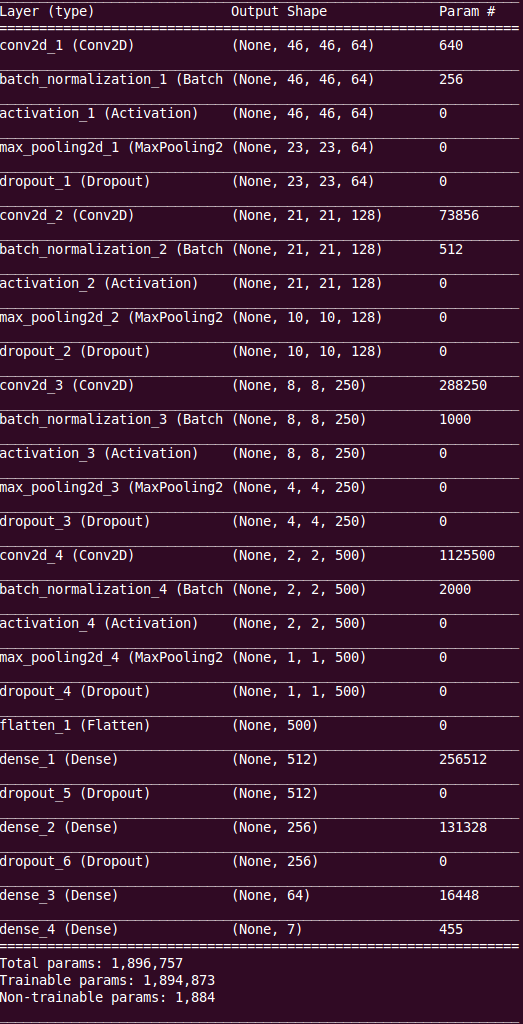
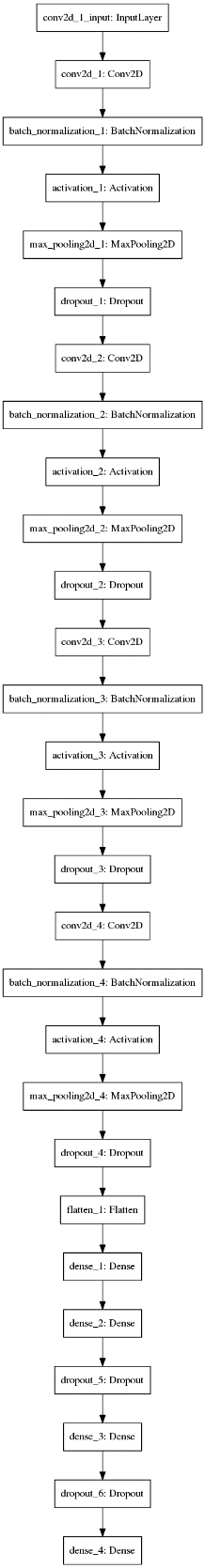
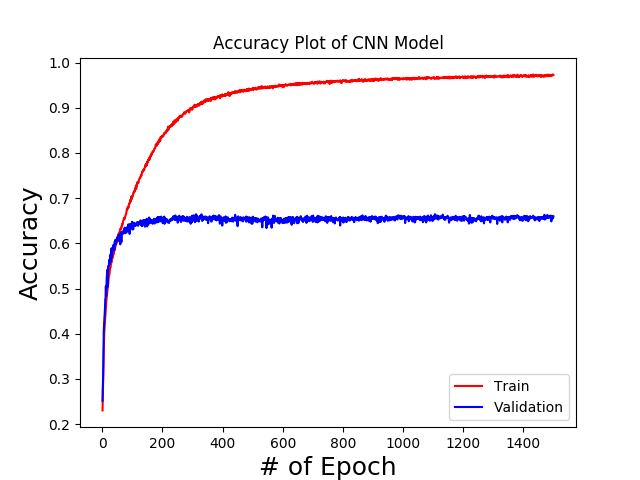
學號：R05921069 系級： 電機碩一 姓名：黃武昱

