Deep Learning for Vision (DLV)

Object Detection

Denis Coquenet

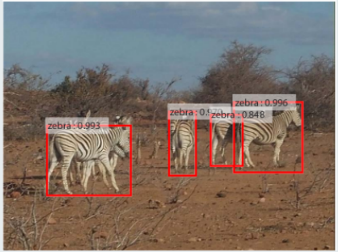
2024-2025

Université de Rennes

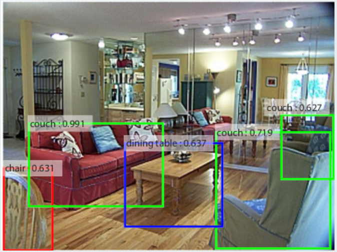
istic Informatique Électronique

Table of contents

What is in the image? and where?



Single class



Multiple classes

Images from [1].

2 tasks:

- Localize all the items with bounding boxes
- Classify them

Why?

- Understanding environments (*e.g.*: autonomous shop)
- Counting (*e.g.*: people in a crowd)
- As first step in complex system (*e.g.*: HTR)

Difficulties

- Classes must be known beforehand
- Number of items to recognize for each class change from one image to another

Formulation

Goal

Learn $f_\theta : \mathcal{X} \to \mathcal{Y}$

Input: an image $x \in \mathbb{R}^{H \times W \times C}$, a set of N_c classes $\mathcal{C}$
Output: $\{(b_{i,1}, b_{i,2}, b_{i,3}, b_{i,4}, c_i)\}_i$
$\{b_{i,j} \in \mathbb{R}\}_j$ the bounding box coordinates in pixels (2 corners or one corner + size)
$c_i \in \mathbb{R}^{\mathsf{N}_c}$ the corresponding class

Evaluation

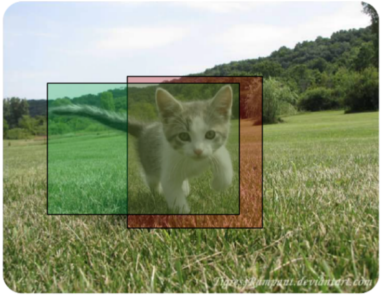
Left: ground truth. Middle: prediction. Right: comparison.

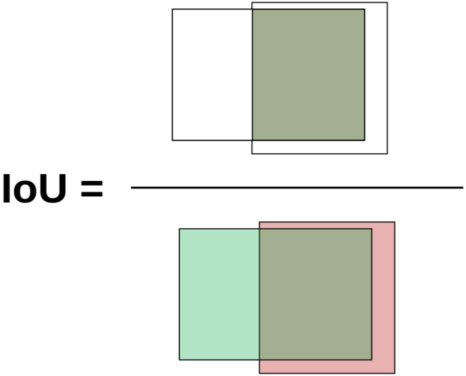
Images from https://docs.kolena.io/metrics

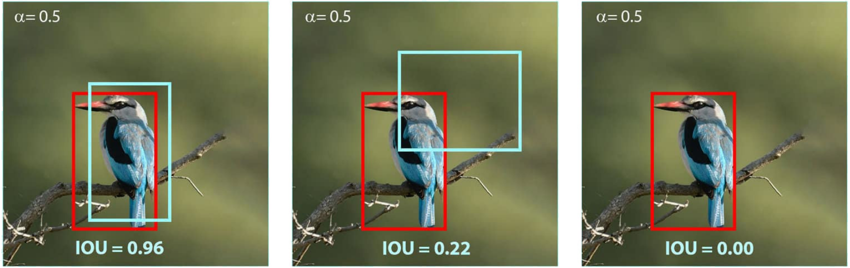
True Positive (TP): correct predictions
True Negative (TN): correct "no" predictions
False Positive (FP): wrong prediction
False Negative (FN): missed prediction

Intersection over Union (IoU, or Jaccard index)

$$\text{IoU}(y, \hat{y}) = \frac{y \cap \hat{y}}{y \cup \hat{y}} = \frac{\text{TP}}{\text{TP} + \text{FN} + \text{FP}}$$







Compute metrics at object level

- TP if IoU $\geq \alpha$ (left)
- FP if IoU $< \alpha$ (middle)
- FN if no prediction (right)

Precision

- How much item predictions were correct? (confidence)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

Recall

- How much of the annotated items have been found?

