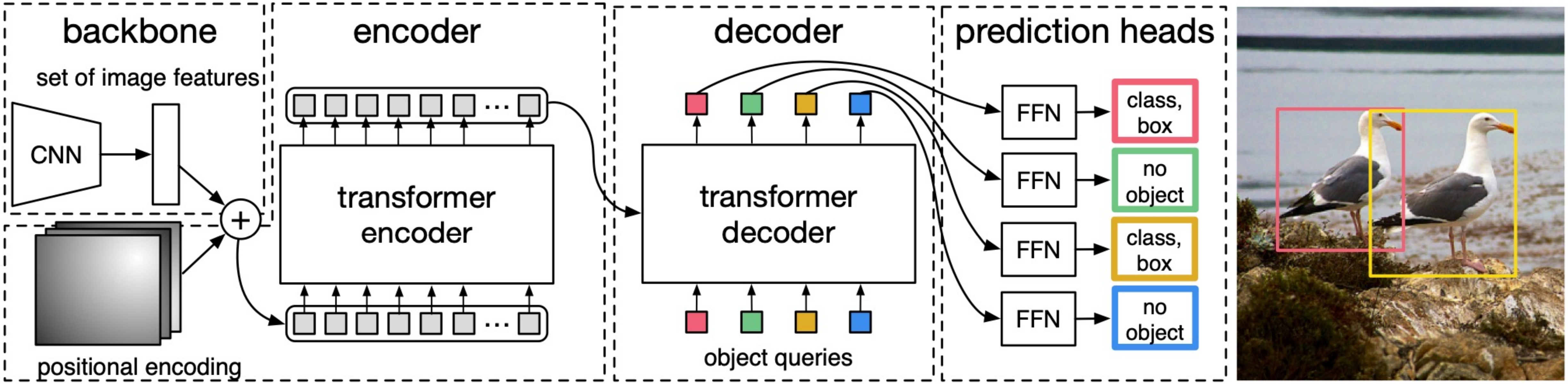
Homework #1
Object Detection

R10942198 林仲偉

1. (5%) Draw the architectures for both CNN-based and Transformer-based methods

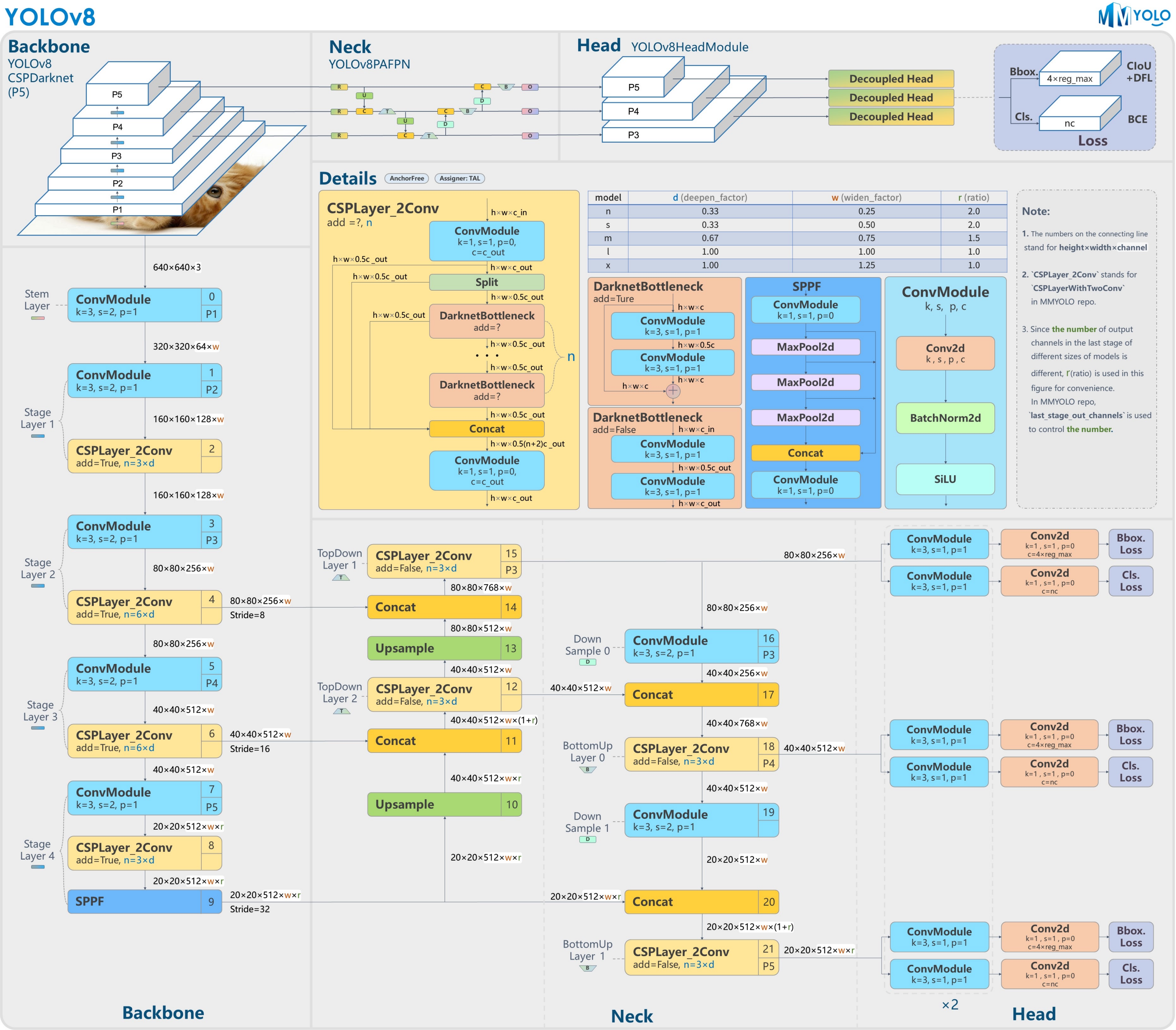
- Transformer-based method: DETR



Ref: https://arxiv.org/pdf/2005.12872.pdf

- CNN-based method: yolov8

Ref: https://github.com/open-mmlab/mmyolo/blob/dev/configs/yolov8/README.md



2. (10%) Report and compare the performance of two methods on validation set

metrics	DETR	yolov8
MAP	0.4121	**0.4894**
MAP_50	0.6974	**0.7477**