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？

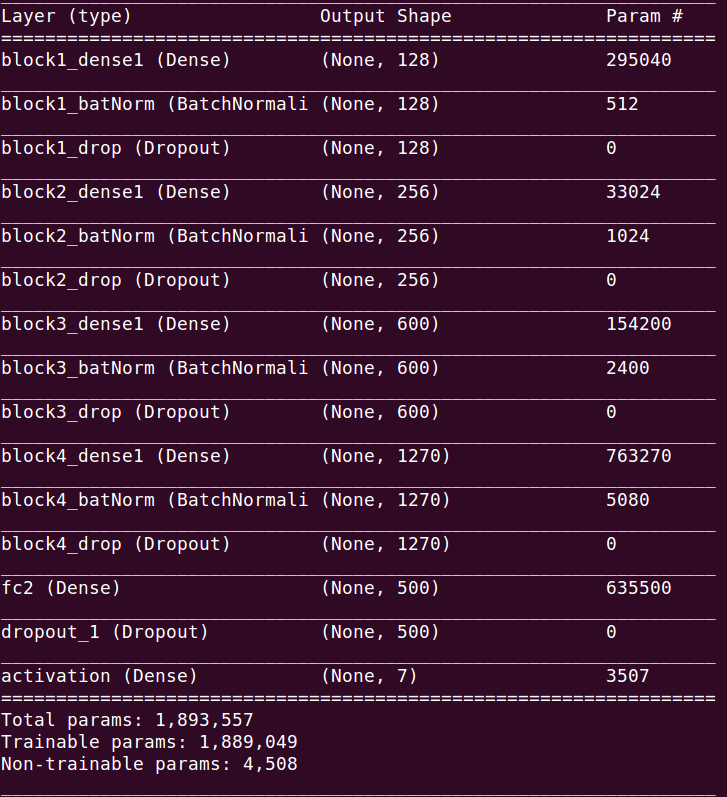
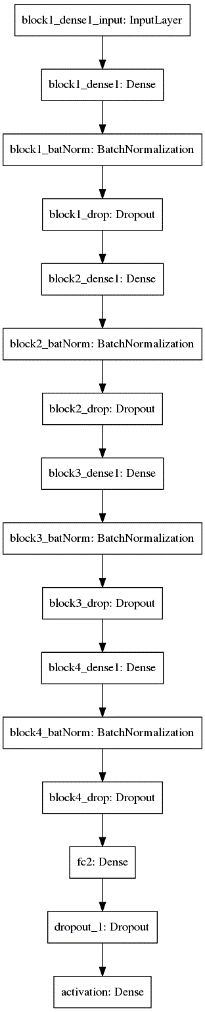
答：在做CNN的部分，我是用了4層的convolution2D layer然後都有分別加上Batchnormalization，每一層我選用relu當作我的Activation function，之後也有用Maxpooling去縮減圖片的大小和Dropout去除不好的feature，最後在使用2層的Dense也是選用relu跟最後一層Dense選用sofmax去得到篩選的結果。

這是CNN根據train data跟validation data得到的accuracy圖:

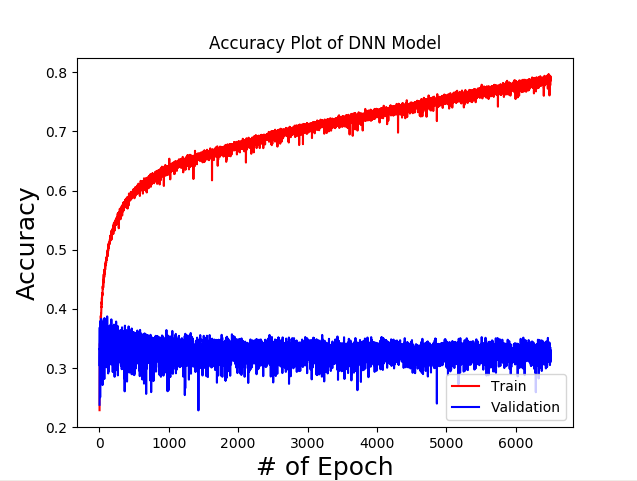


從圖表可得知，Validation在大約150次epoch後就幾乎維持在64%~66%之間震盪，然後Train的還持續上升中，這也表示發生了overfit，這部分雖然已經加入Batchnormalize去防止了，所以這部分還需要繼續研究一下。

