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

https://arxiv.org/abs/1506.02640

1. introduction

복잡한 파이프라인으로 구성된 기존모델과 달리 이미지에서 bounding box 좌표와 클래스 확률을 즉시 예측함으로써 object detection을 단일 회귀 문제로 재구성. 이미지를 한번만 보고 객체의 위치와 종류를 파악 가능. yolo is extremely fast. yoly reasons globally about the image when making predictions. yolo learns generalizable representations od objects.

2. unified detection

- YOLO는 구조가 하나로 통합되어 Single Neural Network 구조 형성
- 단 하나의 네트워크로 전체 이미지의 feature를 사용해 모든 클래스와 그에 대한 bounding box를 동시에 예측
- 즉, 전체 이미지와 이미지 내 모든 object에 대해 추론

Detection 과정

1.input 이미지를 S × S grid로 분할 (논문에서는 7x7)

2.grid cell당 Bounding box와 Class probability 예측

-각 그리드 셀은 bounding box B와 해당 box의 confidence score(신뢰도 점수)를 예측

-confidence score

해당 box가 객체를 얼마나 포함하는지, 그 정확도는 얼마인지 등 모델이 얼마나 신뢰 가능한지를 나타내는 점수
confidence score = Pr(Object) * IOU = 객체가 존재할 확률과 IOU 값의 곱
box 내 객체가 존재하지 않으면 객체 존재 확률 Pr(Object)가 0이므로 confidence score = 0

각 bounding box는 x, y, w, h, confidence score로 구성 (x,y : box 중심좌표 / w,h : box 너비, 높이)

각 그리드 셀은 조건부 클래스 확률 Pr(Class i |Object)를 예측

그리드 셀이 객체를 포함할 때, 해당 객체가 i번째 클래스일 확률
-각 그리드 셀당 확률값이 가장 큰 하나의 클래스만 예측

3.NMS를 통해 최종 예측

3. network design

-전체 구조 : convolutional layers 24 (feature 추출) + fully connected layer 2 (확률과 좌표 예측)
-1000 class의 ImageNet 데이터셋으로 20개의 conv layer를 pre-train
-1 × 1 컨볼루션 레이어를 번갈아 사용해 선행 레이어의 feature space를 축소
-활성함수 : leaky ReLU (마지막 layer만 linear 함수)
-최종 output : 7 × 7 × 30 tensor (S=7, B=2, C=20)

4. Training

pretrained:이 네트워크는 20개의 Conv layer로 구성되어 있고, 88%의 정확도.

원래 ImageNet dataset의 이미지는 (224x224)인데 좋좋 detection에서는 더 세분화 된 시각 정보 (fine-grained visual information)가 필요하므로 input resolution을 2배인 (448x448)로.

<loss function>

- Object가 존재하는 grid cell i의 predictor bounding box j에 대해, x와 y의 loss를 계산

- ground thuth box와 bounding box의 x, y좌표의 위치차를 구해서 loss계산

- Object가 존재하는 grid cell i의 predictor bounding box j에 대해, w와 h의 loss를 계산

- 큰 box에 대해서는 small deviation을 반영하기 위해 제곱근을 취한 후, sum-squared error (같은 error라도 larger box의 경우 상대적으로 IOU에 영향 적게 줌), 큰 에러가 아닌데 그 값들의 크기가 크기에 큰 에어로 판별할 오류 방지

-Object가 존재하는 grid cell i의 bounding box j에 대해, confidence score의 loss를 계산(Ci=1)

- 디폴트 값은 1임(존재하는 것에 대한 loss를 구함) 따라서 1-confidence socre를 빼서 loss를 구함.

- Object가 존재하지 않는 grid cell i의 bounding box j에 대해, confidence score의 loss를 계산 (Ci=0)

- 객체가 존재하지 않으면 디폴트 0 하지만 confidence score는 클래스 확률값을 곱했기에 0보다 항상높음
- 그럼 이것이 0보다 얼마나 높은지에 대한 loss를 구하는것

-Object가 존재하는 grid cell i에 대해, conditional class probability의 loss 계산 (Correct class c: pi(C)=1, otherwise: pi(C)=0)

- 각각의 grid cell이 총 20개의 class probability를 가지고 있게 되는데 ground-thuth값이 있고 예측한 conditional probability가 20개가 있는데 이 두 값을 계산해주는 것이 마지막 lossfunction

4. limitaions od yolo

-각 그리드는 두개의 bbox를 예측하고 하나의 class를 가질 수 있기 때문에 공간적 제약이 생길 수 밖에 없다. 때문에 작은 물체에 대한 탐지가 어려움이 생긴다.

-bounding box의 형태가 학습 데이터를 통해서만 학습되기 때문에, 새로운 Input으로 이전에 학습되지 않은 형태의 Bounding box를 예측해야 할 경우, 제대로 된 예측을 할 수 없다.

-몇 단계의 Convolutional Layer를 거쳐 나온 Feature map을 대상으로 Bounding box를 예측하기 때문에 객체의 위치를 정확히 파악하는 Localization이 다소 부정확해지는 경우가 발생

5. conclusion

-YOLO 모델은 간단하게 구현이 가능하며, 전체 이미지를 바로 학습시킬 수 있다

-Classified-based 접근 방식과는 다르게 Detection 성능과 직접적인 관계를 맺고 있는 loss function을 통해 학습

-YOLO는 Real-Time Object Detection의 성능 지표를 올렸으며, 일반화가 잘 되기 때문에 새로운 도메인에도 쉽게 적용 가능한 모델이다.