1. The architecture of the object detector, detr

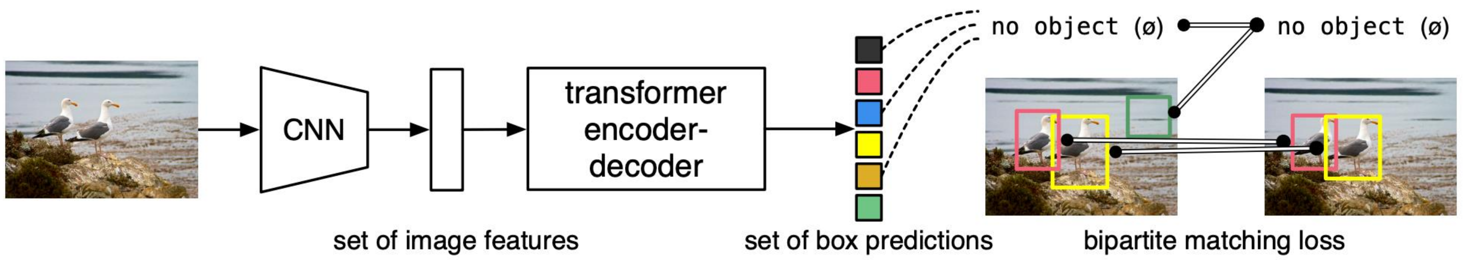


Fig. 1: DETR directly predicts (in parallel) the final set of detections by combining a common CNN with a transformer architecture. During training, bipartite matching uniquely assigns predictions with ground truth boxes. Prediction with no match should yield a "**no object**" (∅) class prediction.

2. Implementation details

epochs = 132

lr = 1e-4

resume = './detr-r50_17.pth'

cocopath = './HW1_2024_dataset'

./models/detr.py →num_classes = 17

./datasets/coco.py

```
def build(image_set, args):
    root = Path(args.coco_path)
    assert root.exists(), f'provided COCO path {root} does not exist'
    mode = 'instances'
    PATHS = {
        "train": (root / "train2017"/'images', root / "annotations" / f'{mode}_train2017.json'),
        "val": (root / "valid2017"/'images', root / "annotations" / f'{mode}_val2017.json'),
    }
```

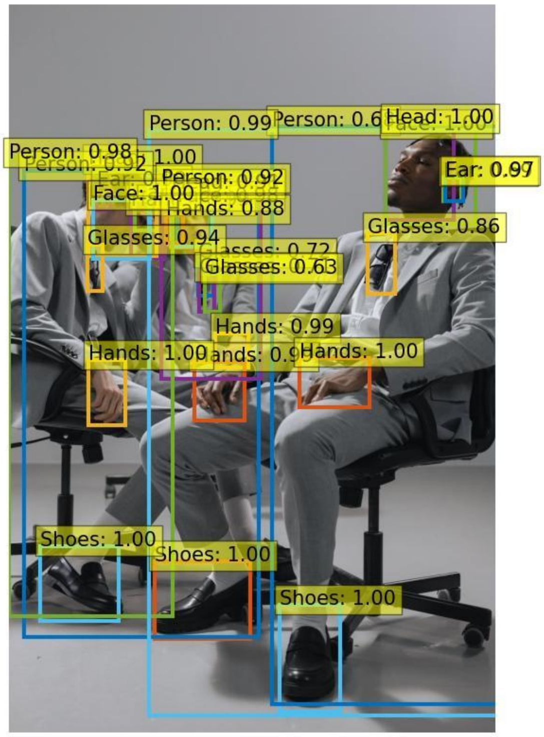
3. Table of your performance for validation set (AP, AP50, AP75)

```
mAP50 (across all instances): 0.7697
mAP75 (across all instances): 0.5447
mAP50-95 (across all instances): 0.5054
aps_per_instance is 0.5054

Process finished with exit code 0
```

epochs	mAP(50-95)	mAP50	mAP75
130	0.4467	0.7122	0.4665
132	0.5054	0.7697	0.5447
134	0.5029	0.7684	0.5407

4.Visualization and discussion



Long tail effect: Although it is not absolute, it can generally be observed from the chart that having more training data tends to have a positive impact on the performance of the mAP50-95 score.

Class ID	Train Count	mAP50-95
0	3676	0.51221915
1	2080	0.454894565
2	80	0.310729792
3	2399	0.56663667
4	36	0.456041667
5	169	0.374939178
6	211	0.10000188
7	1255	0.201501107
8	514	0.553172853
9	752	0.256473611
10	251	0.387189935
11	4201	0.33893176
12	3223	0.490931184
13	31	0.38516129
14	1257	0.209334071
15	70	0.114238095
16	131	0.313933904