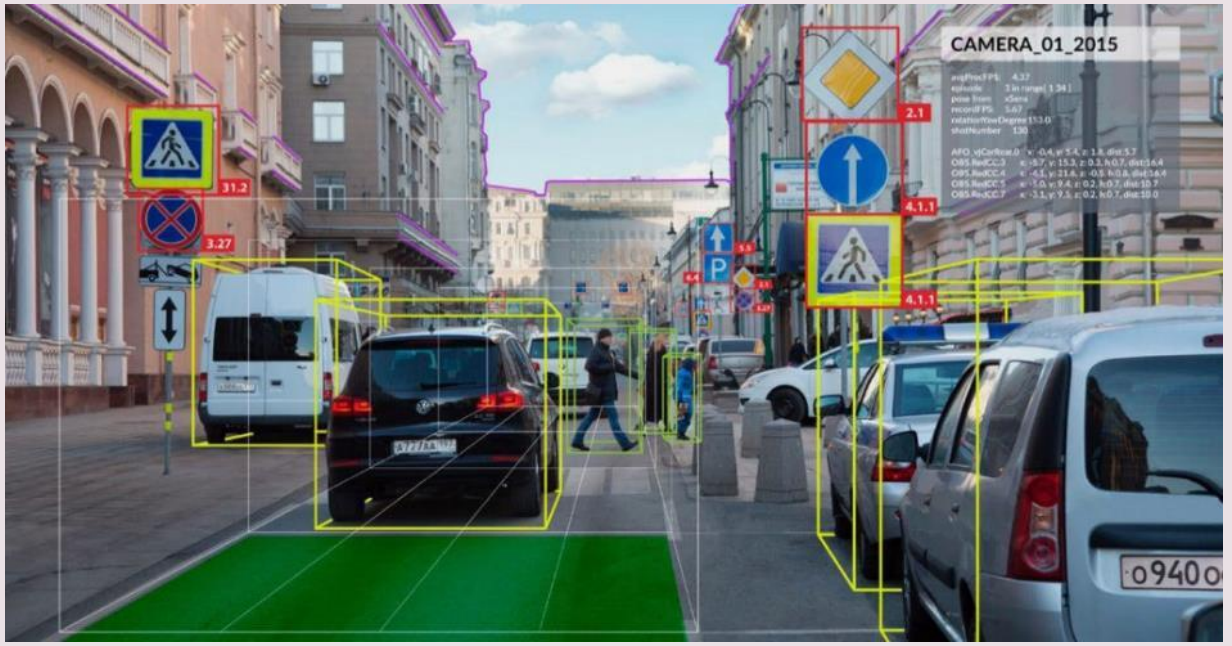
You Only Look Once: Unified, Real-Time Object Detection

YOLO 논문 리뷰

1. Introduction

인간의 시각 시스템은 이미지를 한 번 보고도 많은 정보를 파악할 수 있다.



빠르고 정확한 인간의 시각 시스템을 object detection에 적용한다면?

복잡한 구조나 센서 없이도 이미지를 인식하게 만들 수 있다.

특히 자동차 운전이나 보조 기기처럼 real-time 시각 정보를 사용하는 task에 유용하게 적용할 수 있다.

1. Introduction

이전 detection 모델의 특징

bounding box 찾기

object classification

- 최근 detection 모델은 classification 모델의 변용이다. 그래서 object classification, bounding box proposal 등의 과정이 따로 분리되어 있다.
- 예를 들어 R-CNN의 경우, bounding box proposal, classification, post-processing이 각각 다른 시스템으로 이루어져 있다.

