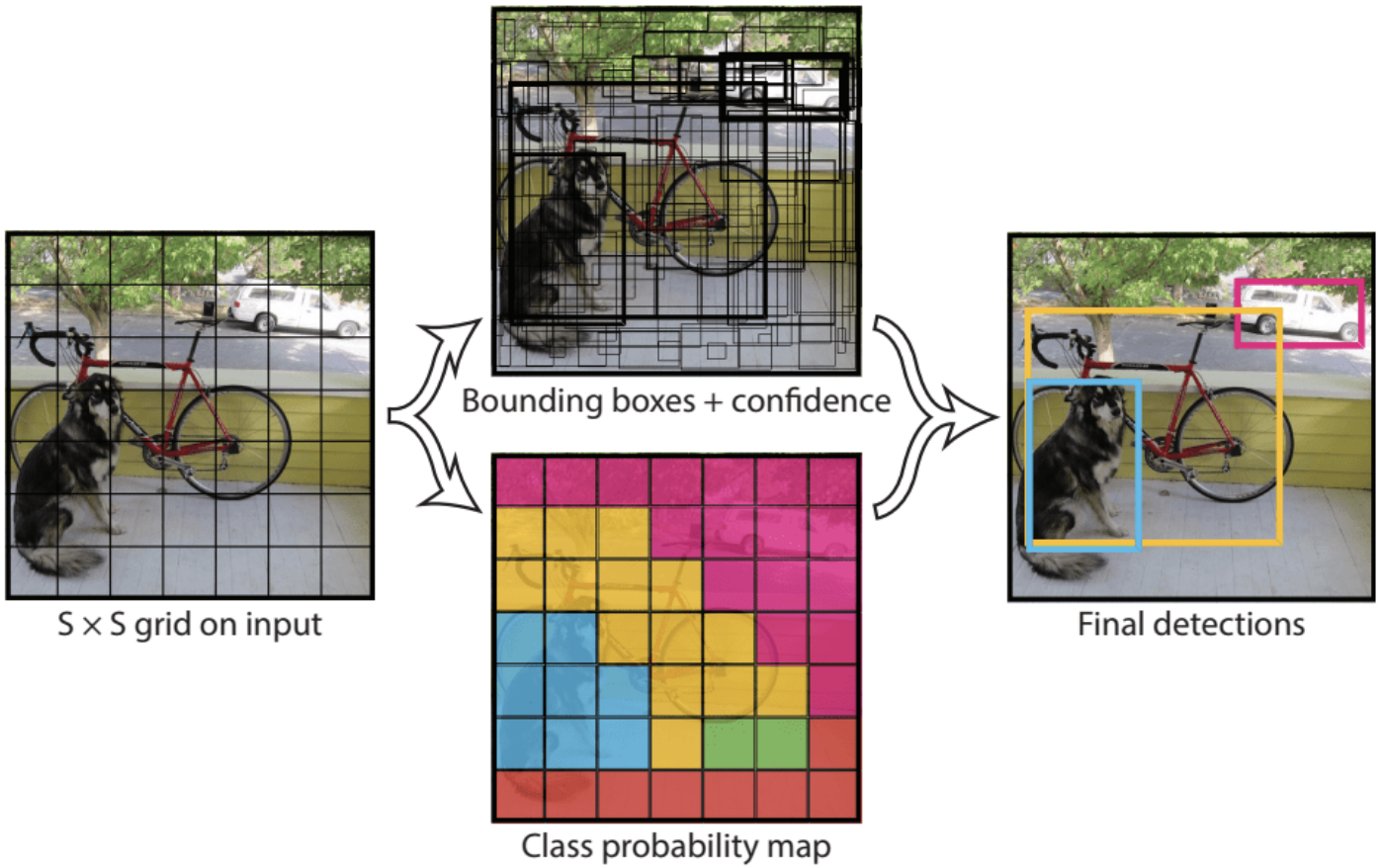
MAC5921 – Deep Learning

Aula 09 – 14/09/2023

Nina S. T. Hirata

YOLO – You Only Look Once



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

https://arxiv.org/abs/1506.02640

Entrada: Imagem

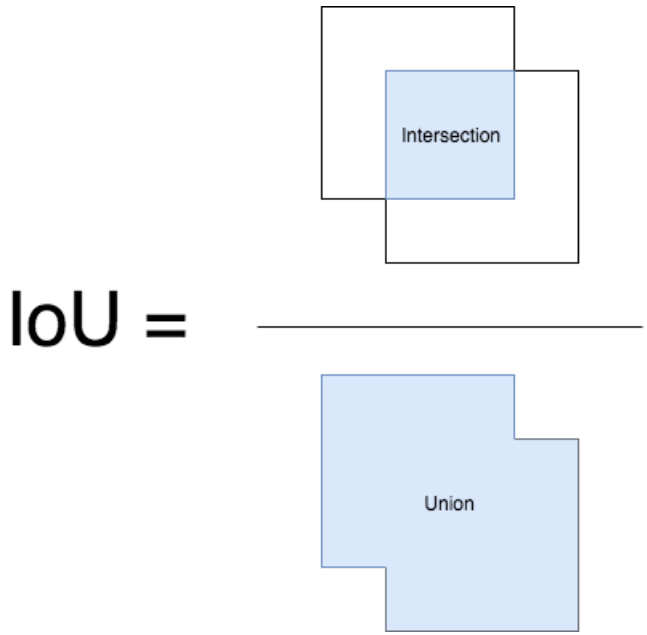
Target (ground-truth): uma lista de objetos contidos na imagem;

Cada objeto é um par formado por um BB (bounding box) e um rótulo de classe

BB: (x, y, w, h), em que (x, y) é o centro do BB

K rótulos de classe possíveis

IoU – Intersection over union



Qualidade da localização é medida em termos de IoU

$\mathrm{IoU} = 0 \longrightarrow$ overlap nulo

$\mathrm{IoU} = 1 \longrightarrow$ overlap perfeito

Treinamento

- Imagem é particionada em $S \times S$ células

- Para cada target object, a célula i que contém o centro do BB fica responsável pela detecção do objeto ($C_i = 1$ se célula i contém objeto, 0 c.c.)

- Para cada célula i, são calculadas K class prediction scores $P(\text{class}_k|\text{object})$

- Para cada célula são preditas B BBs

- Para cada anchor box j na célula i, são preditas as coordenadas (x, y, w, h) e um score C_{ij} (objectness)

- Assim, para cada célula i são preditos

  K class prediction scores $P(\text{class}_k|\text{object})$

  5 valores para cada BB

- Total de valores preditos para uma imagem

$$S \times S \times (5 * B + K)$$

YOLO – função de perda

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right]$$

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_{ij} \right)^2 + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_{ij} \right)^2$$

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

(os índices i e j deveriam começar em 1 ..., ou não?)

