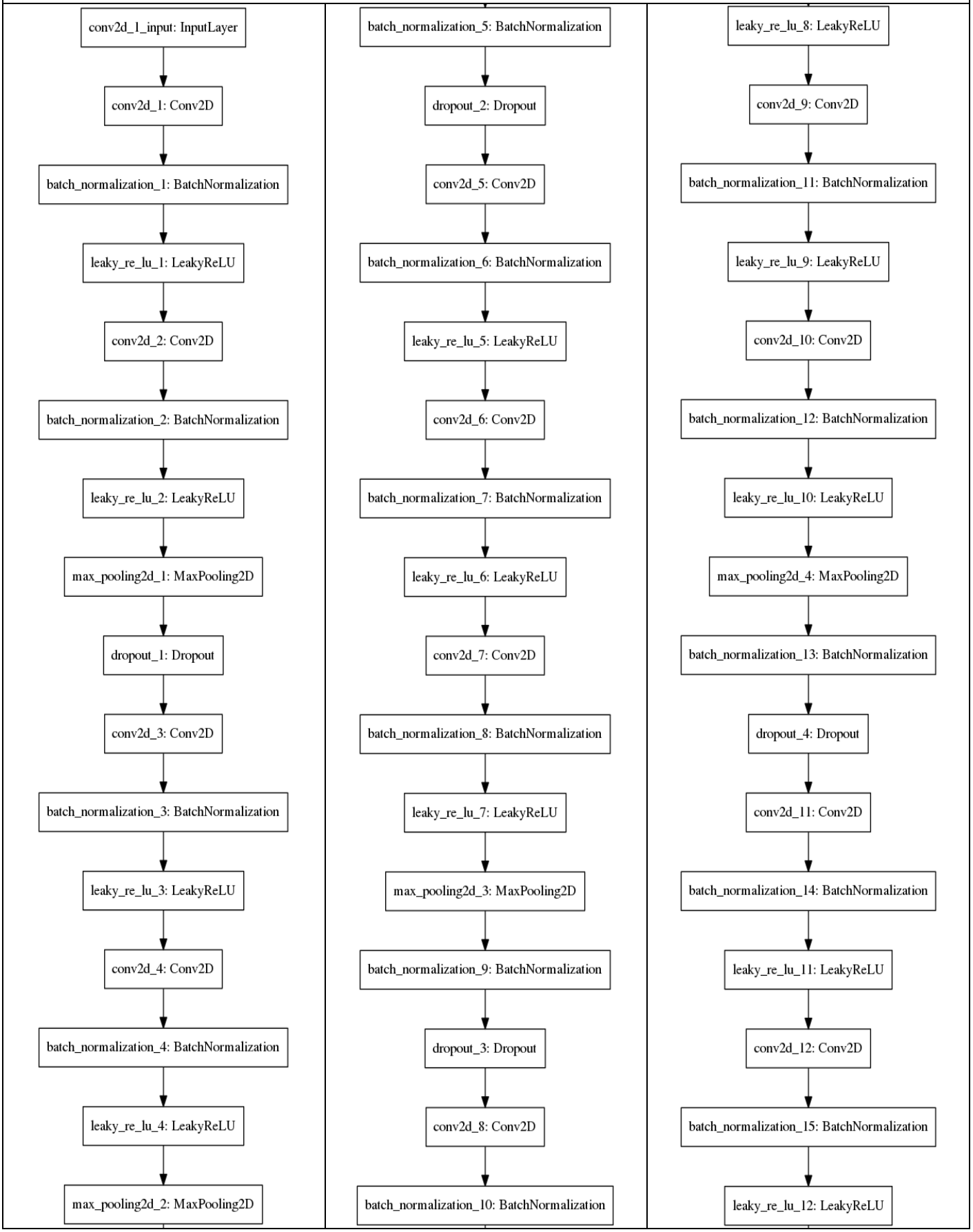
學號：R06922002 系級： 資工碩一 姓名：姚嘉昇

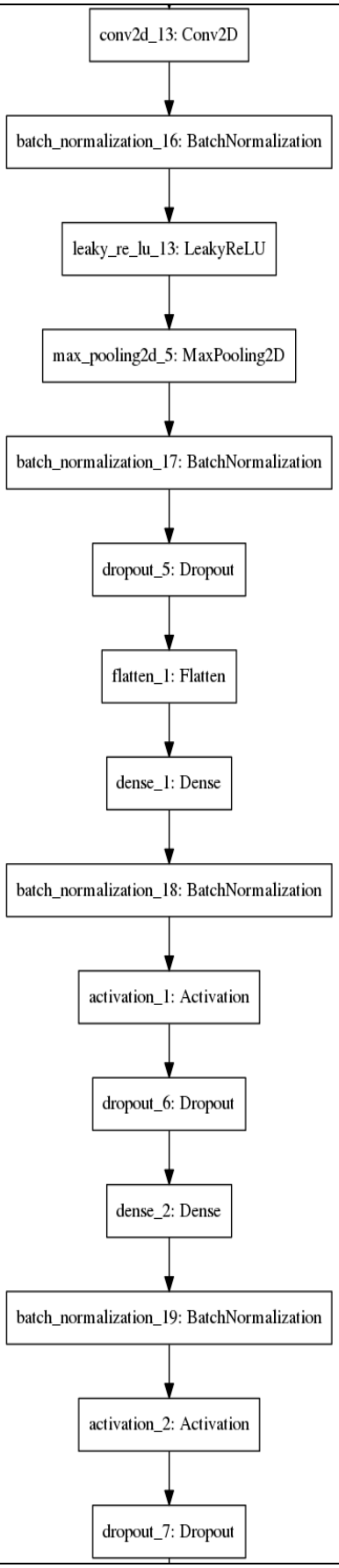
1. (1%) 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？
   (Collaborators:)

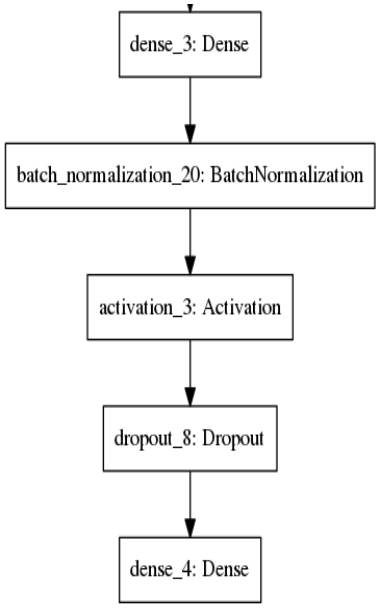
答：

模型架構(一)



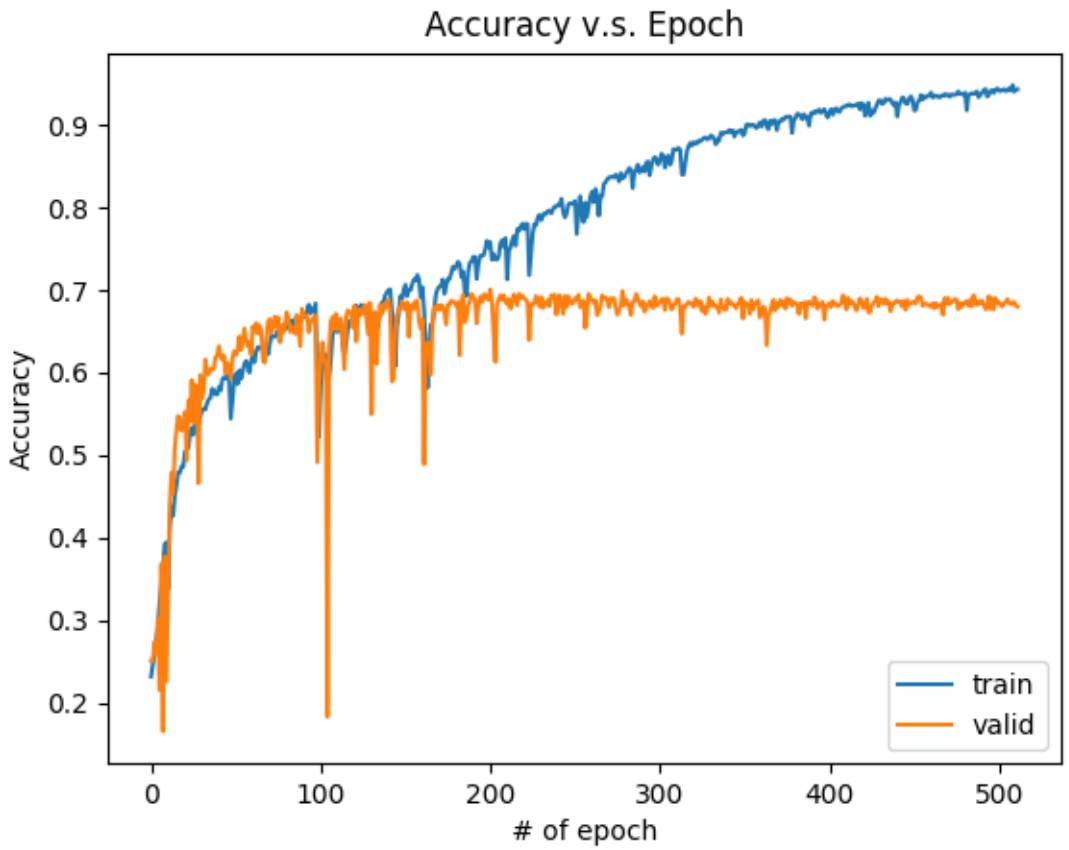
模型架構(二)





Total params: 37,759,399
Trainable params: 37,729,751
Non-trainable params: 29,648

訓練過程



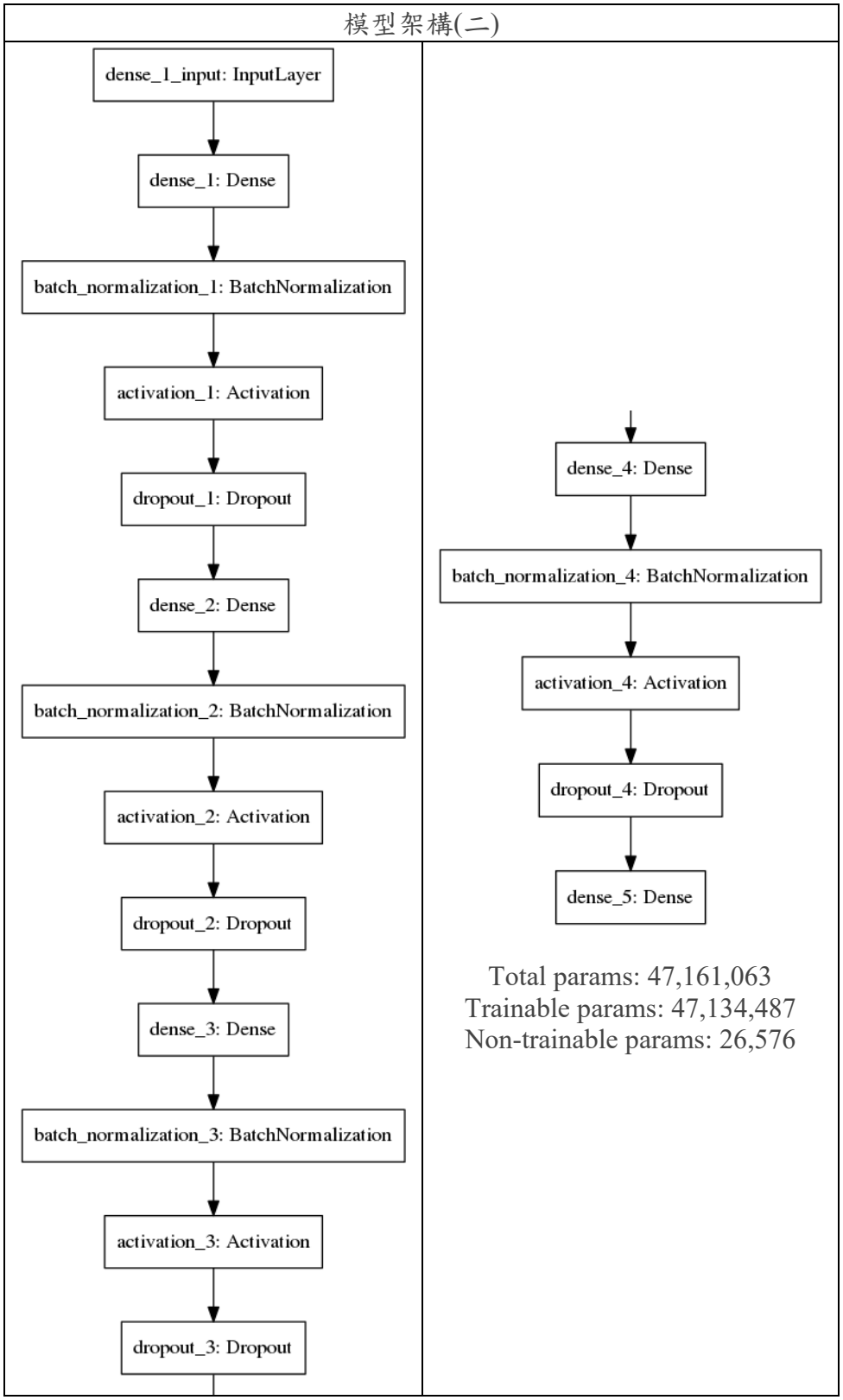
	Training set		Testing set (Kaggle)		
	Training	Validation	Public	Private	Average
Accuracy	0.94839	0.70060	0.70075	0.68821	0.69448
Epoch	Best at 508	Best at 200	Training epoch = 256		