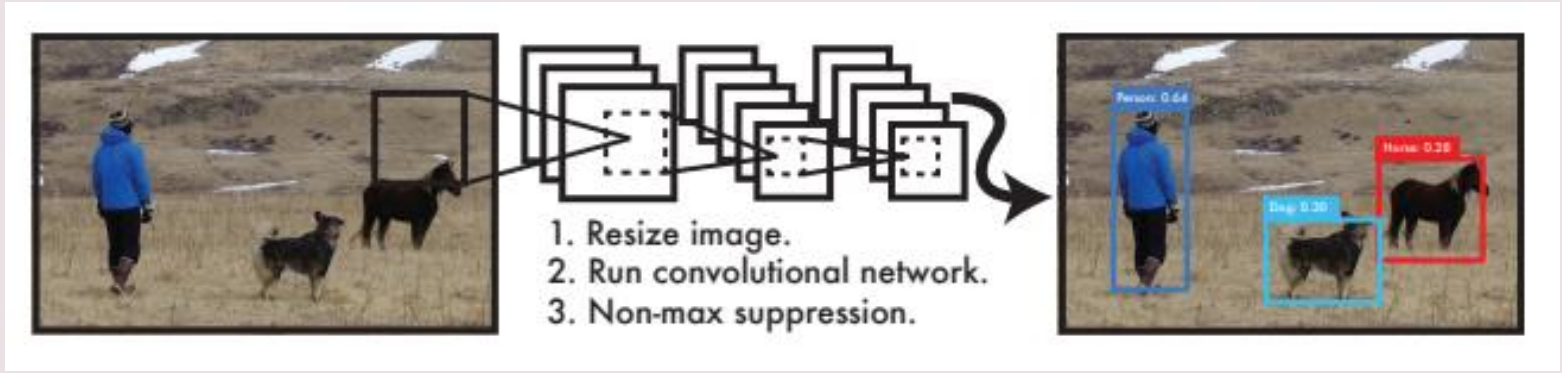
이러한 구조는 각 시스템이 따로 학습되어야 하기 때문에 속도가 느리고, 최적화하기에 어렵다.

1. Introduction

YOLO의 등장

bounding box 찾기

object classification



YOLO는 bounding box를 찾고, classification까지
한 개의 신경망으로 통합한 모델이다.
이를 Unified model이라고 한다.

1. Introduction

이전 모델과 비교했을 때 YOLO의 장점

1. 속도가 빠르다.

   구조가 간단해졌기 때문에 속도가 빨라진다. 그래서 real-time 이미지로 이루어진 스트리밍 영상도 빠르게 처리할 수 있다. 게다가 YOLO는 real-time 데이터를 처리하는 모델 중에서 높은 정확성을 보였다.

2. 이미지 전체를 활용한다.

   sliding window, RPN 등의 기법과 달리 YOLO는 수행 내내 이미지 전체를 활용한다. 그래서 contextual 정보를 보다 정확하게 얻을 수 있다. 이는 특히 background patch에 대한 에러를 감소시켰다.

3. 새로운 양식에 잘 적용된다.

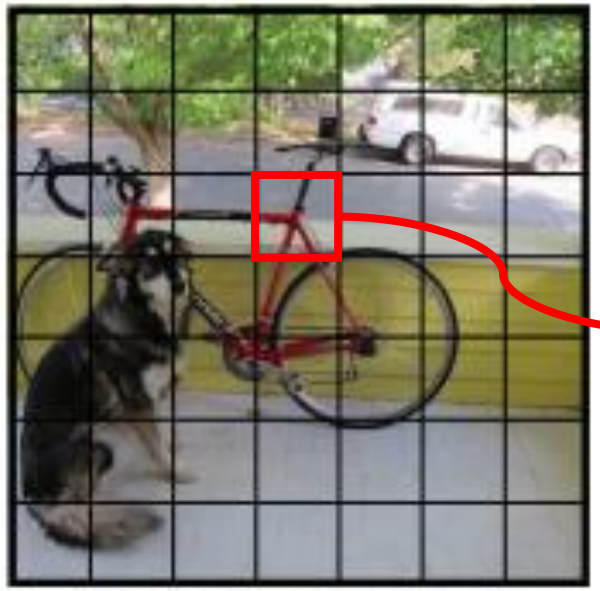
   YOLO는 object의 더 보편적인 정보를 학습하기 때문에 새로운 양식의 이미지에도 잘 적용된다. 예를 들어, 실제 사진으로 학습한 YOLO 모델은 미술 작품에도 잘 적용된다.

2. Unified Detection

- Unified detection은 object detection 모델의 분리된 요소를 하나의 신경망으로 통합한 모델이다.
- 이미지의 전체를 사용하여 detection을 수행한다. 그래서 모든 class에 대한 bounding box를 동시에 찾을 수 있다.