λ_{coord} e λ_{noobj}: evitar que células sem objeto dominem a função de perda

$\mathbb{1}_{ij}^{\text{obj}}$: indica que o BB j tem maior objectness score $\hat{C}_{ij}$ na célula i, e portanto é responsável pela detecção do objeto em i

$\mathbb{1}_{i}^{\text{obj}}$: indica se o objeto está na célula i

$p_i(k)$ seria $P(class_k|obj)$ (?)

Predição

Podemos pensar que $\hat{C}_{ij} \approx P(object) * IoU(BB_{gt}, BB_j)$, com idealmente $P(object) = 1$ se o objeto está na célula i e $P(object) = 0$ caso contrário

Em relação à célula i, calcula-se para cada BB j o valor $P(class_k|object) * \hat{C}_{ij}$

Assim temos $P(class_k|object) * \hat{C}_{ij} = P(class_k|object) * P(object) * IoU(BB_{gt}, BB_j)$, que é equivalente a $P(class_k) * IoU(BB_{gt}, BB_j)$

Esse último valor captura a probabilidade de o objeto ser da classe k e quão bem o BB ajusta-se ao objeto

Um mesmo objeto pode ser detectado por BBs associados às células vizinhas da célula responsável. Para remover detecções duplicadas, aplica-se non-maximum supression, usando o score acima para ordenar os BBs.

Seleciona-se o BB com maior score e remove-se overlapping BBs (pode-se adotar um limiar para o IoU)

Do que resultou, seleciona-se o BB com maior score e repete-se a remoção, até não sobrar mais BBs

Depois de chegar a uma lista de BBs finais preditos, ainda é preciso decidir se se trata de detecção correta (classe e localização).

Métrica comumente usada: mAP (mean average precision)
https://github.com/rafaelpadilla/review_object_detection_metrics

Pode ser interessante olhar antes ROC $\times$ Precision-Recall curves
https://acutecaretesting.org/en/articles/precision-recall-curves-what-are-they-and-how-are-they-use