YOLO

**Abstract**

Object detection (이하 O.D)는 사전에 decection을 먼저 수행한다. ( Box를 찾는걸 의미하는 듯 )

하지만 YOLO는 O.D를 공간적으로 분리된 Bound Box들과, class의 확률과 연관된 회귀문제로 새롭게 정의하는 접근법이다.

전체 이미지에서 Single Neural Network를 통해 단 한번의 평가를 통해 직접적으로 Bound Box를 설정하고 class의 확률을 계산한다.

파이프라인이 Single Neural Network 이므로, Detection 성능의 즉각적 최적화가 가능하다.

통합된 아키텍쳐의 속도가 극적으로 빠르며, 45FPS의 실시간 이미지 연산이 가능하다. 작은버전인 Fast YOLO는 다른 실시간 Decection 모델보다 2배 높은 정확도를 보여주면서, 155FPS의 성능을 보여준다.

다른 Detection 모델과 비교했을 때, Localization error가 높은 편, 하지만 배경으로 인한 error는 더 낮다. 그리고, object의 일반적 표현들을 학습하는 특성이있다.

**Introduction**

인간은 이미지를 잠깐 스쳐보더라도, object의 위치와, 상호작용을 즉각적으로 파악 할 수 있다.

이러한 특성은 사람에게 복잡한 Task를 가능하게 하는데, 이런 것들과 유사한 O.D 알고리즘은 엄청난 잠재력을 가지고 있다.

R-CNN과 같은 O.D 시스템은 복잡한 파이프라인을 가지고 있으며, 따라서 느리고 최적화 하기 어렵다.

YOLO는 위에서 설명한 바와 같이, 픽셀로부터 box좌표를 얻고, Classification 과정을 single regression problem으로 재정의했다. YOLO를 사용하면, 이미지를 한번만 보고 현재 image에 있는 object가 무엇이며 어디있는지 알 수 있다.

**Trade Off**

Classifier 능력이 다소 떨어짐, Object를 빠르게 식별할 수 있으나, 작은 것은 잘 검출하지 못함,

**Unified Detection**

1) YOLO는 이미지를 S\*S Grid로 나눈다.

+ Grid cell안에 Object의 중심이 포함되면, 그 cell은 해당 object를 Detect한다.

2) 각 Grid cell은 Bound Box와, 그 Box에 대한 Object 존재 확률과, 그 Object가 무엇인지 예측했을 때, 정답일 확률을 계산 -> 후자를 Confidence Score라고 표현

**Network Design**

1) CNN으로 구현했으며, GoogleNet 모델로부터 영감을 받음

2) 구조는 24개의 Convolutional layer와 2개의 FC layer ->

3) 첫 Convolutional layer는 image로부터 특징을 추출하고, FC Layer에선 확률과 좌표 예측