You can modify this report template, and upload your results in PDF format. Reports in other forms/formats will result in ZERO point. Reports written in either Chinese or English are both acceptable. The length of your report should NOT exceed 6 pages (**excluding bonus**).

Name: XXX Dep.:電信碩一 Student ID:R06666666

1. (5%) Print the network architecture of your YoloV1-vgg16bn model and describe your training config. (optimizer,batch size….and so on)

```
VGG(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(2, 2), padding=(1, 1))
    (1): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace)
    (3): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (4): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): ReLU(inplace)
    (6): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (7): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (8): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (9): ReLU(inplace)
    (10): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (11): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (12): ReLU(inplace)
    (13): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (14): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (15): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (16): ReLU(inplace)
    (17): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (18): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (19): ReLU(inplace)
    (20): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (21): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (22): ReLU(inplace)
    (23): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (24): Conv2d(256, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (25): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (26): ReLU(inplace)
    (27): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (28): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (29): ReLU(inplace)
    (30): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (31): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (32): ReLU(inplace)
    (33): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
    (34): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (35): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (36): ReLU(inplace)
    (37): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (38): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (39): ReLU(inplace)
    (40): Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (41): BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (42): ReLU(inplace)
    (43): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)
  )
  (yolo): Sequential(
    (0): Linear(in_features=25088, out_features=4096, bias=True)
    (1): ReLU()
    (2): Dropout(p=0.5)
    (3): Linear(in_features=4096, out_features=1274, bias=True)
  )
)
```

Batch_size : 24

learing rate:

0~10ep : 1e-4

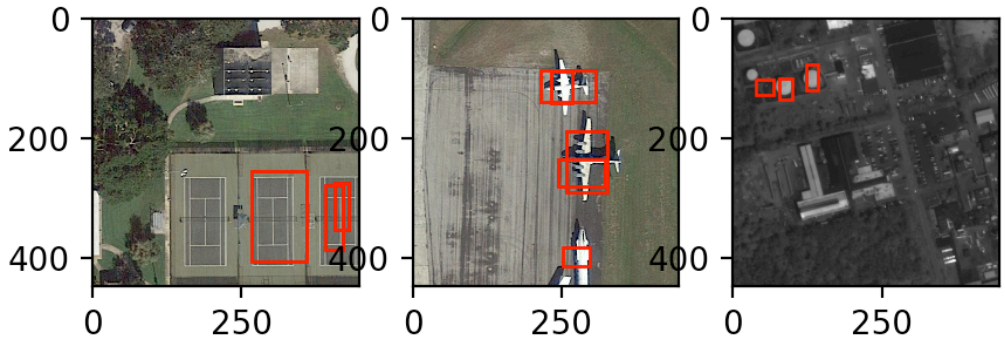
10~30ep : 5e-5

30~60ep : 1e-5

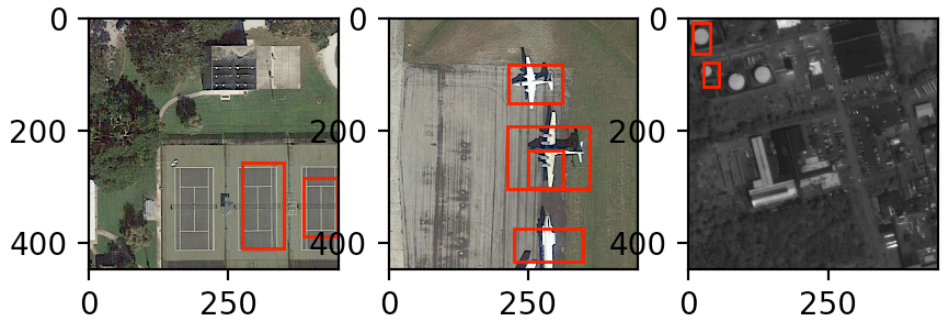
60~ ep : 5e-6

2. (10%) Show the predicted bbox image of “val1500/0076.jpg”, “val1500/0086.jpg”, “val1500/0907.jpg” during the early, middle, and the final stage during the training stage. (For example, results of 1st, 10th, 20th epoch)

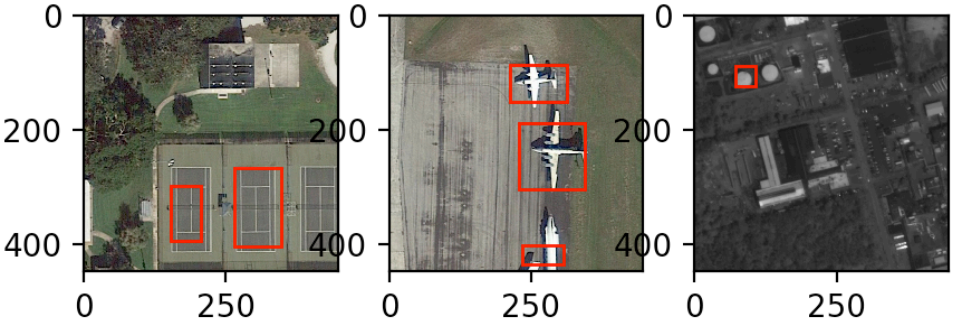
EPOCH:10



EPOCH : 30



EPOCH:80



3. (10%) Implement an improved model which performs better than your baseline model. Print the network architecture of this model and describe it.

   來不及，太晚才找到BUG了～～～

4. (10%) Show the predicted bbox image of “val1500/0076.jpg”, “val1500/0086.jpg”, “val1500/0907.jpg” during the early, middle, and the final stage during the training process of this improved model.

5. (15%) Report mAP score of both models on the validation set. Discuss the reason why the improved model performs better than the baseline one. You may conduct some experiments and show some evidences to support your reasoning.

```
classname: plane
ap:  0.09090909090909091
classname: baseball-diamond
ap:  0.0
classname: bridge
ap:  0.0
classname: ground-track-field
ap:  0.045454545454545456
classname: small-vehicle
ap:  0.09090909090909091
classname: large-vehicle
ap:  0.007215007215007215
classname: ship
ap:  0.012987012987012986
classname: tennis-court
ap:  0.009404388714733543
classname: basketball-court
ap:  0.09090909090909091
classname: storage-tank
ap:  0.022727272727272728
classname: soccer-ball-field
ap:  0.004545454545454546
classname: roundabout
ap:  0.0
classname: harbor
ap:  0.0028116213683223993
classname: swimming-pool
ap:  0.0
classname: helicopter
ap:  0.0
classname: container-crane
ap:  0.0
map: 0.02361703598372635
```

6. bonus (5%) Which classes prediction perform worse than others? Why? You should describe and analyze it.

```
defaultdict(<class 'int'>, {'baseball-diamond': 515, 'bridge': 2114, 'harbor': 7457, 'ship': 34585, 'small-vehicle':
     116228, 'soccer-ball-field': 590, 'large-vehicle': 23746, 'plane': 8723, 'tennis-court': 3279, 'container-crane
    ': 136, 'swimming-pool': 1977, 'ground-track-field': 621, 'roundabout': 537, 'storage-tank': 5199, '
    basketball-court': 661, 'helicopter': 434})
```

這是training dataset中的各個class個數，從數量來看，container-crane數量最少，所以我覺得這個類別的prediction perform會最差。

且從圖片中看container-crane其實感覺會有點像large-vehicle，而large vehicle數量又很多，感覺機器很容易將其分類成vehicle，而我的機器確實將其分成了small-vehicle.

reference :

https://blog.csdn.net/c20081052/article/details/80236015

老師投影片