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- Two metrics for two different aspects

F1 score

$$\text{F1 score} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Recall}}}$$

$$= \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

➤ F1 score high only if both Precision and Recall are high

Intuition

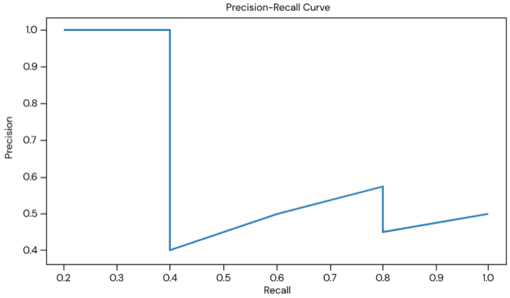
If Precision $\rightarrow 0$ then F1 score $\rightarrow 0$ even if Recall $= 1$
If Recall $\rightarrow 0$ then F1 score $\rightarrow 0$ even if Precision $= 1$

Average Precision (AP)

➤ How the proportion of correct predictions evolves as the model finds the expected elements?

$$\mathsf{AP} = \int p(r)\, dr$$

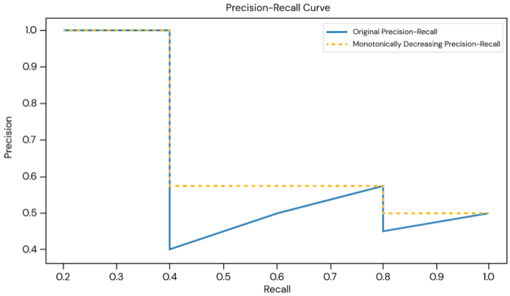
= Area under the curve Precision/Recall



Approximation of the Average Precision

$$\text{AP} = \sum_i (r_{i+1} - r_i) * p_{\text{interp}}(r_{i+1})$$

$$p_{\text{interp}}(r_{i+1}) = \max_{\tilde{r} \geq r_{i+1}} p(\tilde{r})$$



- $r_0 = 0$
- $p(0) = 1$

mean Average Precision (mAP)

- Average AP over the set of classes $\mathcal{C}$

$$\text{mAP} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \text{AP}_c$$

Can be weighted by the number of sample by class

- Average mAP for different IoU thresholds

$$\text{mAP}^{50:95:5} = \frac{1}{10} \sum_{k} \text{mAP}^{\text{IoU}>k}$$

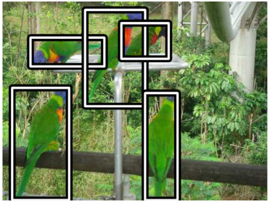
IoU thresholds from 50% to 95% with a step of 5%

- Be careful when comparing!

PASCAL VOC (Visual Object Class)

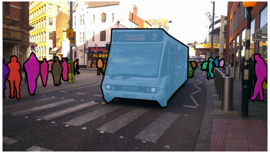
Bounding boxes annotations

- Pascal VOC 2007: 3k train, 3k val, 5k test, 20 classes
- Pascal VOC 2012: 6k train, 6k val, 11k test, 20 classes



MS COCO (MicroSoft Common Objects in COntext)

- 2014: 83k train, 40k val, 40k test, 80 classes
- 2017: 118k train, 5k val, 40k test, 80 classes



➤ more costly annotations than for classification

Two main approaches

Two-stage detectors

- Detection of regions of interest
- Classification of these region

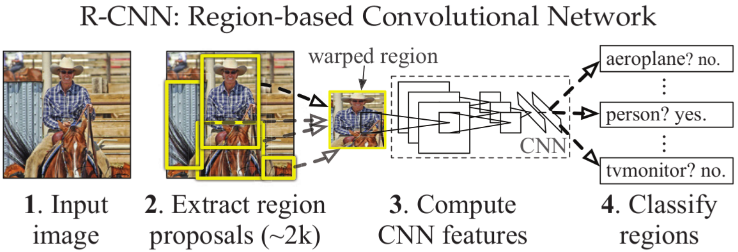
e.g., R-CNN, Faster R-CNN

One-stage detectors

A grid is applied on the image whose all cells are considered as a proposal of region of interest

