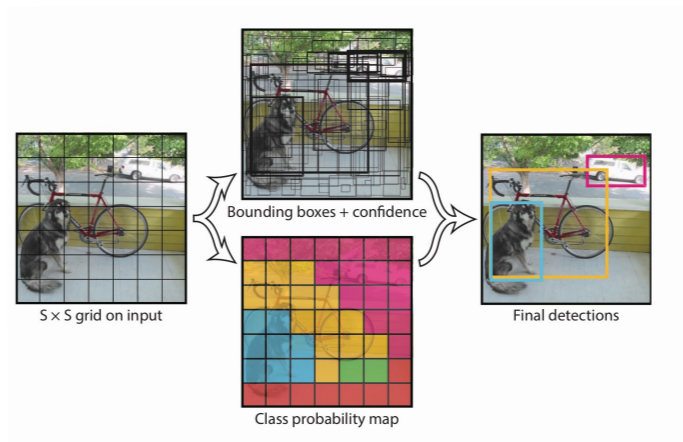


Yolo

- ⇒ 2-stage detector 쓰지말고 1-stage로 Region proposal + Class 수렴
 대신 speed ↑↑, but accuracy ↓ over.




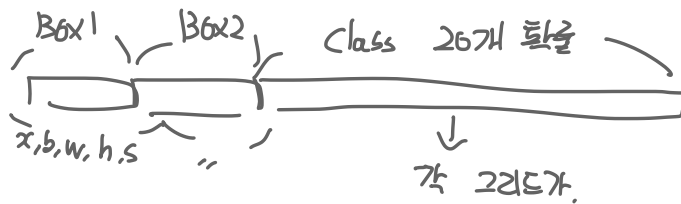
⇒ 1. 5×5 grid 로 나뉜다

2. 각 grid에 bounding boxes의 중심 좌표가 해당 셀 내에 있다면
 해당 셀이 bb를 담당한다 (response)
 (논문에서는 7×7 보다 많음 bb 2개로 box 98개 존재)
 → box 중심, x, b, w + confidence 학습
 \hookrightarrow box 안 객체 존재 = IOU
 " 안 존재 = 0
- 각 grid 별로 어떤 물체를 담당하고 있는지 보는데
 20개 class (Yolo에서 20) 중 grid는 어떤 class 인지 가장 높은 확률
3. 2에서 얻은 2개 결과를 $\frac{\text{max}}{\text{자물개 갯}}$ 한후 결과 도출 * $\text{Pr}(\text{class} | \text{object})$

Encoder as $(S \times S + 5 \times B + C)$
 \downarrow
 $x, b, w, \text{anh}, \text{ReLU}$


* Google Net 기법을 사용 (inception model $\rightarrow 1 \times 1$)

 $\rightarrow \rightarrow \rightarrow$ FC $\rightarrow 7 \times 7 \times 30$ 으로 만든다.
1470

- $7 \times 7 \times 30$ 에서 30은
 feature를 나누어줘 있다

 이러한 30개가 7×7 만큼 있다는 뜻

- Loss function 1개로 최적화- 학습
 (근데 1개라서 하이퍼 파라미터 튜닝 했다)

* **질문** FC 1470 $\rightarrow 7 \times 7 \times 30$ 으로 만들때 위치정보 없는데
 이게 어떻게 되는건가요?

FCL 이기 때문에 결국  이기 때문에
 위치 정보 고려 된다. 하지만 낭비되는 weight 가 있다. \rightarrow 앵커박스 사용
 \rightarrow 그래서 YOLO 2에서는 fully convolutional network로 ID 한다!!

- 단점 \Rightarrow bb로 학습해 각본것은 따로 알아둬

일반적이지 않은 사례 사람도 감지 힘들

\rightarrow 처리량도 작음, input size 많음

\Rightarrow YOLO 2 \Rightarrow FC \rightarrow FC, 앵커박스 도입

