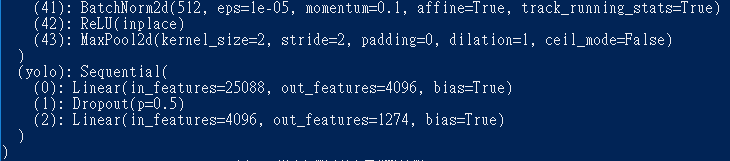
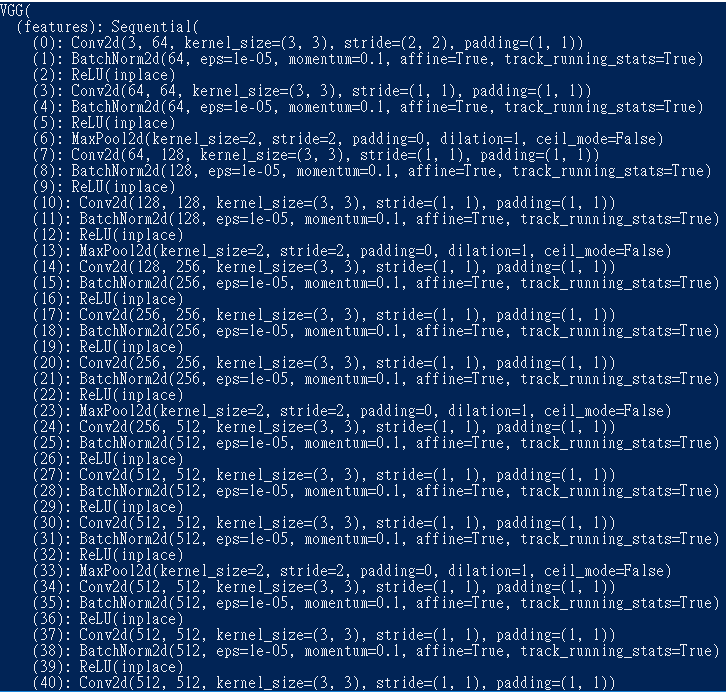
Name: 李俊賢 Dep.:資工碩一 Student ID:R07922103

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



Model：Model主要是在VGG16bn之後接上一層fully connected layer(7\*7\*512 to 4096)、 一層dropout(p=0.5)、及最後再接上一層fully connected layer(4096 to 7\*7\*26)， 之後再rezise成7\*7\*26

Optimizer：Optimizer是使用SGD，其中lr=0.001、momentum=0.9、weight\_decay=0.0001

Batch Size：32

NMS\_confidence\_threshold：0.01

NMS\_IOU\_threshold：0.5

1. (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)

|  |  |  |  |
| --- | --- | --- | --- |
| image  epoch | val1500/0076.jpg | val1500/0086.jpg | val1500/0907.jpg |
| epoch1  (mAP=0.0011) |  |  |  |
| epoch81  (mAP=0.0761) |  |  |  |
| epoch176  (mAP=0.1134) |  |  |  |

1. (10%) Implement an improved model which performs better than your baseline model. Print the network architecture of this model and describe it.

Improved Model：Improved Model基本架構跟Base Model一樣，VGG16bn沒有做更動，

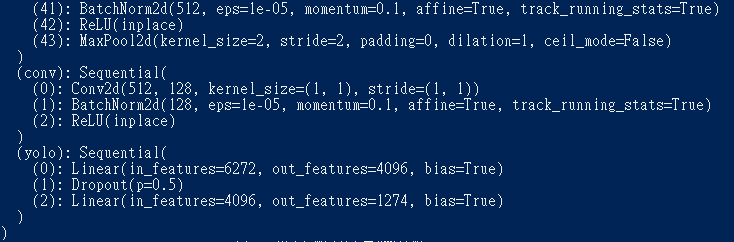
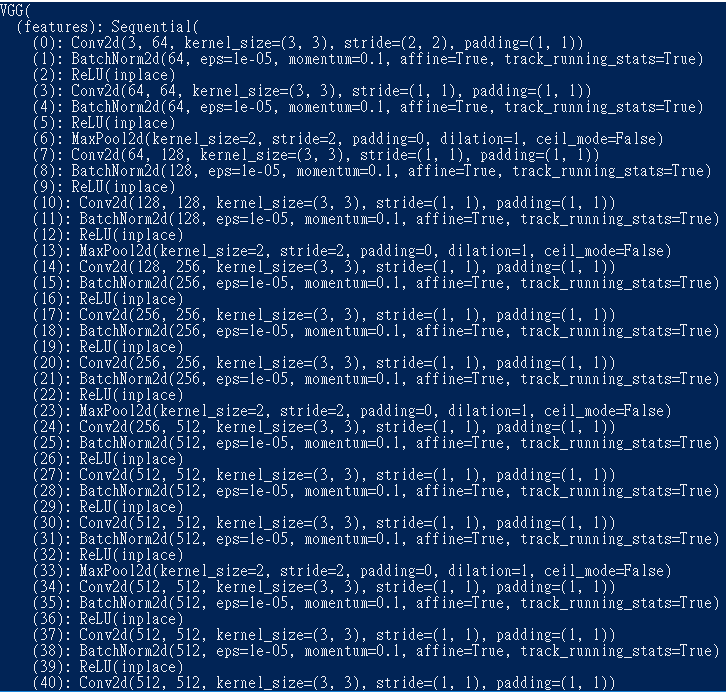
yolo部分也沒更動，只有在VGG16bn與yolo層之間，加了一個1\*1的convolution layer，其中該layer包含Conv2d(512, 128, kernel\_size=(1,1), stride=(1,1))、BatchNorm2d(128)、ReLU(True)。如下圖有我們的完整架構。

Optimizer：使用Adam，lr=0.0001、weight\_decay=0.00001

Batch：32

NMS\_confidence\_threshold：0.01

NMS\_IOU\_threshold：0.5



1. (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.