e.g., SSD, YOLO, RetinaNet

Table of contents

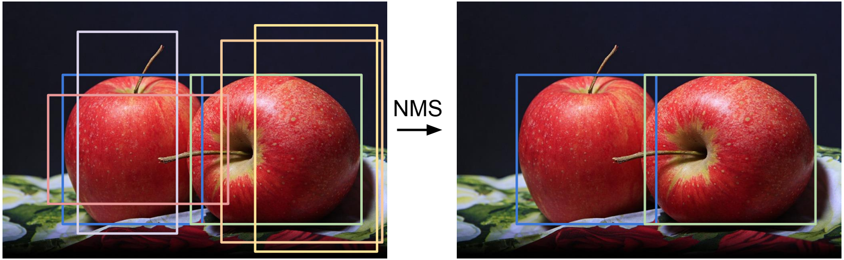


Approach

- Region proposal with selective search (rule-based algorithm)
- Feature extraction per proposal with AlexNet (pre-trained on ImageNet, removing classif. head)
- Classification with class-specific SVMs (trained on AlexNet features)

➤ 2,000 proposed regions!

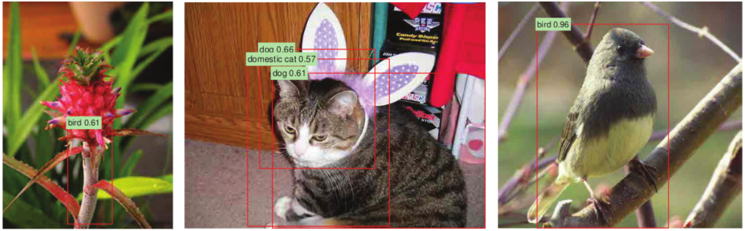
Goal: remove redundant predictions



Algorithm

For each class:

1) Sort the predictions by confidence level
2) Keep the most confident prediction
3) Remove all other predictions which overlap too much (using IoU)
4) Repeat 2) and 3) until there is no more predictions

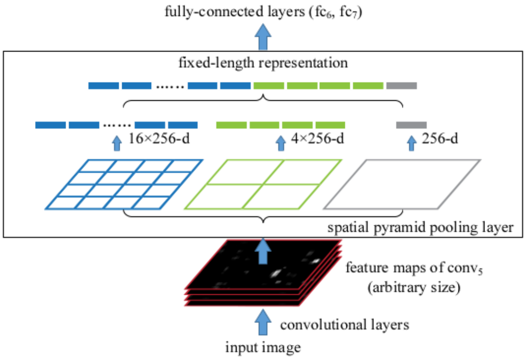


SOTA performance (66% mAP, VOC 2007) but some drawbacks

- Region proposal technique based on rules
  - ➤ Not optimized for a given task/dataset
- Feature extraction performed on all proposals independently
  - ➤ Inference time: 10 to 45 seconds per image on GPU
- SVM trained on top of CNN features
  - ➤ Two trainings

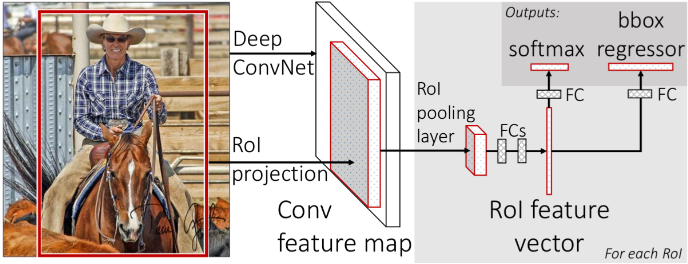
Spatial Pyramid Pooling (2014) [3]

Goal: take input images of arbitrary size with fully-connected layers



Apply max-pooling on some fixed-length grids (adaptive pooling)
➤ Convert any feature maps (H, W, d) into a fixed-length feature vector, here: (21, d)

Fast R-CNN (2015) [4]



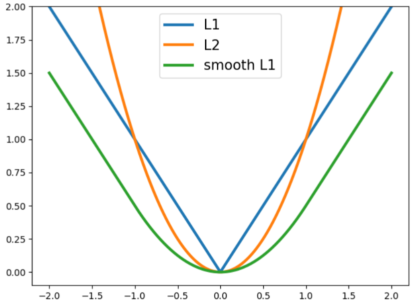
Approach

- Region proposals based on rules
- Feature maps extraction with CNN
- Adaptive (RoI) pooling on feature map crops (proposals)
- Classification + regression (to refine proposals)