MAP_75	0.4107	**0.5176**
MAP_large	0.5575	**0.6082**
MAP_medium	0.2930	**0.3622**
MAP_small	0.0918	**0.1379**
MAR_10	0.4275	**0.4804**
MAR_100	0.4920	**0.5493**
MAR_small	0.1442	**0.2622**
MAR_medium	0.3946	**0.4577**
MAR_large	0.6286	**0.6631**

Comparison:
Yolov8 大獲全勝，所有 metrics 皆比DETR 優秀

3. (10%) Report the implementation details of both methods

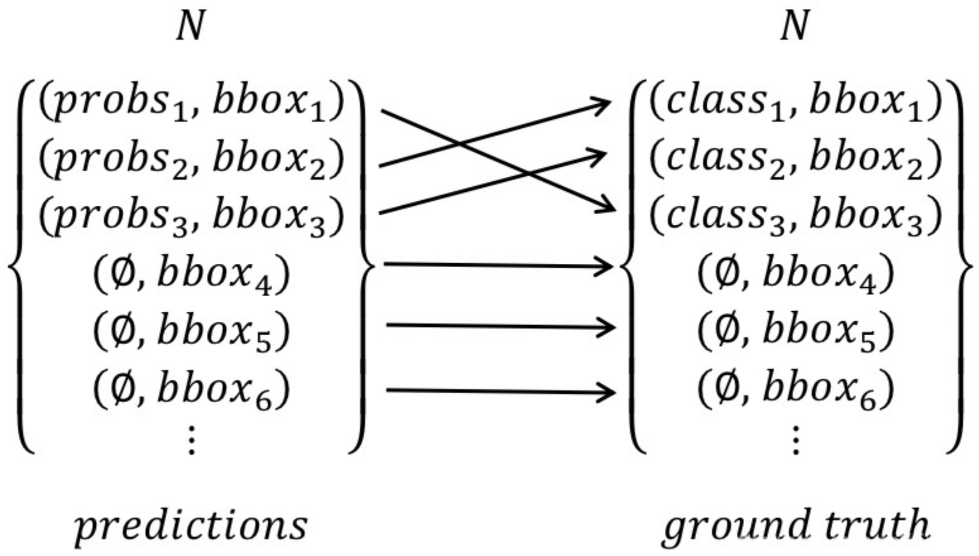
DETR: Loss function

- Bipartite matching loss:

$$\operatorname{argmin}_{\sigma \in \sigma_N} \sum_{i}^{N} L_{match}(y_i, \hat{y}_{\sigma(i)})$$

