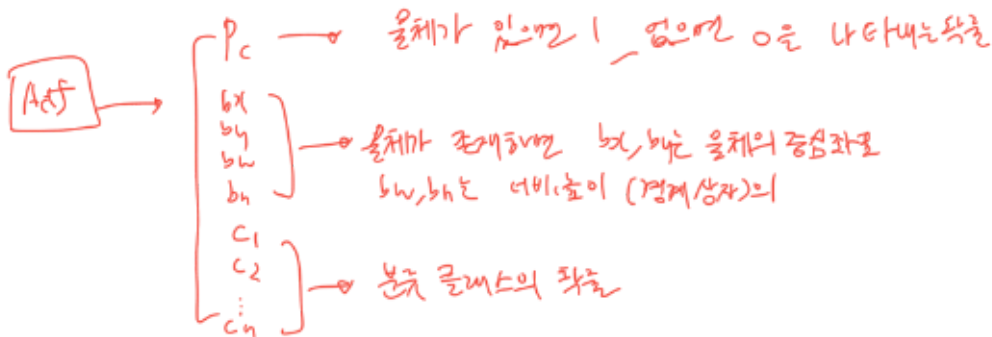


YOLO 알고리즘

각 이미지 격자셀에서 경계상자(bounding box)와 사물의 데루리(대수리)를 정확히 예측하는 방법을 파악.



YOLO (You only Look Once)

\therefore 이미지 전체를 1번만 본다.

기존 \rightarrow 경계상자를 인식해서 있는지 확인
 or
 sliding window 방법 창을 이동시키다 \rightarrow CONV로 반복연산으로 가능
 FC를 (CONV연산으로 진행)

Unified Detection

\rightarrow Image \rightarrow SXS grid 

\rightarrow 경계상자 (b_x, b_y, b_w, b_h)로 표현, confidence는 경계상자에 사물이 얼마나 들어있는지.

YOLO는 GoogLeNet V1을 따르는 모델

Inception 모듈 사용.

2단계 FC하지 않고 앞에 CONV를 추가함.

\downarrow
 7×7 의 작은 grid라서 bounding box의 b_x, b_y 를 포함하는 grid셀을 물체의 중심셀로 잡는다.

물체 \rightarrow 작은 image에서 같은 grid에 여러 물체 중심이 포함된다면 classification이 아니라 object part 이다.

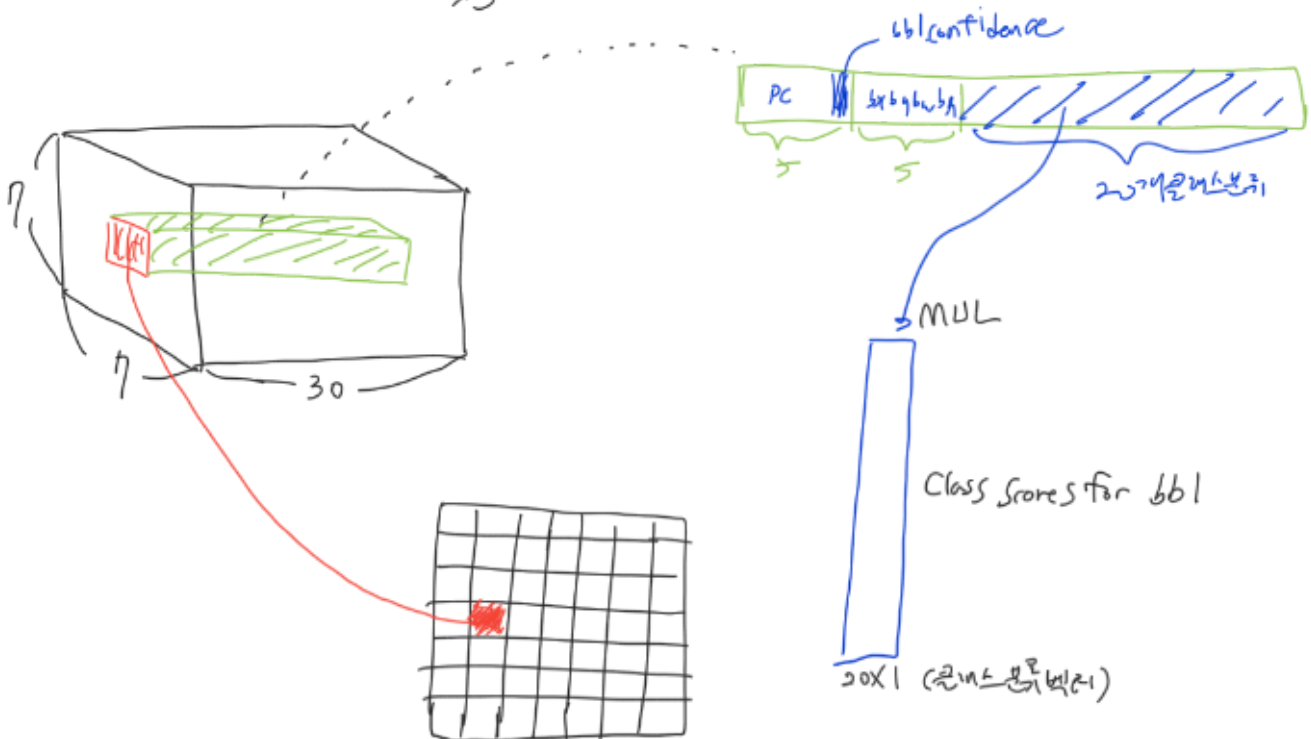
그런데 class에 따라 바뀔 수 있다.

하지만 특징맵에 있는 각 grid cell마다 색을 달리하여
이 영역 grid는 어떤 물체를 나타내는 일부를 것을 책임진다.

↓

YOLO의 loss function은 SSE (Sum-squared Error)를 사용

하나의 grid cell을 중심으로 하는 bounding-box는 하나만 갖게끔
손실함수를 구성



* Non-Maximum Suppression: NMS (가장 높은 것만 bounding box로 안짓는다)

bb1 bb2 bb3 ... bb8 ⇒ $1 \times 1 \times 2 \rightarrow 1$ (grid cell 당
2개 경계상자
가질을 가장

grid size에 의존해볼. current grid size 보다 작은 물체가 나타난다면

← 가장 높은 것의 grid cell을 찾지 못하기 때문
인식 안됨

⇒ bounding-box가 0(마지 바깥으로 나가게 된다면?)



마지막 수정: 오후 4:43