  -> reasons globally
- end-to-end training이 가능하다.
- real-time 데이터 처리의 속도와 정확성을 높였다.

2. Unified Detection



S × S grid on input

- 이미지를 S*S grid로 나눈다.
- 특정 object의 중심을 포함한 grid cell은 그 object의 detection을 담당한다.
- 각 grid cell은 B개의 bounding box와 박스에 대한 confidence score를 예측한다.
- confidence score: 이 박스가 object를 포함하고 있다고 확신할 수 있는 정도와 박스가 object를 정확히 예측할 수 있는 정도에 관련된 점수

$$confidence\ score = \Pr(Object) * IOU_{pred}^{truth}$$

2. Unified Detection

$$confidence\ score = \Pr(Object) * IOU_{pred}^{truth}$$

grid cell 내 object의 존재 여부

predicted box와 ground truth box의 IOU

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

IOU는 겹치는 정도에 대한 점수를 말한다.

2. Unified Detection

bounding box는 5개의 예측값$(x, y, w, h, confidence)$으로 이루어져 있다.

x, y: 박스의 중심좌표

w, h: 박스의 가로와 세로 길이

각 grid cell은 C개의 class에 대한 조건부 확률 $\Pr(Class_i|Object)$을 예측한다.

bounding box의 개수와 상관 없이 한 세트의 class 확률만 예측한다.

$$\Pr(Class_i|Object) * \Pr(Object) * IOU_{pred}^{truth} = \Pr(Class_i) * IOU_{pred}^{truth}$$

class-specific confidence score

2. Unified Detection

S x S grid on input

1. 이미지를 S*S grid로 나눈다.

Bounding boxes + confidence

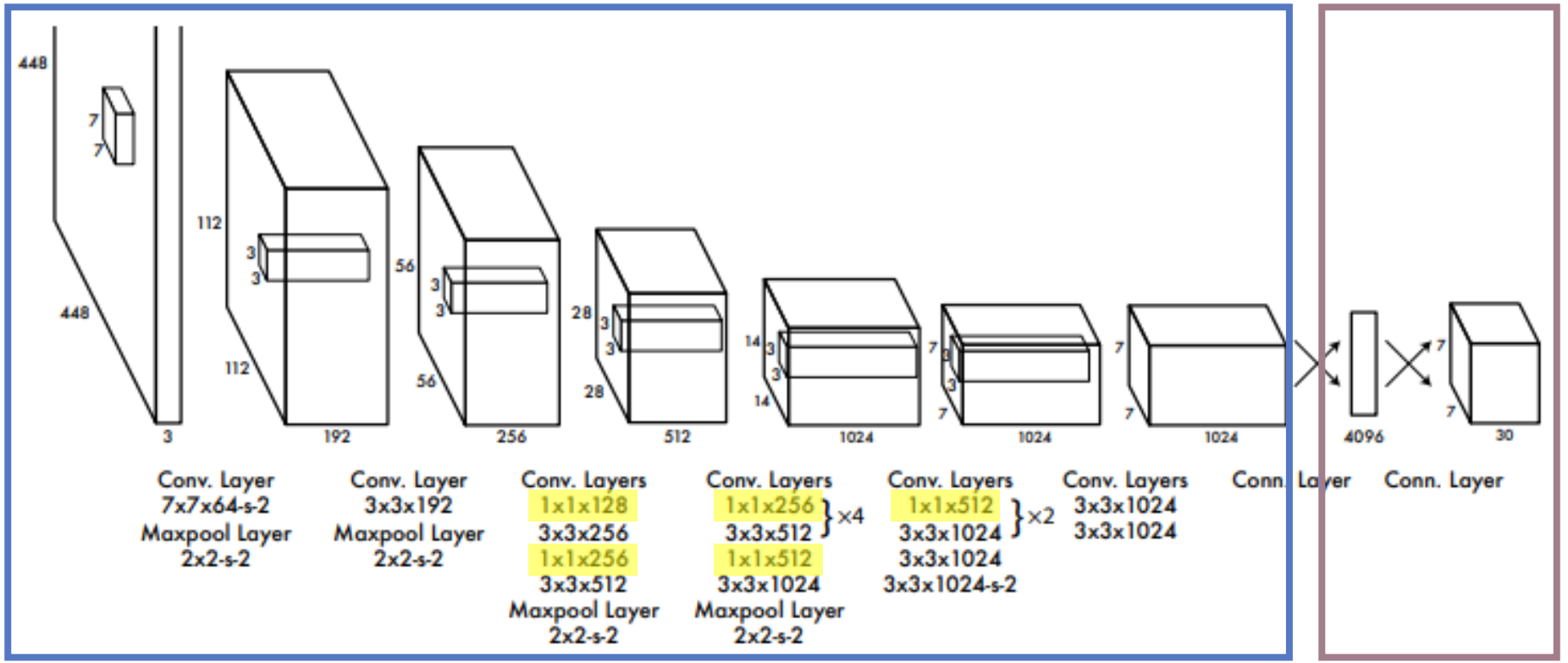
2. grid cell 마다 B개의 bounding box를 예측한다.