我的 CNN model 主要架構與 VGG-16 相似，而增加了 Batch Normalization、替換部分 Activation 為 LeakyReLU，並使用 Keras 提供的 ImageDataGenerator 來製造更多 data 給 model 訓練。

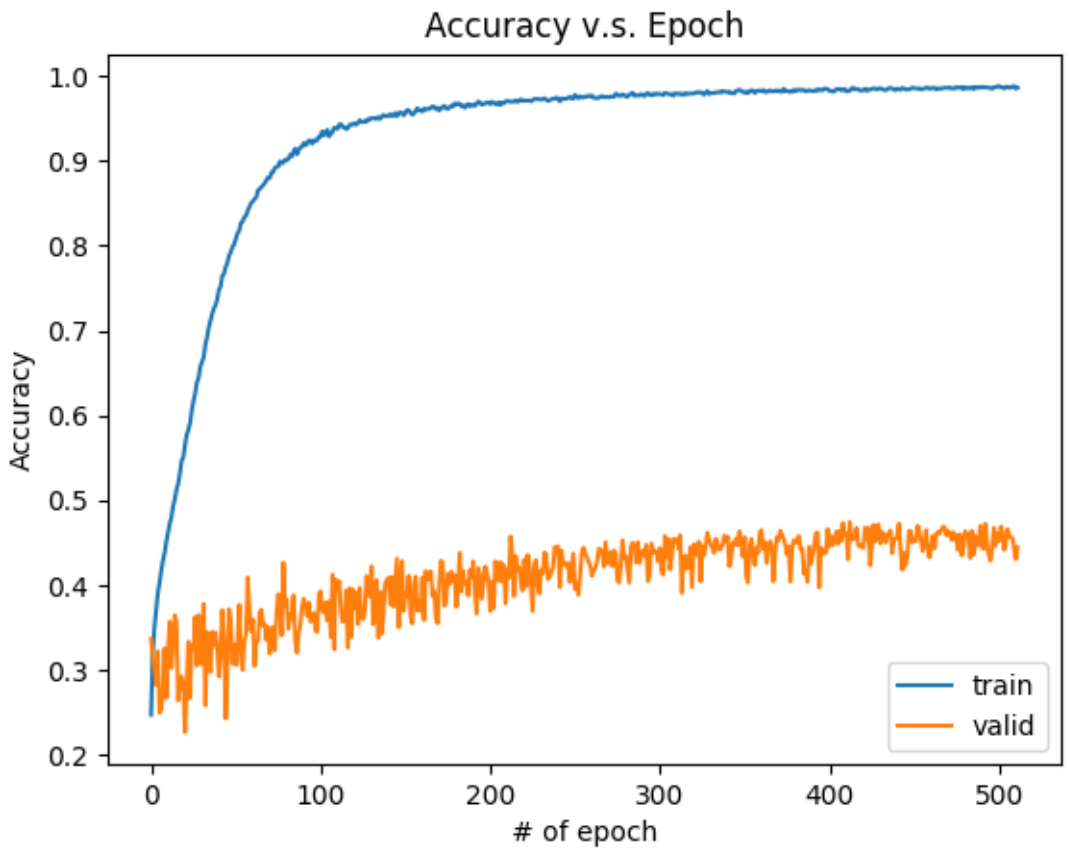
因為其訓練過程中，Validation 的分數在第 200 個 epoch 有最高的 accuracy，而其在第 200 個 epoch 之後 accuracy 已趨於穩定，所以我選擇 epoch 為 256 作為 early stopping，再把所有的 data 重新訓練一次後，才上傳 Kaggle。