➤ Inference time: 1.5s for proposals + 0.3s/image

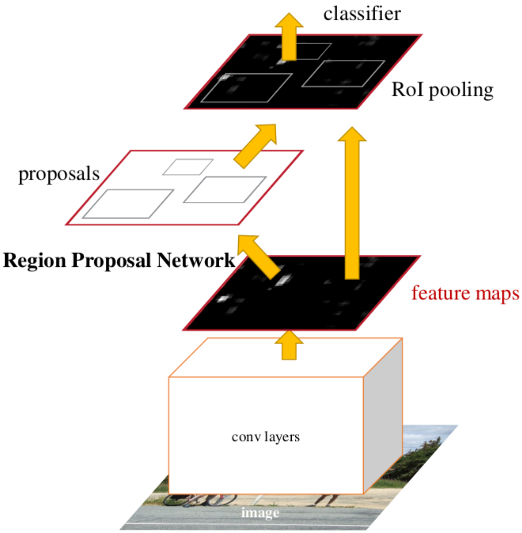
Multitasking

➤ The network is trained to perform multiple tasks
e.g., classification and regression
➤ Tasks can be trained either simultaneously or alternatively
Multitask loss: $\mathcal{L} = \mathcal{L}_{\text{CE}}(\hat{c}, c) + \lambda \sum_i \mathcal{L}_{\text{smoothL1}}(\hat{b}_i, b_i)$



$$\mathcal{L}_{\text{L1}}(\hat{y}, y) = |\hat{y} - y|$$
$$\mathcal{L}_{\text{L2}}(\hat{y}, y) = (\hat{y} - y)^2$$

$$\mathcal{L}_{\text{smoothL1}}(\hat{y}, y) = \begin{cases} 0.5(\hat{y} - y)^2 & \text{if } |\hat{y} - y| < 1 \\ |\hat{y} - y| - 0.5 & \text{otherwise} \end{cases}$$

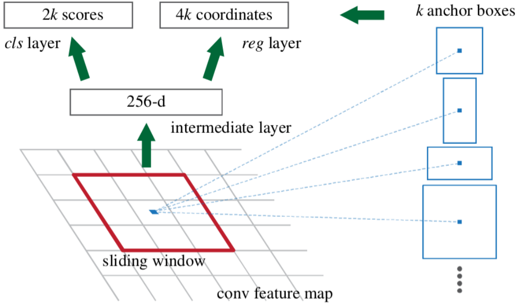


Faster R-CNN = Fast R-CNN + RPN

- No more rule-based proposal algorithm
  ➤ Region Proposal Network (RPN)
- Common CNN backbone

➤ RPN as CNN to preserve shift-equivariance property

Region Proposal Network



9 anchors per 2D position

Classification

object/non-object classification

positive if	max IoU with a GT
	IoU > 0.7 with a GT
negative if	IoU < 0.3 with all GT
discarded	otherwise

➤ Imbalanced classification (too many negatives): randomly sample 256 anchors (half positive)

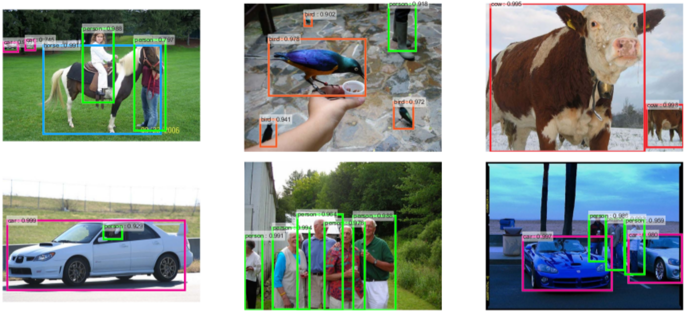
Regression

Determine shift (position and size) with respect to the anchor

- One regressor per anchor
- Trained for positive samples only

➤ Another hybrid loss

Faster R-CNN (2015) [1]