- 在 N! 種組合中，使用 Hungarian 算法找出使得 L_{match} 最小的組合。

$$L_{match} = -\mathbf{1}_{\{c_i \neq \phi\}} \hat{p}_{\sigma(i)}(c_i) + \mathbf{1}_{\{c_i \neq \phi\}} L_{box}(b_i, \hat{b}_{\sigma(i)})$$



Ref: https://blog.csdn.net/zjuPeco/article/details/107209584

3. (10%) Report the implementation details of both methods

DETR - Other training details:

- Pre-train weight: detr-r101
- Data augmentation: Random select from:
  a) Random resize to scales = [480, 512, 544, 576, 608, 640, 672, 704, 736, 768, 800]
  b) Random crop with size=[384, 600], then do (a).
- Loss: Bipartite matching loss
- Post-processing: filter out background bounding boxes.
- Other training config:
  - epoch=600, learning rate=1e-4, weight decay=1e-4, optimizer=AdamW, batch=2

3. (10%) Report the implementation details of both methods

Yolov8: Loss functions

Complete IoU Loss $\mathcal{L}_{box}$：

- 比 IoU loss 多考慮 bounding box 的寬、高而產生的 penalty.

$$\mathcal{L}_{CIoU} = 1 - IoU + \frac{\rho^2\left(\mathbf{b}, \mathbf{b}^{gt}\right)}{c^2} + \alpha v$$

$$v = \frac{4}{\pi^2}\left(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h}\right)^2$$

α: hyper-parameter
w, h: bounding box 寬、高
IoU: predict 和 ground truth bounding box 的 交集除以聯集

$$IoU = \frac{|A \cap B|}{|A \cup B|}$$

Ref: https://zhuanlan.zhihu.com/p/94799295

3. (10%) Report the implementation details of both methods

Yolov8: Loss functions

Distributed Focal Loss $\mathcal{L}_{dfl}$:

- 在 object detection 中，標註位置的 ground truth 通常不會離預測位置太遠，故希望model 能夠也學習到標註位置附近的資訊。

$$\mathbf{DFL}(\mathcal{S}_i, \mathcal{S}_{i+1}) = -\left((y_{i+1} - y)\log(\mathcal{S}_i) + (y - y_i)\log(\mathcal{S}_{i+1})\right)$$

$$\mathcal{S}_i = \frac{y_{i+1}-y}{y_{i+1}-y_i}, \mathcal{S}_{i+1} = \frac{y-y_i}{y_{i+1}-y_i}$$

Cross Entropy Loss $\mathcal{L}_{ce}$:

$$\text{Loss} = -\sum_{i=1}^{\substack{\text{output}\\\text{size}}} y_i \cdot \log \hat{y}_i$$