2. (1%) 承上題，請用與上述 CNN 接近的參數量，實做簡單的 DNN model。其模型架構、訓練過程和準確率為何？試與上題結果做比較，並說明你觀察到了什麼？
(Collaborators:)

答：



訓練過程



	Training set		Testing set (Kaggle)		
	Training	Validation	Public	Private	Average
Accuracy	0.98820	0.47475	0.42630	0.45332	0.43981
Epoch	Best at 500	Best at 412	Training epoch = 512		

我拿的參數量相近（甚至更多）的 DNN 來做訓練，他在 training set 上的 accuracy 很快就衝到了八九十，甚至在 300 的 epoch 後已經接近 100%的 accuracy，很顯然 DNN 很能在 training set 很 fit，但是在 validation set 上就沒這麼好了，一開始 accuracy 只有 0.3 左右而且震盪嚴重，就算到最後 512 個 epoch 後，還是只有 0.47 的 accuracy。

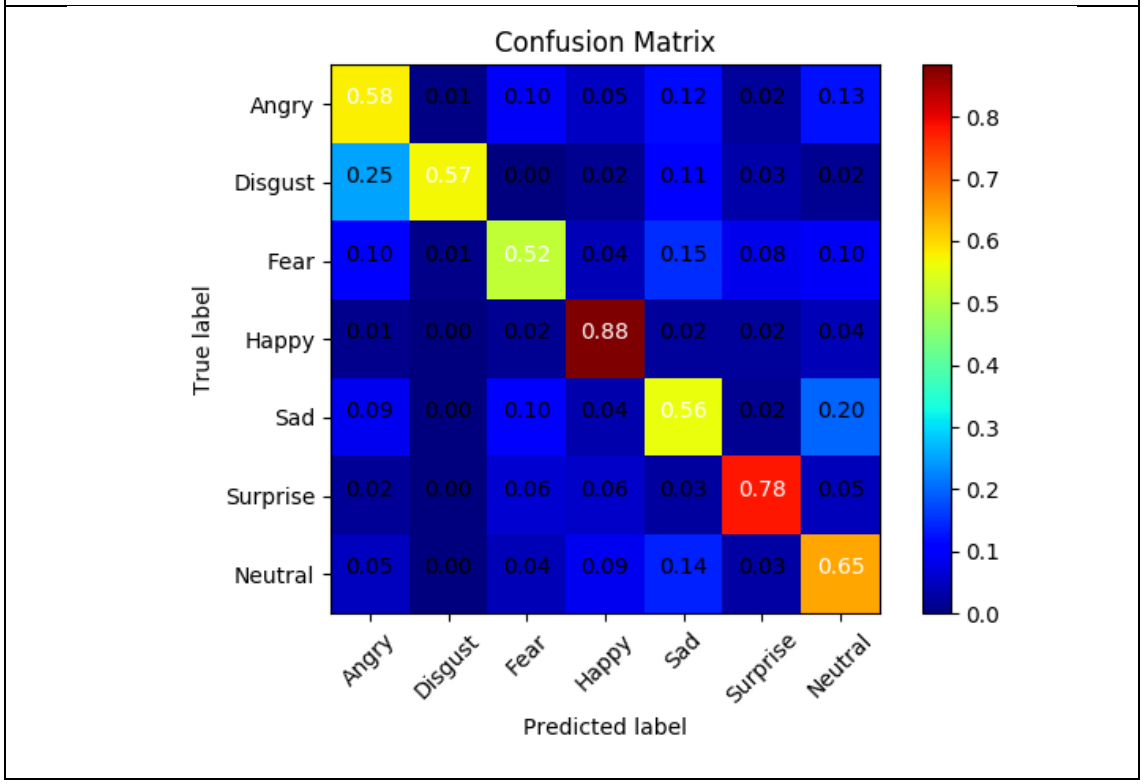
所以，在我們有限的資源下，一樣的參數量，使用 CNN 除了能夠在 training set 上 fit，也能快速地在 validation set 上 fit，所以在數位影像處理這塊，CNN 還是有所優勢。不然如果用 DNN 訓練了一大堆參數，卻沒有 CNN 的好，反而是浪費運算資源。

3. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析]
(Collaborators:)

答：

原本是 Disgust 的圖片有 0.25 被 predicted 成 Angry；原本是 Sad 的圖片有 0.20 被 predicted 成 Neutral。這兩對是我的 model 最容易混淆的，可能可以針對這幾種 label 的資料再加上更多 data augment 加以訓練，而還有其他高於 0.10 低於 0.20 的 pair 我就不加以討論了。

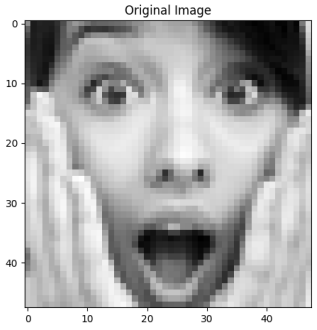
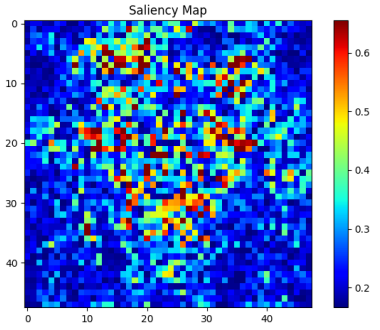
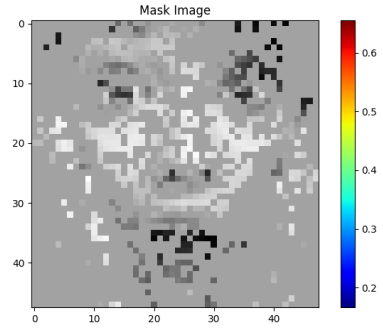
Confusion Matrix



4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？
(Collaborators:)

答：

從 Saliency Map 中來看，紅色的點集中在兩個眼睛、兩個眼袋、鼻子還有嘴巴上圍。而在 Mask Image 中(我使用 0.33 作為 threshold)，可以看到除了剛剛 Saliency Map 紅色的點有保留下來，雙手的部分也有被保留一些下來，可見這個 model 可能有藉由臉部以外的東西(例如：雙手)來判讀表情。