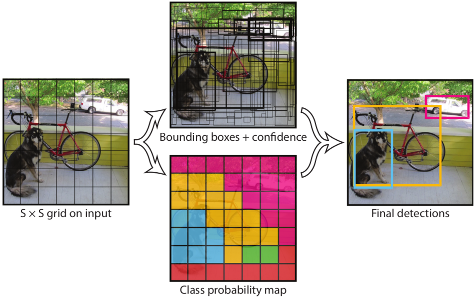


- 0.2s / image with a VGG backbone

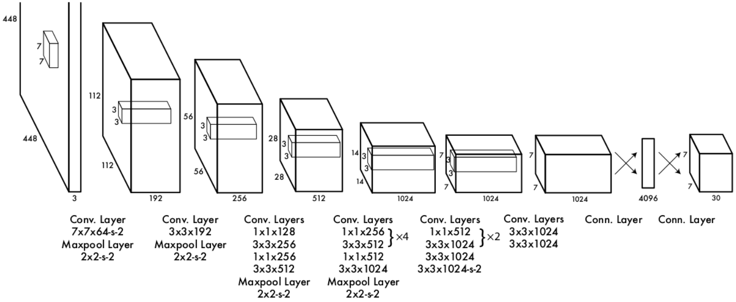
Architecture	mAP (%) on Pascal VOC 2007
R-CNN	66.0
Fast R-CNN	66.9
Faster R-CNN	69.9

Table of contents

- You Only Look Once (YOLO)

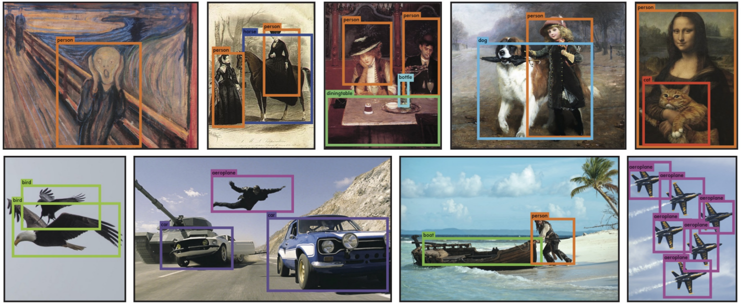


- For each grid cell: B bounding box predictions $\hat{b} = (x, y, h, w, k)$ + N_c class probabilities c_i
- k: confidence level measuring $\mathrm{P(Object)} \times \mathrm{IoU}(\hat{b}, b)$
- $c_i = \mathrm{P}(\mathrm{Class}_i|\mathrm{Object})$



- End-to-end CNN, pre-trained on ImageNet (S=7, B=2, $N_c = 20$)
- NMS used at inference time
- Real-time system (trained on Pascal VOC 2007+2012):
  VOC 2007: 45 FPS for 63.4% mAP
  (Faster R-CNN: 7 FPS for 73.2% mAP)

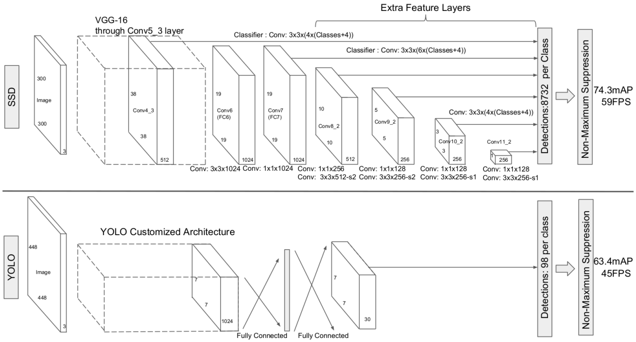
YOLO (2016) [5]



Limitations

- Limited number of predictions per cell (B)
  - ➤ people in crowd
- Only one class per cell
  - ➤ Cannot recognize multiple objects of different classes if there are too close

- Single-Shot Detector (SSD)



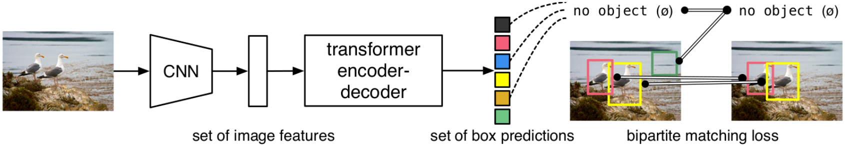
Multi-scale detection

- SSD: $38^2 \times 4 + 19^2 \times 6 + 10^2 \times 6 + 5^2 \times 6 + 3^2 \times 4 + 1^2 \times 4 = 8,732$
- YOLO: $7^2 \times 2 = 98$

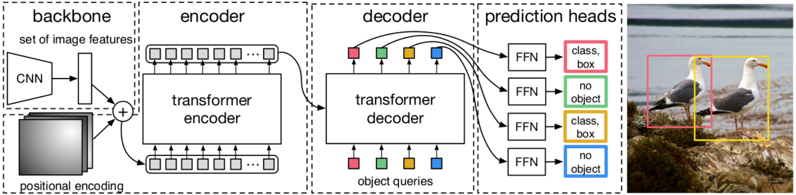
➤ Multi-scale detection strategy

Prediction from layer						
conv4_3	conv7	conv8_2	conv9_2	conv10_2	conv11_2	mAP (%)
✓	✓	✓	✓	✓	✓	74.3
✓	✓	✓	✓	✓	✗	74.6
✓	✓	✓	✓	✗	✗	73.8
✓	✓	✓	✗	✗	✗	70.7
✓	✓	✗	✗	✗	✗	64.2
✗	✓	✗	✗	✗	✗	62.4

- DEtection TRansformer (DETR)



Set of N learned object queries (must be high enough)
All predictions in parallel
- No need for anchors nor NMS



Training strategy

- Dedicated bipartite matching loss
- Auxiliary loss after each decoding layer
- CNN backbone pre-trained on ImageNet
- Slow convergence: 300 epochs, 16 GPU V100, 3 days (10x more than Faster R-CNN)

➤ 10 FPS / 44.9 % mAP on COCO 2017

➤ 28 FPS / 42.0 % mAP with lighter backbone

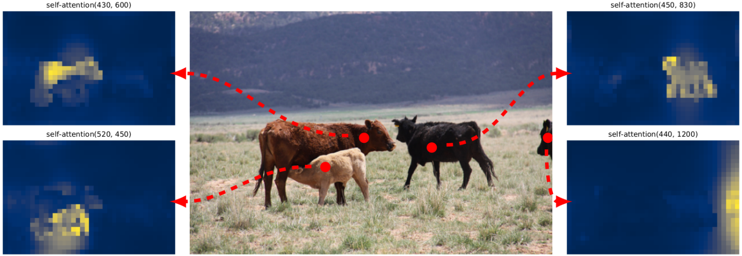
Adaptation of 1D positional encoding to 2D

$$\begin{aligned}
\mathrm{PE}(p_x, p_y, 2k) &= \sin(w_k \cdot p_x) \\
\mathrm{PE}(p_x, p_y, 2k+1) &= \cos(w_k \cdot p_x) \\
\mathrm{PE}(p_x, p_y, d_{\text{model}}/2 + 2k) &= \sin(w_k \cdot p_y) \\
\mathrm{PE}(p_x, p_y, d_{\text{model}}/2 + 2k + 1) &= \cos(w_k \cdot p_y)
\end{aligned}$$

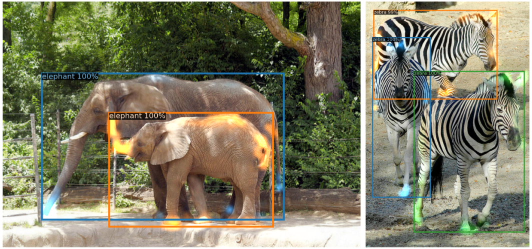
$\forall k \in [0, d_{\mathrm{model}}/4]$, with $w_k = 1/10000^{2k/d_{\mathrm{model}}}$

- First half dimensions dedicated to horizontal axis
- Second half dimensions dedicated to vertical axis

Self-attention



Cross attention



Two-approaches

- While 2-step approaches used to have better performance than single-stage detector, this is not true anymore
- Single-stage detector faster
- Transformer alternative competitive without anchor/NMS

Limitations of object detection

- Bounding boxes may not be accurate enough (tumor detection)
- Number of items to recognize conditioned by hyperparameters (# of anchors / cell, # of object queries)

[1] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: *Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems*. 2015, pp. 91–99.

[2] Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation". In: *2014 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 2014, pp. 580–587.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition". In: *13th European Conference on Computer Vision*. Vol. 8691. 2014, pp. 346–361.

[4] Ross B. Girshick. "Fast R-CNN". In: *2015 IEEE International Conference on Computer Vision*. 2015, pp. 1440–1448.

[5] Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, and Ali Farhadi. "You Only Look Once: Unified, Real-Time Object Detection". In: *IEEE Conference on Computer Vision and Pattern*. 2016, pp. 779–788.

[6] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott E. Reed, Cheng-Yang Fu, and Alexander C. Berg. "SSD: Single Shot MultiBox Detector". In: *14th European Conference on Computer Vision*. Vol. 9905. Lecture Notes in Computer Science. 2016, pp. 21–37.

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. "End-to-End Object Detection with Transformers". In: *16th European Conference on Computer Vision*. Vol. 12346. 2020, pp. 213–229.