1. (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？

答：

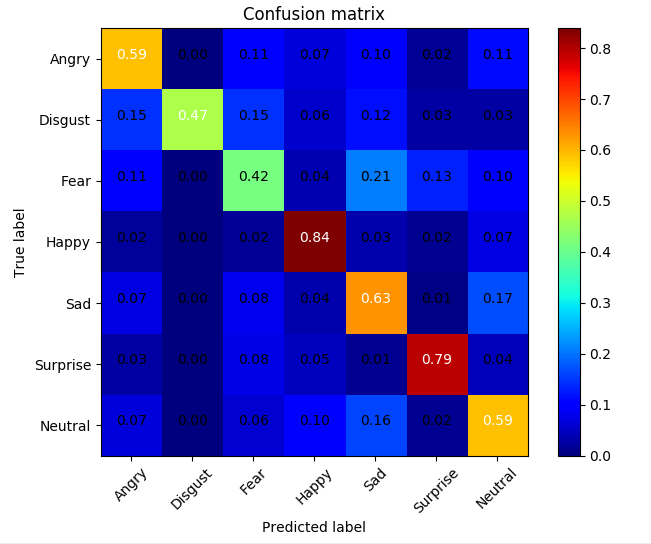
在DNN的部分，我是使用了5層的Dense的和一層輸出Dense層，架構跟CNN是很類似的，然後，參數也是很接近，在與CNN最大不同之處在於，CNN需要訓練的時間，遠大於DNN所需要的時間，畢竟CNN是需要針對一張圖片做很多次mark分成很多filter，這方面的處理量是遠大於Dense的處理量，雖然DNN的速度很快，但精準度的部分，還是CNN遠大於DNN。