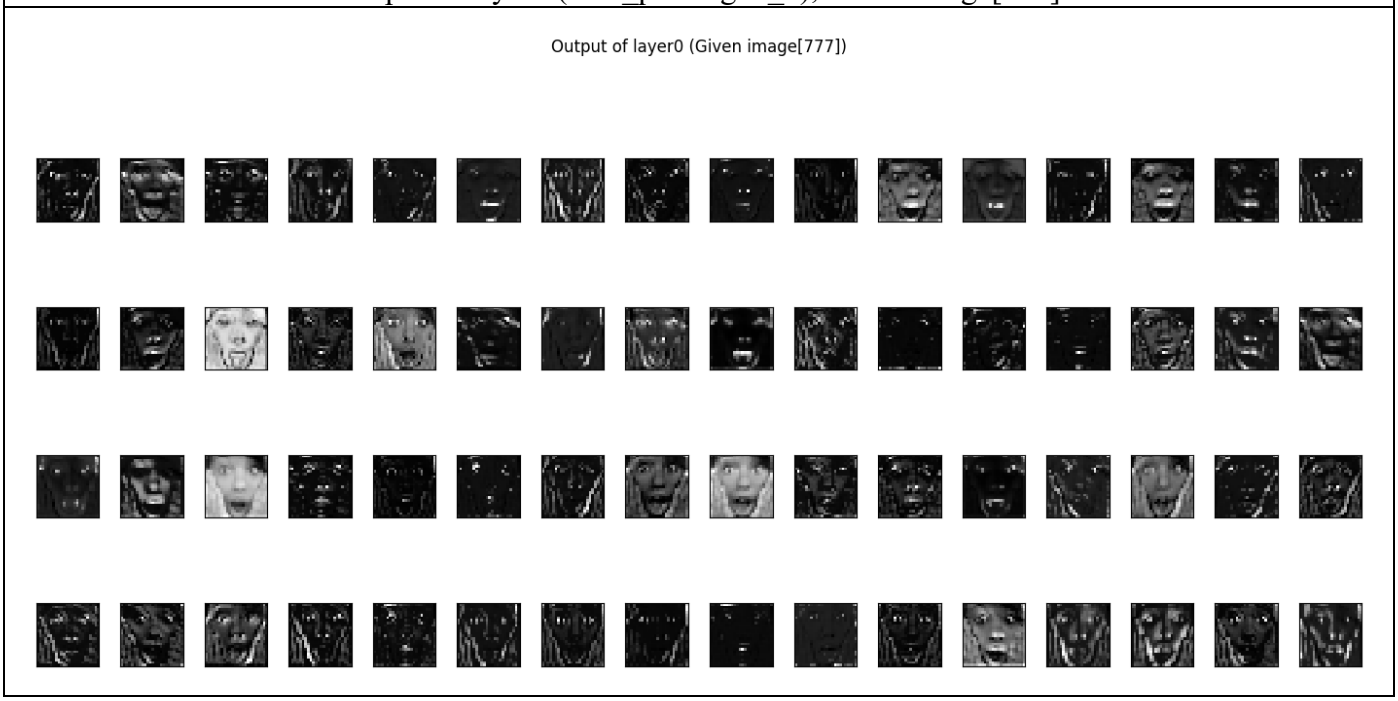
Original Image	Saliency Map	Mask Image
		

5. (1%) 承(1)(2)，利用上課所提到的 gradient ascent 方法，觀察特定層的 filter 最容易被哪種圖片 activate。
   (Collaborators:)

答：

我拿 VGG-16 架構的第一個 Convolution block 的 Max pooling 那層(max_pooling2d_1)來觀看，可以看到拿第 777 張圖片丟進去後， 這層所輸出的資料如下圖所示，可以看出有些層保留了人臉的輪廓，有些層留下了雙眼，有些層只剩下嘴巴，藉由這方法，可以讓我清楚知道每個 filter 分別做了什麼事情。