Class probability map

3. grid cell 마다 C개의 class에 대한 확률을 예측한다.

Final detections

2-1. Network Design



convolution layers

- 이미지로부터 feature를 추출
- 총 24개(fast YOLO는 9개)
- 3*3 convolutional layer 앞에 1*1 reduction layer 사용

fully-connected layers

- output의 확률과 좌표를 예측
- 총 2개

2-2. Training

1. pretrain

20개의 convolutional layer를 ImageNet(1000 class) 데이터셋으로 학습시킨다.

2. detection 모델로 변환

pretrain 모델에 4개의 convolutional layer와 2개의 fc layer를 추가하여 성능을 높였다. 추가된 레이어의 가중치는 랜덤하게 초기화.

그리고 고화질의 시각 정보를 위해 이미지 해상도를 224*224에서 448*448로 증가시켰다.

3. 정규화

마지막 레이어는 class 확률과 bounding box의 좌표를 예측한다. 이때 bounding box의 w, h와 x, y는 원본 이미지에 따라 정규화되어 0과 1사이의 값을 갖는다.

4. 활성화 함수

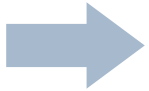
모든 레이어에 대해 활성화 함수로 leaky ReLU를 사용

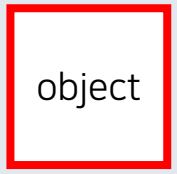
$$\phi(x) = \begin{cases} x, & if\ x > 0 \\ 0.1x, & otherwise \end{cases}$$

2-2. Training

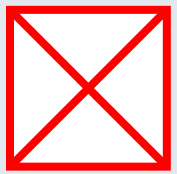
5. loss function

- sum-squared error를 사용하면 최적화하기에는 쉬우나 localization error와 classification error의 가중치를 동일하게 취급하기 때문에 올바르게 최적화되지 않는다.
- 또한 대부분의 grid cell은 object를 포함하지 않아서 대부분의 cell의 confidence score가 0이 되도록 학습하는 경향이 생길 수 있다.



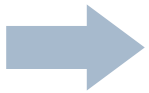


$\lambda_{coord} = 5$를 사용하여 loss 증가



$\lambda_{noobj} = 0.5$를 사용하여 loss 감소

- 또한 크기가 큰 박스와 작은 박스의 error를 동일하게 취급하는데 큰 박스 내의 작은 편차는 작은 박스 내의 편차보다 덜 중요하게 다뤄져야 한다.



bounding box를 예측할 때, w와 h에 제곱근을 취했다.

원래 값이 아니라 제곱근을 예측하면 w와 h가 커질수록 예측값의 증가율은 감소하기 때문이다.

2-2. Training

YOLO 학습 과정에서 한 개의 object는 한 개의 bounding box 예측만 필요로 한다. 그러므로 ground truth와 가장 높은 IOU를 가진 1개의 박스만 object에 predictor로 할당된다. 이 predictor는 특정 크기, 각도, class를 잘 예측하게 된다.