從上圖可以發現，Validation的accuracy一直持續在30%~40%之間震盪，

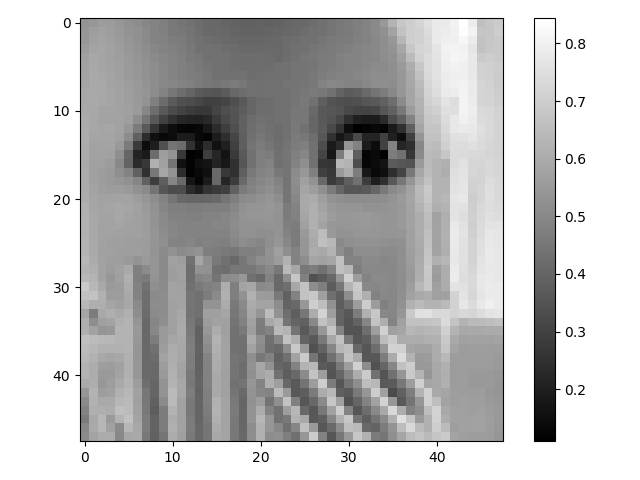
雖然Train也有在上升，但在半途中，震盪幅度也大於CNN。

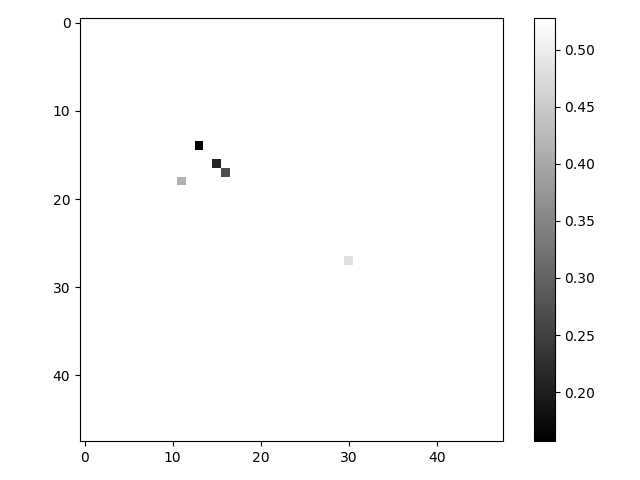
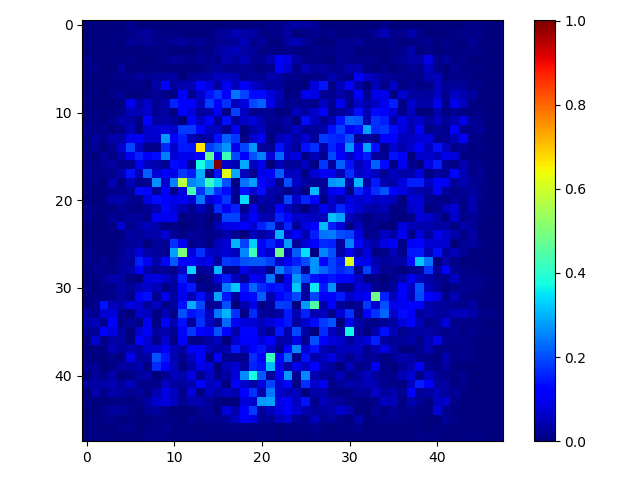
1. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析]

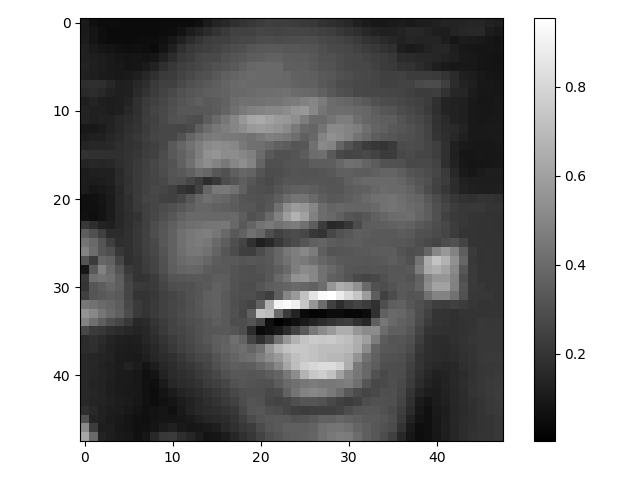
答：

這是用20%的Validation來測試訓練好的model，並且繪製出來的分類圖，由上圖可得知，Happy是最好被辨識出來的，然後Fear反而是最難被辨識出來了，從圖片分析可以推測，Happy的部分較誇大，所以可以得到較多的feature，然後Fear的部分，幅度較小，得到的feature較小，故在分辨時，較少的參考數據，會被跟Sad混淆而出錯。

1. (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？

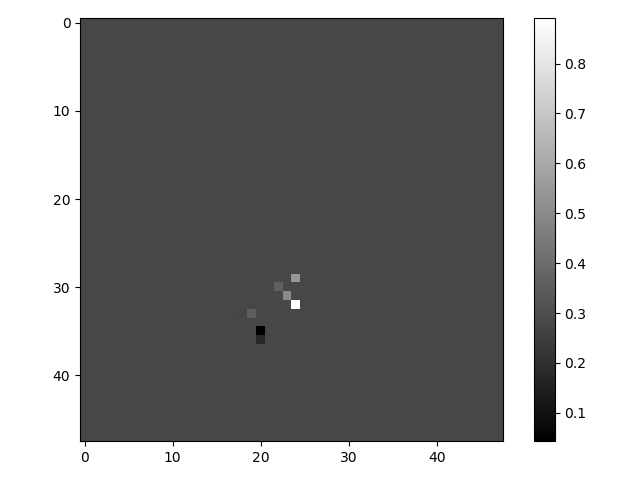
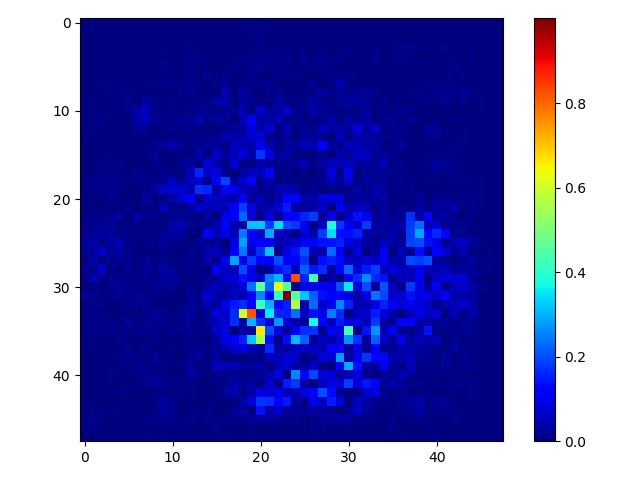
答：首先這是原始驚訝的圖片，雖然有手擋到臉部表情，我們卻仍然能直觀的感覺它是驚訝，但是，電腦卻無法直觀的分辨，我們人類之所以能直觀的辨識，是因為長時間看過類似的表情，才能推測出來，但對於沒有預訓練過大量資料的model要辨別這個是極度的困難。

