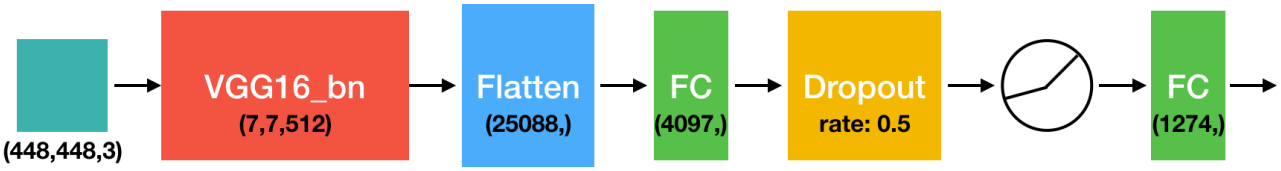
名字： 劉兆鵬 系所：電機所碩一 學號：R07921052

1. Print the network architecture of your Yolo V1-vgg16bn model and describe your training config. (optimizer, batch size… and so on)

Model Architecture:

使用的 baseline model 為使用 vgg_16bn 作為 backbone model，並於其輸出的 feature 攤平成一維的向量，經過 4096 維的 fc 後，在使用 fc 轉換至 yolo label 的維度 (7*7*26)。且於中間 fc 的中間加入 LeakyReLU 以及 dropout。最後在使用輸出 reshape 為 (7,7,26) 來與 ground truth 計算 loss。其架構如下所示。



Optimizer:

我使用的 optimizer 為 SGD 且初始的 learning rate 為 1e-3，並於 30 epoch 後將 learning rate 乘以 factor 0.1，使 learning rate 逐漸降低，讓模型訓練步伐降低。且我於 SGD 加入 momentum 0.9 以及 weight decay 5e-4，避免 model 容易 overfitting 產生。

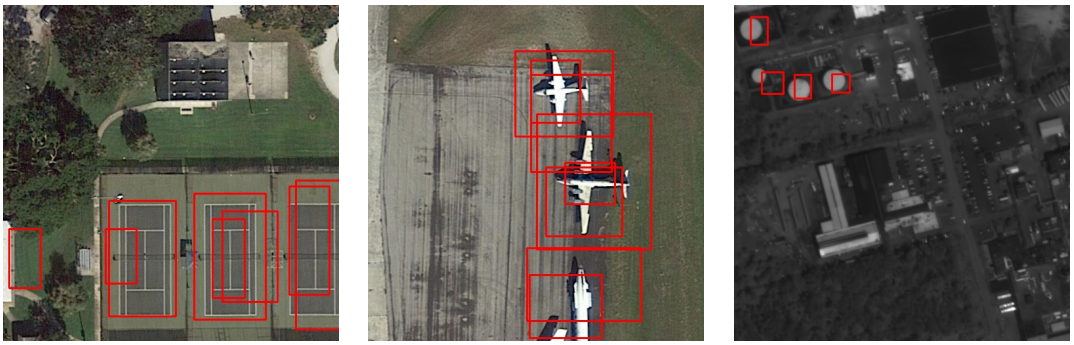
Batch size:

我所使用的 batch size 初始為 16，讓 model 一次看較少的圖片並於一個 epoch 中調整參數多次，使模型可以對每一小捆圖片都進行調整。接著於每 10 個 epoch 將 batch size 增加 4 的大小直到最大 batch size 32，增加 batch size 的用意為希望模型一開始從小 batch 的圖片多訓練幾次，讓 model 可以快速符合訓練資料。而增加 batch size 是因為希望 model 在看完每個圖片後，可以一次看多一點圖片，進行大範圍圖片的 gradient 方向調整參數，使 model 可以根據整體的圖片內容進行調整。

2. Show the predicted box image of “val1500/0076.jpg”, “val1500/0086.jpg”, “val1500/0907.jpg” during the early, middle, and the final stage during the training stage.(ex: results of 1st, 10th…)