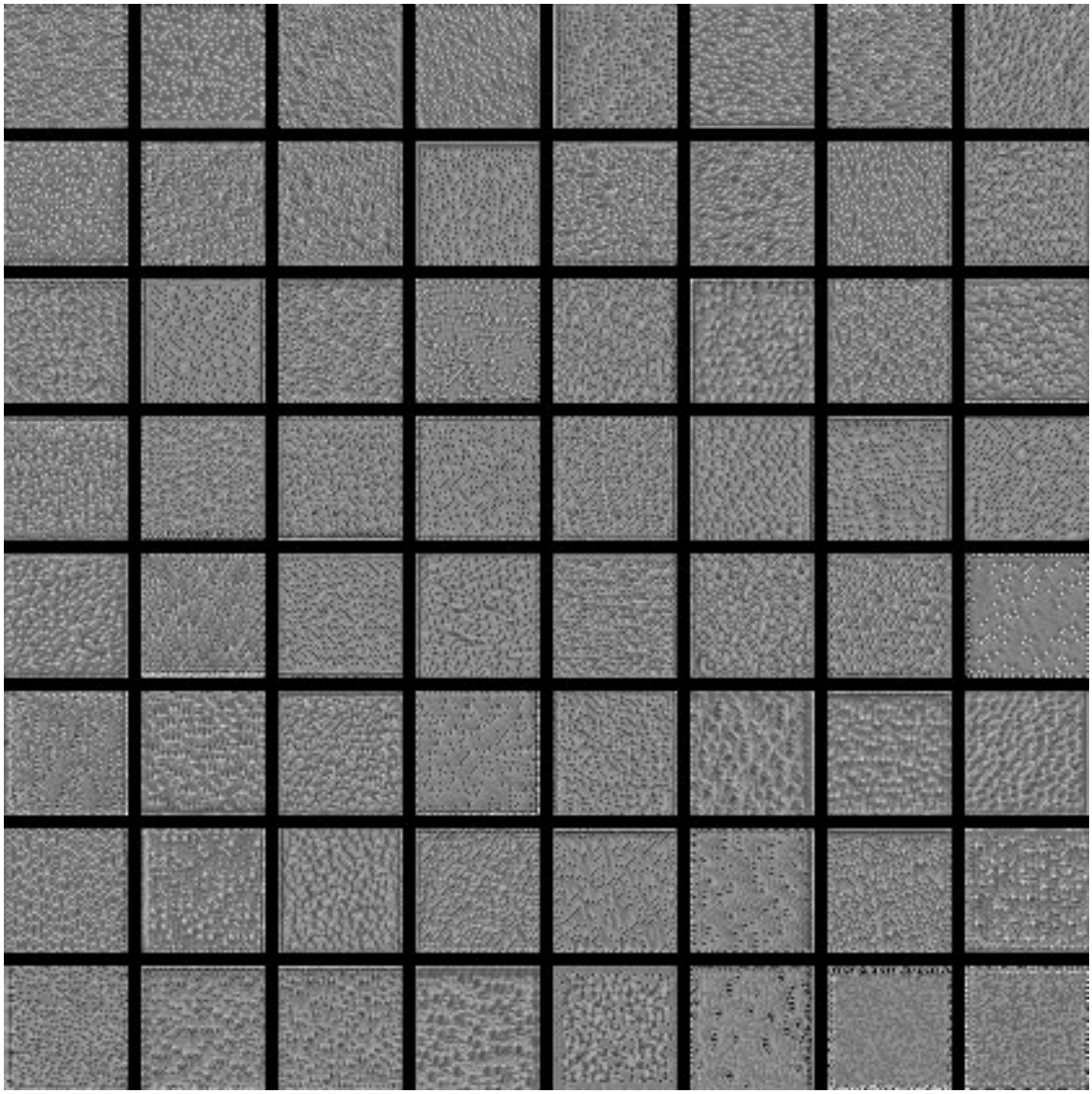
Output of layer 0(max_pooling2d_1), Given image[777]



下面的圖片是用 gradient ascent 觀察 max_pooling2d_1 這層，哪種圖片最能 activate 這層的 64 個 filter，下一頁的圖片則是用 VGG-16 架構中第五個 Convolution block 中的第一個 Conv2D 那層 (conv2d_11)來觀察。

可以發現比較淺層的 filter (max_pooling2d_1)比較像是在抓臉部的紋路輪廓或者是 focus 在部分器官，畫出來的紋路都是基本幾何圖形並且在整張圖中有高重複性；而比較深層的 filter (conv2d_11)看起來就是非常複雜了，畫出來的圖片非常扭曲，各種形狀都有，可能針對各種表情都有一種特定的形狀來辨別。

Layer name = max_pooling2d_1, gradient ascent



Layer name = conv2d_11, gradient ascent