由這兩張圖可得知，電腦是直接focus在眼睛的的部分，是因為眼睛部分相對於其他部分是比較容易辨別的。

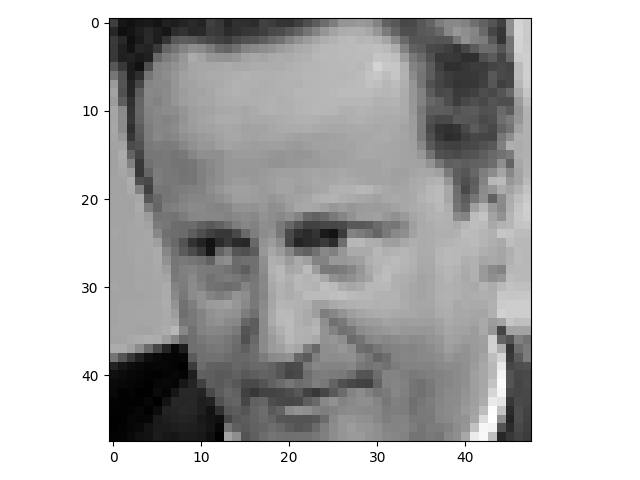
再來我們換另一張表情較容易辨識的圖案來比對一下。

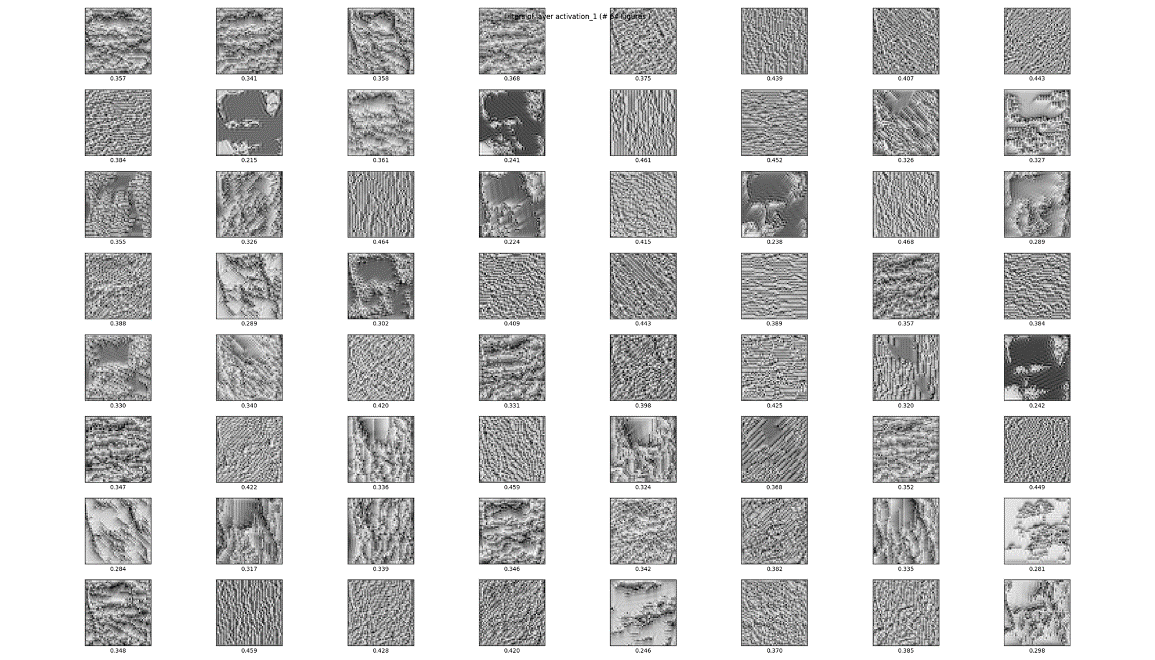
從這張圖，可以得知是笑臉表情的圖案，相對於上圖表情來說，這張圖有很明顯的特徵。

有較多的特徵，那麼電腦在分辨時，就有較多的資料能辨識了。



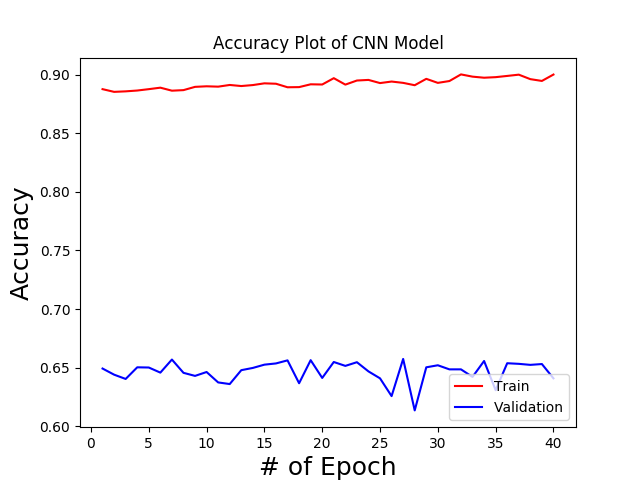
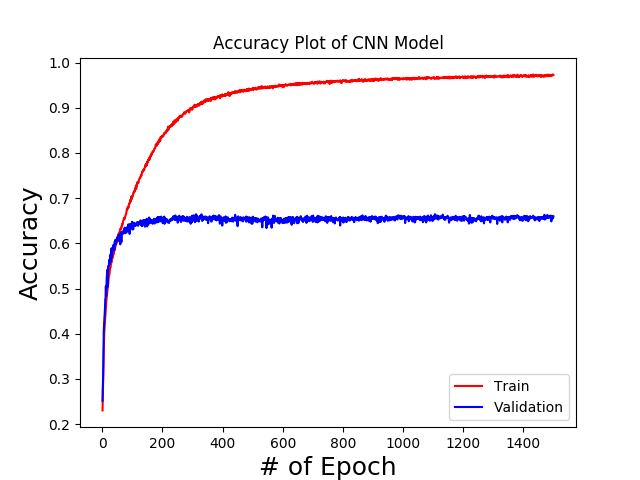
(1%) 承(1)(2)，利用上課所提到的 gradient ascent 方法，觀察特定層的filter最容易被哪種圖片 activate。

答：這是原始圖片

這是選取第一層convolution layer的Activation relu layer後，並且經過gradient ascent出來得圖片，從這裡面可以發現，有機張圖是比較明顯看得出特徵(loss value較低)，有機張圖則是還全看不出來是什麼(loss value較高)，對於我們訓練model而言，只以保留那些比較能看得出臉型(loss value低的)，其餘的都是屬於可以去除的部分。

[Bonus] (1%) 從 training data 中移除部份 label，實做 semi-supervised learning

我把test data丟入訓練好的model，先使用predict\_classes去預測這些data的數值，再用predict\_proba去預測這個data在0~6之間所有可能的機率，並使這兩個資料相乘，看是在哪個數值的機率最高(EX:1的機率是80%)，在把這些data跟從data得到的label加入倒train data中，增加數據量，使精準度上升。

但

右圖為原本的CNN，左圖為加入semi-supervised後的結果，比較之下，CNN的model比較穩定，但理論上有加入semi-supervised應該會比較好，也許我設置的加入ublabel部份的參數過大，需要再調整。