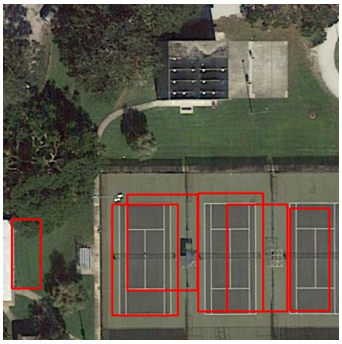
在此我分別使用了第 5 個，第 13 個以及第 80 個 epoch 來做預測繪圖，查看其於所述三張圖片的結果，其比較都使用相同的 threshold 做篩選。如下圖所示。從中可以察覺在前期 model 所輸出的 bounding box 數量較多，因為一開始模型不了解物件的信心。故會於每個 cell 都輸出為物件。後期的 bounding box 能較好的框取物件，且重複的框數量較少。

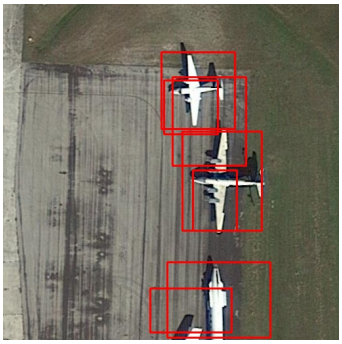
第5個 epoch:



0076.jpg　　0086.jpg　　0907.jpg

第13個 epoch:

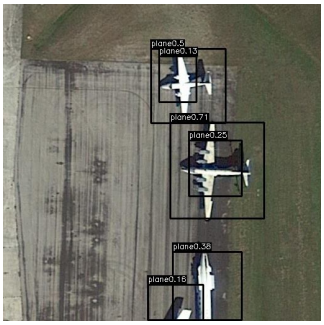

0076.jpg


0086.jpg


0907.jpg

第88個 epoch:


0076.jpg

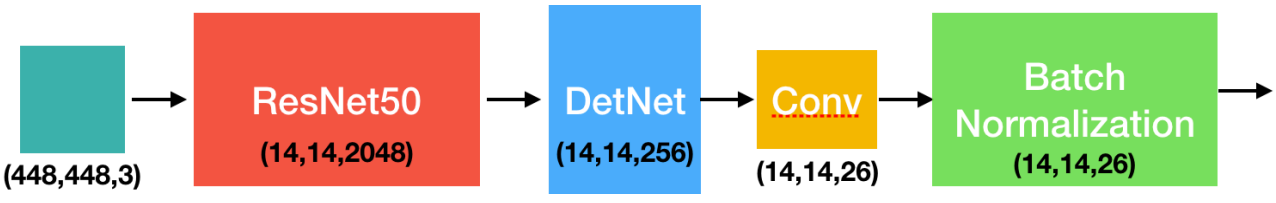

0086.jpg


0907.jpg

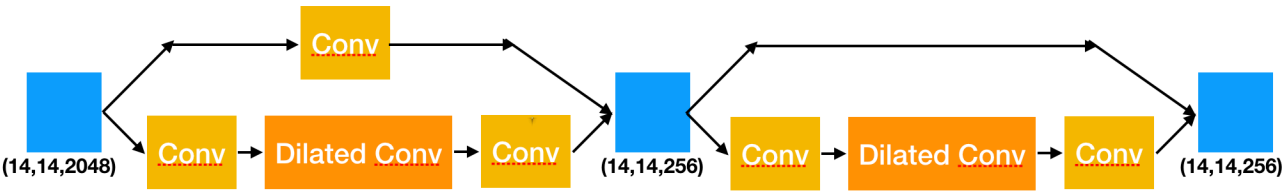
3. Implement an improved model which performs better than your baseline model. Print the network architecture of this model and describe it.

