Model:

使用 Resnet18 作為 pretrain model，會使用 Resnet18 作為 pretrain model 的原因是因為用其他 Model 的運算時間太久，無法快速的微調數值，

Data Loader:

Batch size = 32

Data Augmentation:

因為資料量夠大，所以沒有像第三次作業使用這麼多的圖片變更

```
transforms_train = transforms.Compose([
    transforms.Resize((256,256)),
    transforms.RandomCrop((224,224)),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485,0.456,0.406],std=[0.229,0.224,0.225]),
])
#############################################################################
#                                                          End of your code
#############################################################################

# For VAL, TEST
#############################################################################
#    TODO: use transforms.xxx method to do some data augmentation
#    This one is for validate and test,
#    NOTICE some operation we usually not use in this part
#############################################################################
transforms_test = transforms.Compose([
        transforms.Resize((256,256)),
        transforms.CenterCrop((224,224)),
        transforms.ToTensor(),
        transforms.Normalize(mean=[0.485,0.456,0.406],std=[0.229,0.224,0.225]),
```

Define loss and optimizer:

```
    # TODO: Define loss and optmizer functions
    criterion1, criterion2 = nn.CrossEntropyLoss(), nn.BCEWithLogitsLoss()
    optimizer = torch.optim.Adam(model.parameters(), lr=1e-3)
    scheduler = optim.lr_scheduler.StepLR(optimizer, step_size=20, gamma=0.1)
```

10epoch:1e-3

10epoch:1e-4

10epoch:1e-5

下面是我的圖: