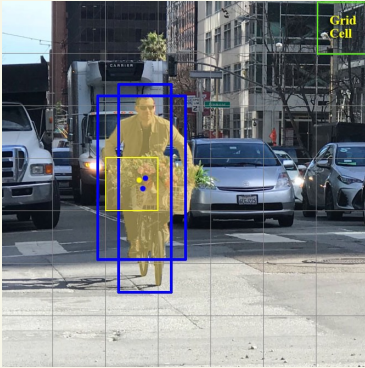


Object Detection 모델

Yolo, Faster-RCNN, MobileNet, SSD 등

Yolo, YoloV2, YoloV3(Yolo 9000)까지 개발

YoloV3 - YoloV2에 비해 노출 (성능 ↑)



5x5 Grid cells로 나눔

각 cell마다 하나의 객체 예측.

미리 설정된 객체의 boundary boxes를 통해 객체의 위치와 크기 파악

- 각 cell마다 하나의 객체만 예측 가능하기 때문에 여러 객체가 겹쳐있으면 몇몇 객체는 탐지 불가능

각 cell은 다음 2개 값에 예측 정보

→ box의 위치를 파악하고 있을 가능성(objectness)과 boundary box의 위치를 결정할 수 있는 방법

- B개의 boundary boxes를 예측하고 각 box는 하나의 box confidence score를 가지고 있다.
- 예측된 box 수에 관계없이 단 하나의 객체만 탐지한다.
- C개의 conditional class probabilities를 예측한다.

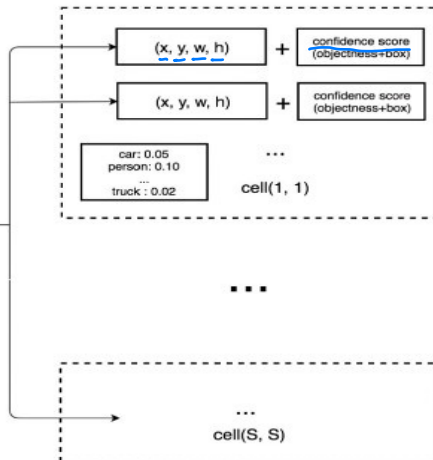
→ 탐지된 객체가 어느 특정 클래스에 속하는지에 대한 정보

(x,y) 객체의 위치
(w,h) 객체의 크기

개별화 5개



YOLO



$$\begin{aligned} \text{box confidence score} &\equiv P_r(\text{object}) \cdot \text{IoU} \\ \text{conditional class probability} &\equiv P_r(\text{class}_i | \text{object}) \\ \text{class confidence score} &\equiv P_r(\text{class}_i) \cdot \text{IoU} \\ &= \text{box confidence score} \times \text{conditional class probability} \end{aligned}$$

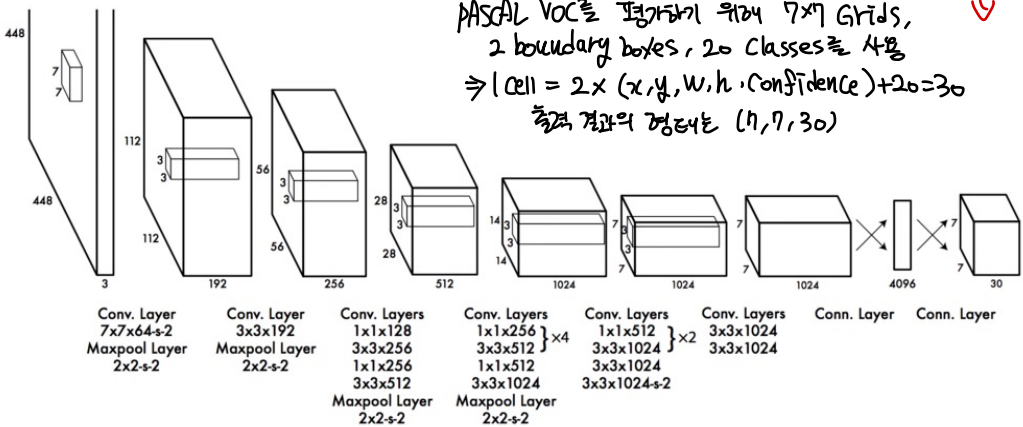
where

$P_r(\text{object})$ is the probability the box contains an object.
 IoU is the IoU (intersection over union) between the predicted box and the ground truth.
 $P_r(\text{class}_i | \text{object})$ is the probability the object belongs to class_i , given an object is presence.
 $P_r(\text{class}_i)$ is the probability the object belongs to class_i .

각 Score는 수학적으로 위와 같이 정의된다.

Class confidence score는 분류와
 지역화(localization) 둘 다에 대한
 Confidence를 측정

network design GoogleNet 구조 변경



YOLO \Rightarrow (7, 7, 30) 텐서를 예측하는 CNN 네트워크를 구축

CNN 모델 = spatial dimension을 1024개의 channel을 가진 7x7 dimension으로 줄임

YOLO = 7x7x2 boundary box 예측을 만들기 위해 2개의 fully-connected layers를 통해 linear regression을 수행

\therefore YOLO는 24개의 Convolutional layers와 2개의 fully-connected layers로 구성

몇몇 convolution layers는 1x1 커널을 사용하여 feature maps의 크기를 줄이고 최종적으로 (7, 7, 30) 형태의 텐서를 생성

Loss function

Yolo는 각 grid cell마다 다수의 bounding boxes를 예측하지만 true positive에 대한 loss를 계산하기 위해 양자된

실제 객체와 예측 값에 대한 차이

객체를 가장 잘 포함하는 box 하나를 선택,

=> ground truth, IOU를 계산해 가장 높은 IOU를 가진 하나 선택

두 가지

• Classification loss

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

where

$\mathbb{1}_i^{obj} = 1$ if an object appears in cell i , otherwise 0.
객체가 cell i 에 나타나면 1, 그렇지 않으면 0

$\hat{p}_i(c)$ denotes the conditional class probability for class c in cell i .
cell i 의 class c 에 대한 class 확률

객체가 탐지되었다면, 각 cell의 classification loss는 각 클래스의 class conditional probabilities의 squared error다. 위에 수식에서 볼 수 있듯이 객체가 탐지되지 않았다면 0이 된다.

• localization loss (errors between the predicted boundary box and the ground truth)
boundary box와 ground truth 값 사이의 errors

$$\lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{i,j}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{i,j}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$$

where

$\mathbb{1}_{i,j}^{obj} = 1$ if the j th boundary box in cell i is responsible for detecting the object, otherwise 0.
어떤 객체가 탐지되었는지, 어떤 경계 상자인지
 λ_{coord} increase the weight for the loss in the boundary box coordinates.
boundary box 좌표의 손실에 대한 가중치를 증가

객체가 탐지되지 않은 경우에 대해서는 loss 값을 계산하지 않는다. 수식을 살펴보면, 위치는 sum squared error를 그대로 적용하지만 크기에 대해서는 각 높이와 너비에 대해 루트 값을 씌워 계산했다. 이렇게 하는 이유는 절대 수치로 계산을 하게 되면 큰 box의 오차가 작은 box의 오차보다 훨씬 큰 가중치를 받게 된다. 예를 들어, 큰 box에서 4 픽셀 에러는 너비가 2 픽셀인 작은 box의 경우와 동일하게 된다. 따라서 YOLO는 bounding box 높이와 너비의 제곱근을 예측하게 된다. 추가적으로, 더 높은 정확도를 위해 λ_{coord} (default: 5)를 loss에 곱해 가중치를 더 준다

· Confidence loss (the objectness of the box)