同樣利用上述參數，產生出如以下表格。

|  |  |  |  |
| --- | --- | --- | --- |
| image  epoch | val1500/0076.jpg | val1500/0086.jpg | val1500/0907.jpg |
| epoch1  (mAP=0.0001) |  |  |  |
| epoch51  (mAP=0.1278) |  |  |  |
| epoch98  (mAP=0.1838) |  |  |  |

1. (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.

|  |  |  |
| --- | --- | --- |
| Model  Class | Base-Model  (mAP=0.1134) | Improved-Model  (mAP=0.1838) |
| plane  (8723筆) | 0.222952 | **0.431699** |
| baseball-diamond  (515筆) | 0.147727 | **0.156064** |
| bridge  (2114筆) | 0.046946 | **0.161032** |
| ground-track-field  (621筆) | **0.111888** | 0.100313 |
| small-vehicle  (116228筆) | 0.090909 | **0.095783** |
| large-vehicle  (23746筆) | 0.117607 | **0.157662** |
| ship  (34585筆) | **0.090909** | 0.064064 |
| tennis-court  (3279筆) | 0.491936 | **0.639486** |
| basketball-court  (661筆) | 0.068181 | **0.218181** |
| storage-tank  (5199筆) | **0.098144** | 0.046161 |
| soccer-ball-field  (590筆) | 0.089610 | **0.407404** |
| roundabout  (537筆) | 0.045454 | **0.103448** |
| harbor  (7457筆) | 0.078532 | **0.139161** |
| swimming-pool  (1977筆) | 0.114164 | **0.221705** |
| helicopter  (434筆) | 0.0 | 0.0 |
| container-crane  (136筆) | 0.0 | 0.0 |

從以上表格看得出來，Improved Model在大多數class的情況下表現都比Base Model好很多，而如此多數的增長，我們可以推測Improved Model比較不overfit原本的testing data。而其證據是根據[1]論文中所提及，convolution 1\*1的卷積不僅可以用於降低資料維度，同時也能學習維度之間的關係性。換句話作解釋就是，今天我原本的維度是512\*7\*7，而我利用128個1\*1的kernel去降維成128\*7\*7，而在這降維之中，每一個output 7\*7都是我一個kernel跟原本512個維度去做不同權重所累加起來的結果，也就是說假設今天我的512維度中有某個維度特別overfit原本的data，我可以利用這128個1\*1kernel去做不同程度的權重並壘加，如此一來的成果就是能降低該overfit維度的影響力，並間接提升其他維度的權重，讓他們之間達到一個平衡有關連的狀態，所以才說1\*1的convolution kernel可以學到不同維度之間的關係性，並同時藉由不同權重減少overfitting的可能性。而在Improved Model中，剛好就是加入了1\*1的convolution layer，並在其後接上一個BatchNorm及ReLU讓整體資料的影響力更鮮明，並不那麼overfit，所以Improved Model才能在mAP上有較好的表現結果。

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

延續第五小題的表格，我們可以看到helicopter跟container-crane的AP都比其他類別來得低上許多，AP根本就只有0。而其中container-crane的ground truth data只有136筆，相較於每個class筆數平均下來的12925筆低上了許多，這是我認為container-crane訓練不好的原因，因為筆數太少。而helicopter的ground truth data筆數有434筆，雖然也是相對較少，但其他與之相近的baseball-diamond有515筆、basketball-court有661筆、roundabout有537筆、soccer-ball-field有590筆、ground-track-field有621筆，這些class都訓練得比helicopter還要好上許多，我認為其原因是因為helicopter在ground truth data中與地板的顏色太過相近，helicopter幾乎就是灰色的小蚊子，而地板也幾乎都是灰色的水泥地或是滑行跑道，他們之間的顏色太過相近以至於Model難以分析出哪邊才是helicopter。

而其他AP不到0.1的class我認為都是筆數不夠多或是物件太小，例如small-vehicle明明有大量的data，但卻因為物件小、密集度高，無法用7\*7如此大的grid去判斷細小的物件，也就是說可能一個grid裡面就包含了好幾個small-vehicle以至於該grid無法正確預測出small-vehicle的正確位子。而其他能高於0.1AP的class大多都是有足夠訓練data，或是物體本身夠大夠鮮明，足以使用7\*7大小的grid去代表，例如large-vehicle以及tennis-court。

1. Reference

[1] M. Lin, Q. Chen, and S. Yan. Network in network. arXiv:1312.4400, 2013.

1. Collaborators

我整份報告有疑問的地方，都是與我 ImLab 的實驗室同學們一起討論，我們僅限於題目定義與演算法的討論，實作的 code 都是各自寫各自的。以下是同學們的 ID，b04901190、R07922002、R07922024、R07922043、R97922120