CVPDL HW1

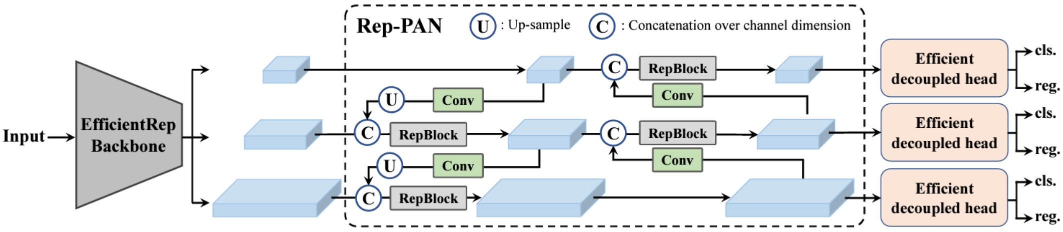
R11922196 林佑鑫

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

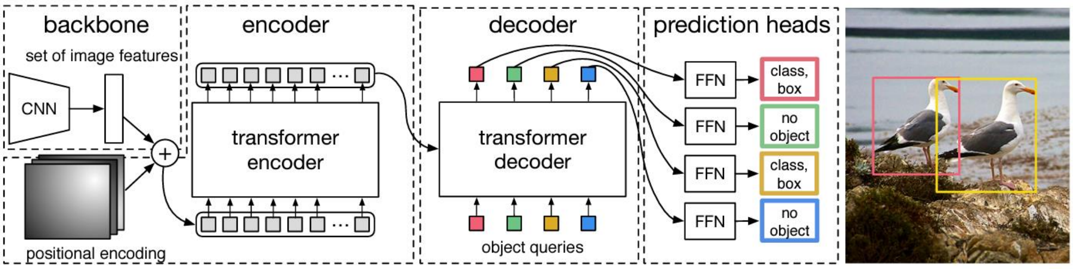
a. The graph should be brief and clear

b. It would be fine to straight copy the figure from the paper

CNN-based: YOLOv6



Transformer-based: DETR

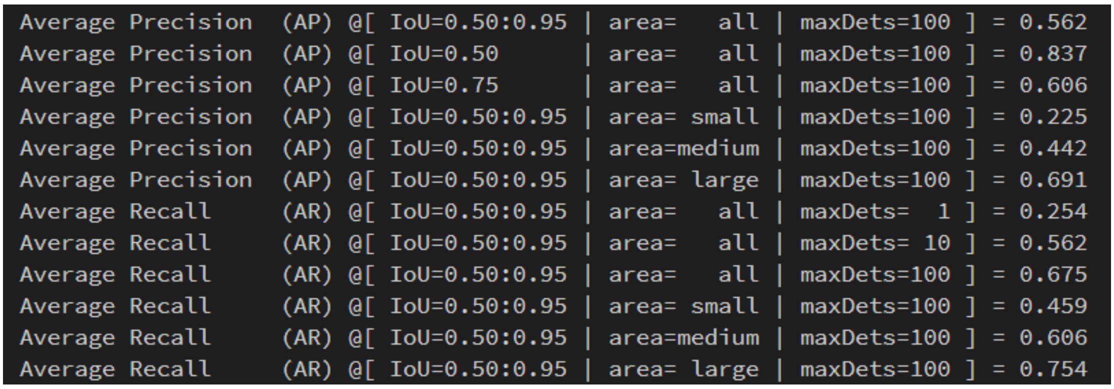


2. (10%) Report and compare the performance of two methods on validation set

a. at least with mAP@[50:5:95], mAP@50, mAP@75

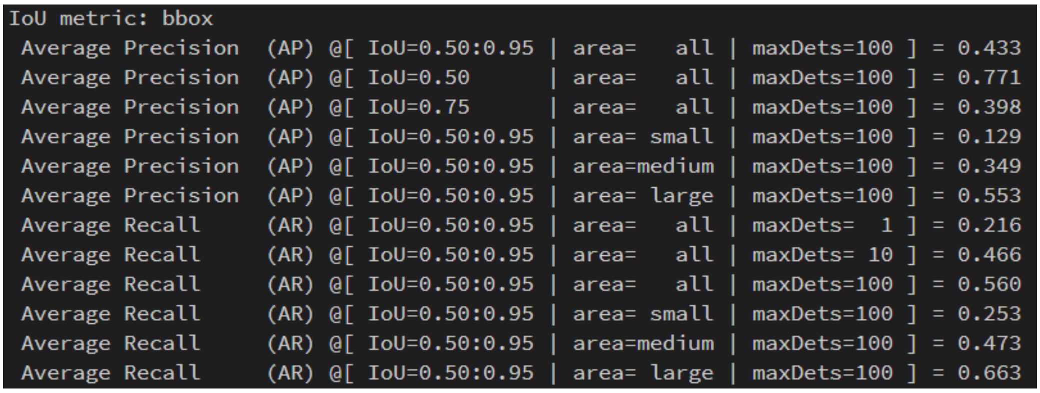
b. use table to organize the results

CNN-based: YOLOv6



```
Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.562
Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.837
Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.606
Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.225
Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.442
Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.691
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.254
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.562
Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.675
Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.459
Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.606
Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.754
```

Transformer-based: DETR



```
IoU metric: bbox
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.433
 Average Precision  (AP) @[ IoU=0.50      | area=   all | maxDets=100 ] = 0.771
 Average Precision  (AP) @[ IoU=0.75      | area=   all | maxDets=100 ] = 0.398
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.129
 Average Precision  (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.349
 Average Precision  (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.553
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=  1 ] = 0.216
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets= 10 ] = 0.466
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=   all | maxDets=100 ] = 0.560
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.253
 Average Recall     (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = 0.473
 Average Recall     (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = 0.663
```

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

a. Ex: augmentation, loss function, cross validation method, ...etc.

CNN-based: YOLOv6

Network Design:

- Backbone: RepBlock for small networks, CSPStackRep Block for large models.
- Neck: PAN topology.
- Head: Efficient Decoupled Head.

Label Assignment:

- TAL.

Loss Function:

- Classification loss: VariFocal Loss.
- Box regression loss: SIoU/GIoU.

Industry-handy improvements:

- Self-distillation.

Pretrained model:

- Pretrained on COCO2017 dataset.

Github repo:

- https://github.com/meituan/YOLOv6

Transformer-based: DETR

Network Design:

- Backbone: CNN to 32x down-sampling feature map
- Transformer encoder: 1x1 convolution and collapse the spatial dimensions into one dimension sequence as input.
- Transformer decoder: Follows the standard architecture of the transformer.
- Prediction feed-forward networks (FFNs): predict BBox, class.

Auxiliary decoding losses:

- Use auxiliary losses in decoder during training.

Github repo:

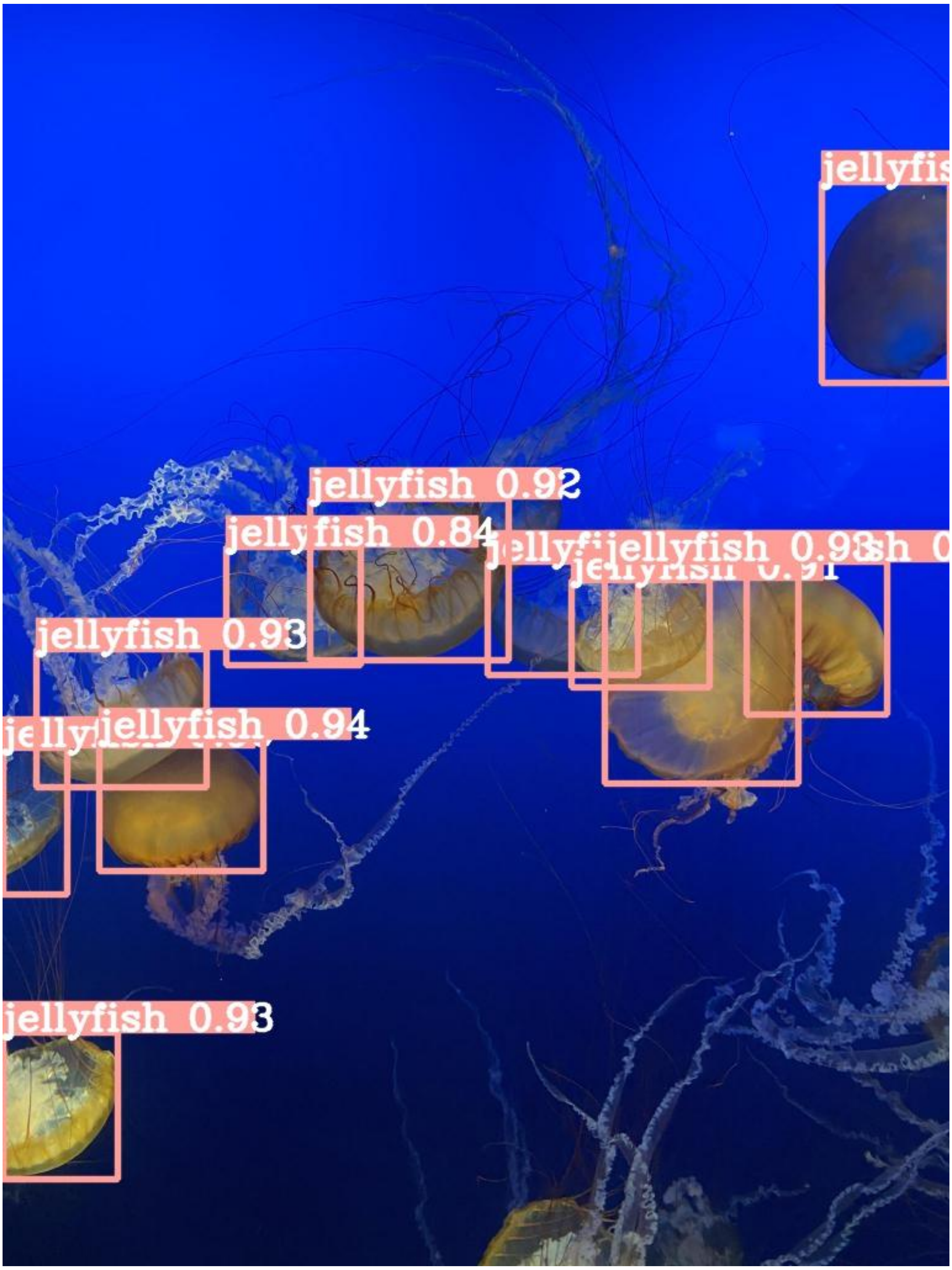
- https://github.com/facebookresearch/detr

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

a. IMG_2574_jpeg_jpg.rf.ca0c3ad32384309a61e92d9a8bef87b9

b. Result should be something like this

CNN-based: YOLOv6



Transformer-based: DETR