Model Architecture:

使用的 improved model 改為使用 ResNet50 作為我的 backbone，而後將 ResNet50 所輸出的特徵經過 detnet unit，來學習不能輸出所串接起來的特徵。最後原先的 baseline model 為使用 fc 來轉換成 label 的維度，在此我使用 Conv 來降低我的 channel 數量使其與我的 label 維度相同，再接上 batch normalization 作為輸出。其架構如下所示。



而 DetNet 的架構如下所示。



Optimizer:

我於 improved model 的訓練方式皆於 baseline model 相同，我使用的 optimizer 為 SGD 且初始的 learning rate 為 1e-3，並於 30 epoch 後將 learning rate 乘以 factor 0.1，使 learning rate 逐漸降低，讓模型訓練步伐降低。且我於 SGD 加入 momentum 0.9 以及 weight decay 5e-4，避免 model 容易 overfitting 產生。

Batch size:

我於 improved model 的訓練方式皆於 baseline model 相同，我所使用的 batch size 初始為 16，讓 model 一次看較少的圖片並於一個 epoch 中調整參數多次，使模型可以對每一小捆圖片都進行調整。接著於每 10 個 epoch 將 batch size 增加 4 的大小直到最大 batch size 32，增加 batch size 的用意為希望模型一開始從小 batch 的圖片多訓練幾次，讓 model 可以快速符合訓練資料。而增加 batch size 是因為希望 model 在看完每個圖片後，可以一次看多一點圖片，進行大範圍圖片的 gradient 方向調整參數，使 model 可以根據整體的圖片內容進行調整。

4. Show the predicted box 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.(ex: results of 1st, 10th…)

在此我分別使用了第 1 個，第 20 個以及第 82 個 epoch 來做預測繪圖，查看其於所述三張圖片的結果，其比較都使用相同的 threshold 做篩選。如下圖所示。從中可以察覺在前期 model 所輸出的 bounding box 數量較多，位置也較不在物件中心，而越後期的模型所偵測的物件框則能較貼近物件邊緣以及對應中心的位置。

第1個 epoch

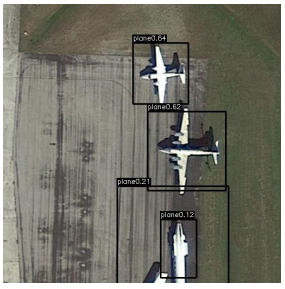
0076.jpg

0086.jpg

0907.jpg

第20個epoch

0076.jpg

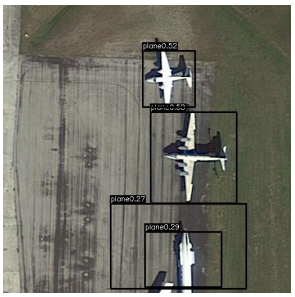


0086.jpg

0907.jpg

第82個epoch


0076.jpg


0086.jpg


0907.jpg

5. 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 evidence to support your reasoning.

Baseline model score:

mAP: 0.1797

Plane	Baseball diamond	Bridge	Ground track field	Small vihicle	Large vihicle	Ship	Tennis court
0.5459	0.0	0.2053	0.0521	0.0759	0.2209	0.1123	0.7061
Basket ball court	Storage tank	Soccer ball field	Roundabout	Harbor	Swimming pool	Helicopter	Container crane
0.0	0.1527	0.2821	0.0	0.3498	0.1723	0.0	0.0

Improved model score:

mAP: 0.5719

Plane	Baseball diamond	Bridge	Ground track field	Small vihicle	Large vihicle	Ship	Tennis court
0.8699	0.7499	0.5179	0.5151	0.5871	0.6368	0.6790	0.8912
Basket ball court	Storage tank	Soccer ball field	Roundabout	Harbor	Swimming pool	Helicopter	Container crane
0.6227	0.7735	0.4369	0.6538	0.6912	0.4974	0.0279	0.0

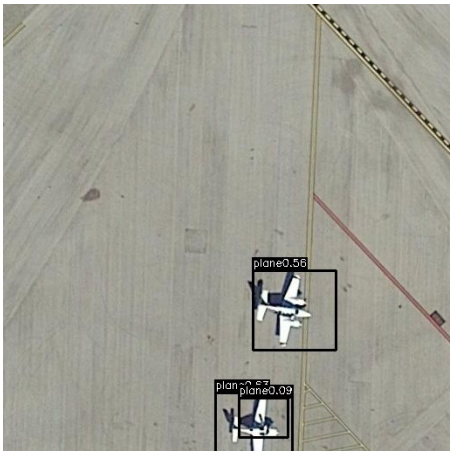
根據上表所列，可以得知使用 improved model 的結果是比 baseline Model 要來得好，而其中我認為較好的原因為 improved model 使用了 ResNet 以及 detnet，這兩者都是