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

object가 존재하는 cell i의 j번째 bounding box에서 x, y의 loss

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

object가 존재하는 cell i의 j번째 bounding box에서 w, h의 loss

ground truth에 대한 predictor에서는 bounding box error를 규제

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

object가 존재하는 cell i의 j번째 bounding box에서 confidence score의 loss

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

object가 존재하지 않는 cell i의 j번째 bounding box에서 confidence score의 loss

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

object가 존재하는 cell i에서 classification의 loss

object가 있는 grid cell에서는 classification error를 규제

2-2. Training

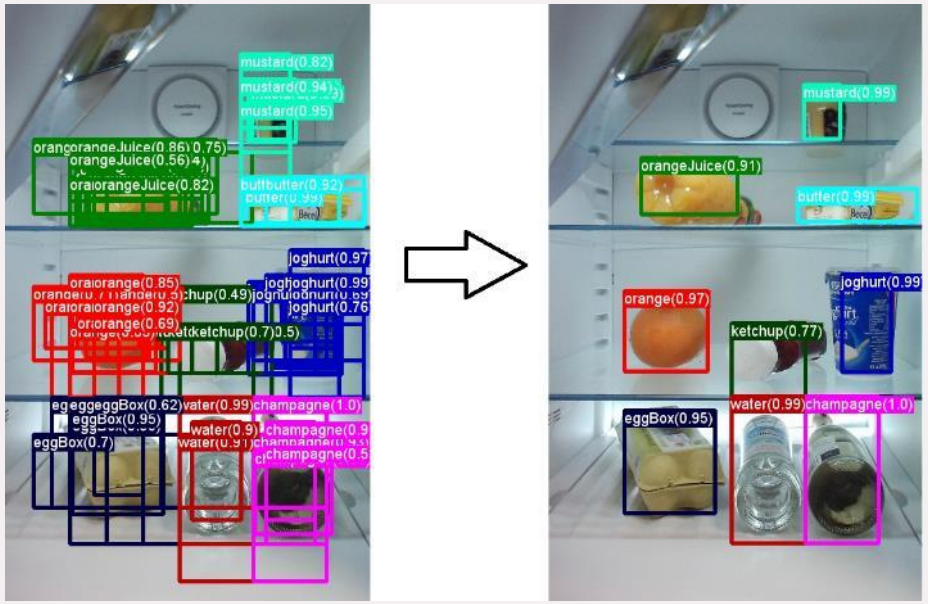
6. parameters

- epoch: 135
- datasets: PASCAL VOC 2007(training용), 2012(testing, training용)
- batch size: 64
- momentum: 0.9
- weight decay: 0.0005
- learning rate schedule: 처음부터 높은 learning rate를 사용하면 gradient가 불안정하기 때문에 첫번째 epoch 에서는 예외적으로 10^{-3}에서 10^{-2} 로 천천히 증가시킨다. 그리고 처음 75epoch 동안은 10^{-2} , 그 후 30epoch 동안 10^{-3} , 최종적으로 30epoch 동안 10^{-4} 로 진행한다.
- overfitting: 과적합을 피하기 위해서 0.5 rate의 dropout 레이어를 첫번째 fc layer 이후에 추가하였다. 그리고 원본 이미지 크기의 20%까지 random scailing과 translation을 수행하였다. 그리고 이미지의 HSV color space의 1.5까지는 랜덤하게 exposure과 saturation를 적용하였다.

2-3. Inference

- 추론 과정에서도 오직 1개의 신경망을 사용한다. 1개의 신경망에 대해서만 evaluation을 진행하면 되므로 속도가 빠르다.
- PASCAL VOC 신경망에서는 한 이미지 당 98개의 bounding box가 예측되었으며, 각 박스마다 class 확률을 예측한다.
- YOLO는 grid 구조 상에서 진행되기 때문에 multiple detection 문제가 발생할 수 있다.

Non-maximal suppression



여러 개의 bounding box 중 가장 정확한 박스를 찾고 나머지는 제거하는 기법

2-4. Limitation of YOLO

- YOLO는 각 grid cell에서 2개의 bounding box와 1개의 class만을 예측할 수 있기 때문에 서로 가까운 object는 잘 예측하지 못한다.
- 데이터로부터 bounding box 예측을 학습하기 때문에 새로운 설정의 이미지에는 잘 적용되지 않는다.
- downsampling layer가 여러 개 있기 때문에 선명하지 못한 feature로 학습해야 한다.
- loss function은 큰 bounding box와 작은 bounding box에서 일어난 error를 동일하게 취급한다.