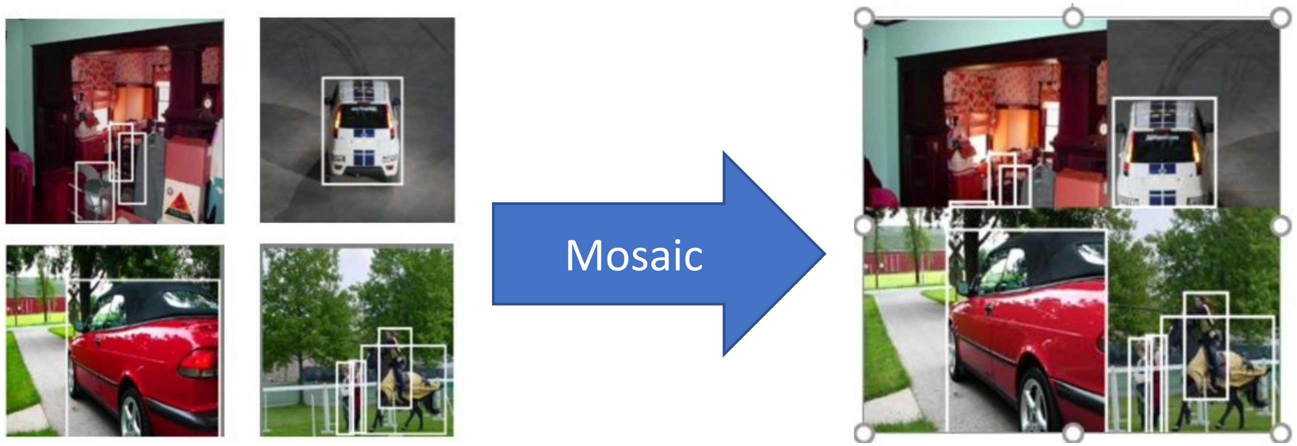
$\hat{y}_i$: predict logit of class i

y_i: ground truth of class i

3. (10%) Report the implementation details of both methods

Yolov8: Data Augmentation

Mosaic: 將數張不同的資料圖擺放在四角，拼接成一張新圖片 。



Mix-up: 將兩張圖片 x_1, x_2 以及其標註 y_1, y_2 依照比例 $\lambda\epsilon[0,1]$ 做插值，生成新的圖片 x_{mix} 以及標註 y_{mix}

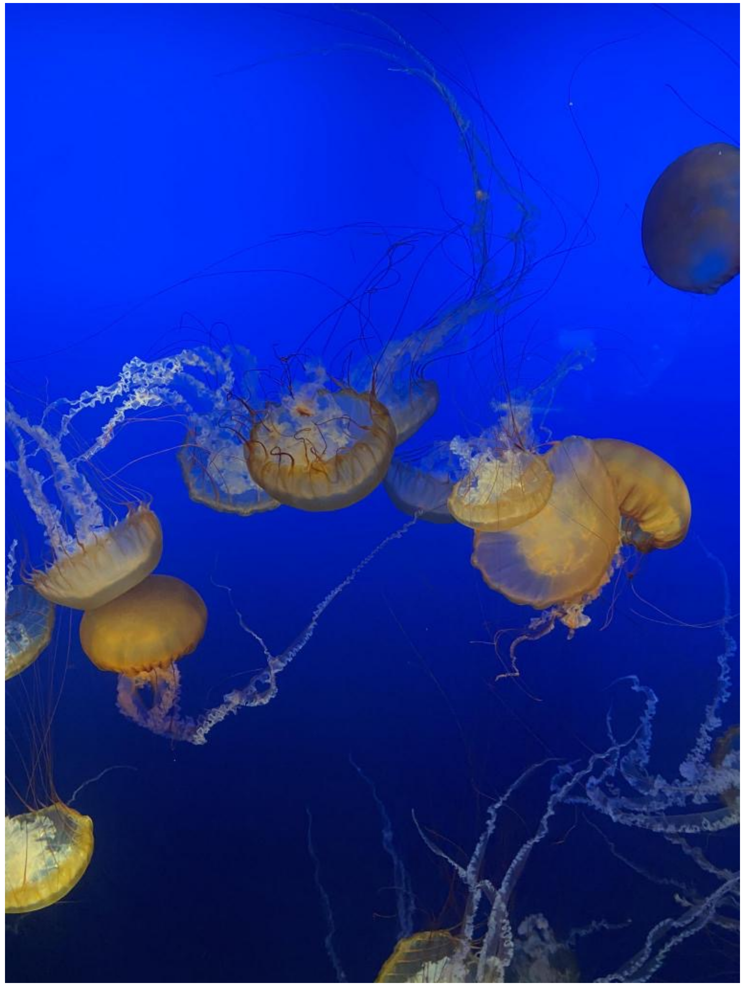
$$x_{mix} = \lambda x_1 + (1-\lambda)x_2$$
$$y_{mix} = \lambda y_1 + (1-\lambda)y_2$$