使用了 skip connection 的方式，使得 model low level 的資訊能夠向後傳遞，由於 CNN 的會隨著越深層使得 feature size 越小，也會導致越小的物件越不容易被偵測出來，故使用 skip connection 將 low level 資訊加入偵測層，可以使得模型能夠偵測出較小的物件。且在使用 improved model 時，我將輸出的 grid cell 數量調整為 14，使一張圖片能夠用擁有的物件 label 數量變多了，讓靠近的物件也能夠分到不同的 grid 中。實驗如下圖所示。

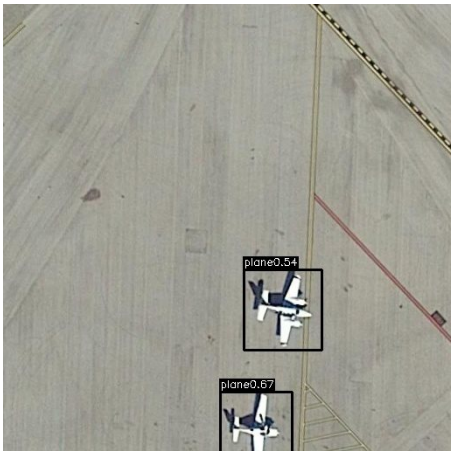
Baseline Model

Improved Model

另外，Improved model 將最後的分類層中把 FC layer 改成 1x1 CNN layer 我認為也是提升偵測物件位置的能力，因為若是使用 fc 是將圖片的 feature 一起計算，然而想要偵測出一個物件的位置應只需要其周圍的資訊就能夠進行預測，故使得模型更能夠精準抓取物件的位置資訊。如下圖所示，可以發現 improved model 框的位置更精準。



Baseline Model



Improved Model

6. Bonus. Which classes prediction perform worse than others? Why? You should describe it.

根據上題所顯示的 AP score 來看結果的話，helicopter 以及 container crane 是較其他類別較難辨識出來的。我認為無法準確的辨識出兩者物件位置的原因可能為其資料數量較少，所以無法使模型有效的學習到物件的長相。故我統計了 training data 中各類別物件的數量，如下表所示。可以發現數量較少的物件其對應的準確度也相對低了許多。從 baseline model 分數中可以清楚得知 data imablance 的問題造成的 AP 無法提升的問題。故若要提

升整體的物件偵測的準確度應該要將提升較少數量物件的圖片資料，使其模型能夠學習到少數類別的物件資訊。

Plane	Baseball diamond	Bridge	Ground track field	Small vihicle	Large vihicle	Ship	Tennis court
8723	515	2114	621	116228	23746	34585	3279
Basket ball court	Storage tank	Soccer ball field	Roundabout	Harbor	Swimming pool	Helicopter	Container crane
661	5199	590	537	7457	1977	434	136

故我將少數類別的圖片重複多次使用，使得訓練時 model 能夠多看幾次少數類別的圖片，而由於我增加了許多種 data augmentation，故能夠避免一些圖片重複訓練的問題產生，而我講少數類別圖片加進去重新訓練後，能夠有效地改善一些 data imabalance 的問題，其 baseline model 結果如下圖所示。

mAP: 0.2399

Plane	Baseball diamond	Bridge	Ground track field	Small vihicle	Large vihicle	Ship	Tennis court
0.5561	0.0	0.2413	0.2764	0.0785	0.2493	0.1143	0.7283
Basket ball court	Storage tank	Soccer ball field	Roundabout	Harbor	Swimming pool	Helicopter	Container crane
0.3996	0.1538	0.3811	0.0771	0.3386	0.2057	0.0387	0.0

7. 參考資料

Github: https://github.com/xiongzihua/pytorch-YOLO-v1
Collaborator: r06942141