3. (10%) Report the implementation details of both methods

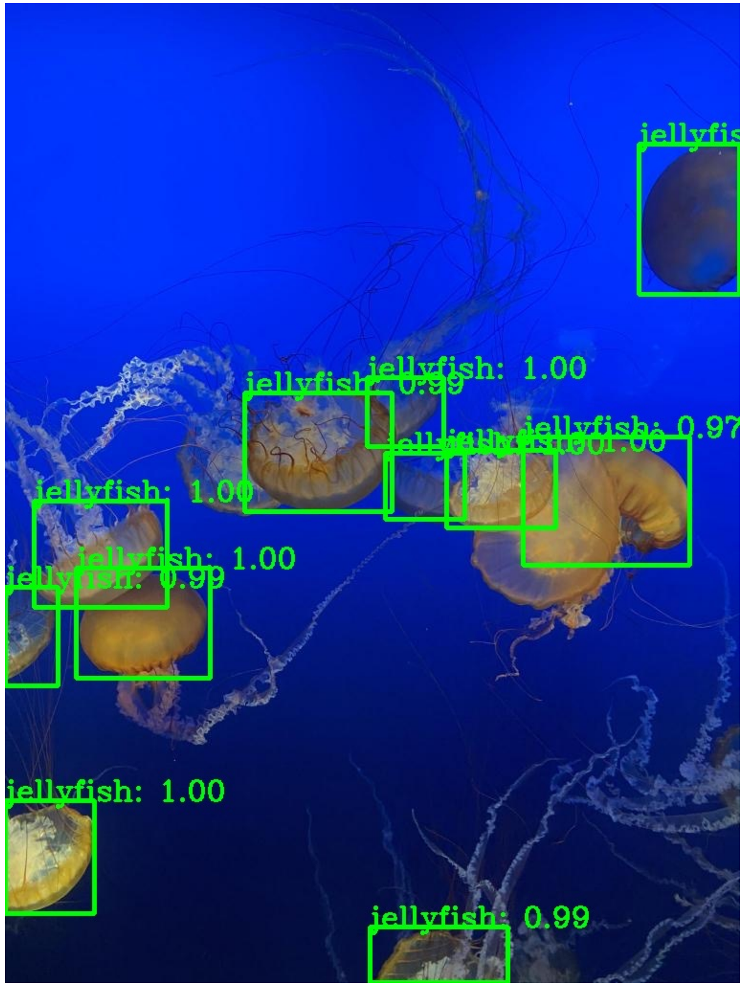
Yolov8 - Other training details:

- Pre-train weight: yolov8x.pt
- Data augmentation: Random select from:
  a) Mosaic
  b) Mix-up
- Loss: $7.5\times\mathcal{L}_{box}+0.5\times\mathcal{L}_{ce}+1.5\times\mathcal{L}_{dfl}$
- Post-processing: filter out the bounding boxes whose confidence score less than 0.25.
- Other training config:
  - epoch=300, learning rate=0.01, weight decay=1e-4, optimizer=SGD, batch=16

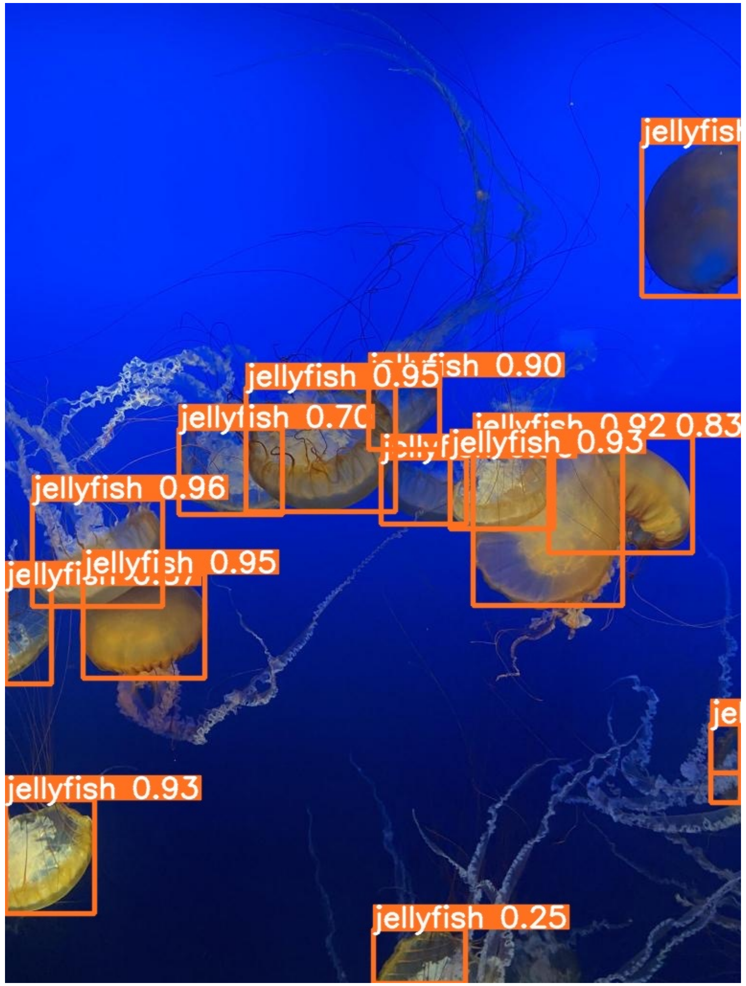
4. (5%) Visualization: draw the bounding boxes of two methods on this test image.



origin



DETR



yolov8