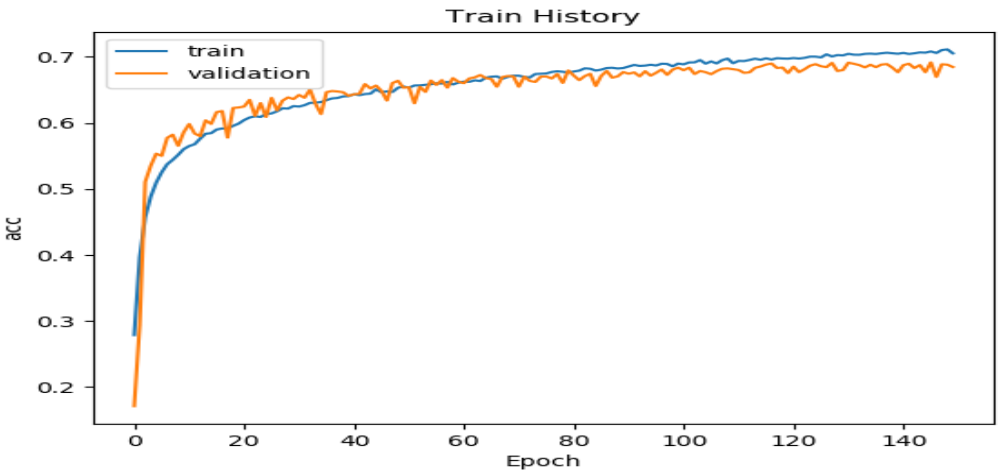
學號：B04901056 系級： 電機三 姓名：張承洋

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:)
＊ training data 左右翻轉(鏡像) ，使用 keras 和下方的模型做大約 150 個 epoch，batch_size=128

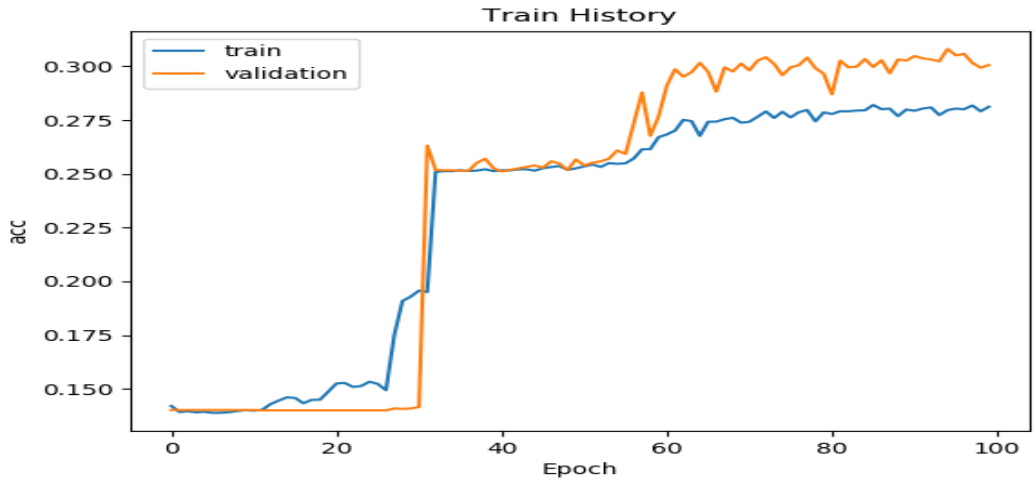


Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 46, 46, 32)	320
batch_normalization_1 (Batch	(None, 46, 46, 32)	128
dropout_1 (Dropout)	(None, 46, 46, 32)	0
conv2d_2 (Conv2D)	(None, 44, 44, 32)	9248
average_pooling2d_1 (Average	(None, 22, 22, 32)	0
dropout_2 (Dropout)	(None, 22, 22, 32)	0
conv2d_3 (Conv2D)	(None, 20, 20, 64)	18496
batch_normalization_2 (Batch	(None, 20, 20, 64)	256
dropout_3 (Dropout)	(None, 20, 20, 64)	0
conv2d_4 (Conv2D)	(None, 18, 18, 64)	36928
zero_padding2d_1 (ZeroPaddin	(None, 20, 20, 64)	0
average_pooling2d_2 (Average	(None, 10, 10, 64)	0
dropout_4 (Dropout)	(None, 10, 10, 64)	0
conv2d_5 (Conv2D)	(None, 8, 8, 128)	73856
batch_normalization_3 (Batch	(None, 8, 8, 128)	512
dropout_5 (Dropout)	(None, 8, 8, 128)	0
conv2d_6 (Conv2D)	(None, 6, 6, 128)	147584
zero_padding2d_2 (ZeroPaddin	(None, 8, 8, 128)	0
max_pooling2d_1 (MaxPooling2	(None, 4, 4, 128)	0
dropout_6 (Dropout)	(None, 4, 4, 128)	0
flatten_1 (Flatten)	(None, 2048)	0
dropout_7 (Dropout)	(None, 2048)	0
dense_1 (Dense)	(None, 7)	14343
activation_1 (Activation)	(None, 7)	0

Total params: 301,671

準確率 : 66.7%

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:)
＊ training data 左右翻轉(鏡像) ，使用 keras 和下方的模型做大約 100 個 epoch，batch_size=128



Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 128)	295040
dropout_1 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 7)	903
activation_1 (Activation)	(None, 7)	0

Total params: 295,943

比較 ：

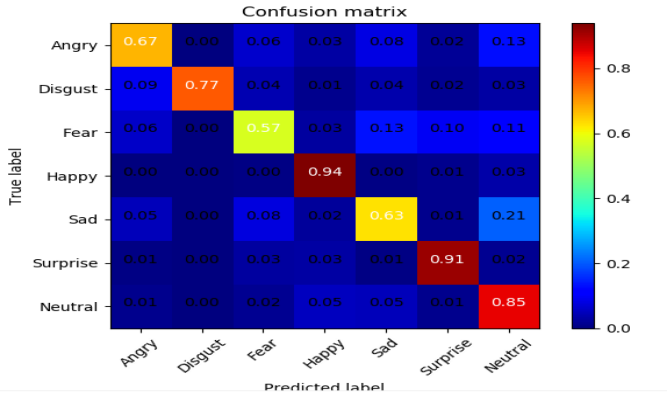
DNN 的準確度比 CNN 低，原因是 DNN 把 input data 當作一個 N*1 的矩陣來作運算，CNN 則是把 input data 當作一張圖片來作運算，藉由特徵來作 classification。

DNN model 就算沒有加 dropout，準確度也不會上升到非常誇張，反之，CNN 如果不加 dropout，準確度就會飆高到一個沒有參考價值的程度。

DNN 的 model 的結構比 CNN 單純，一個 epoch 所花的時間比 CNN 少非常多。

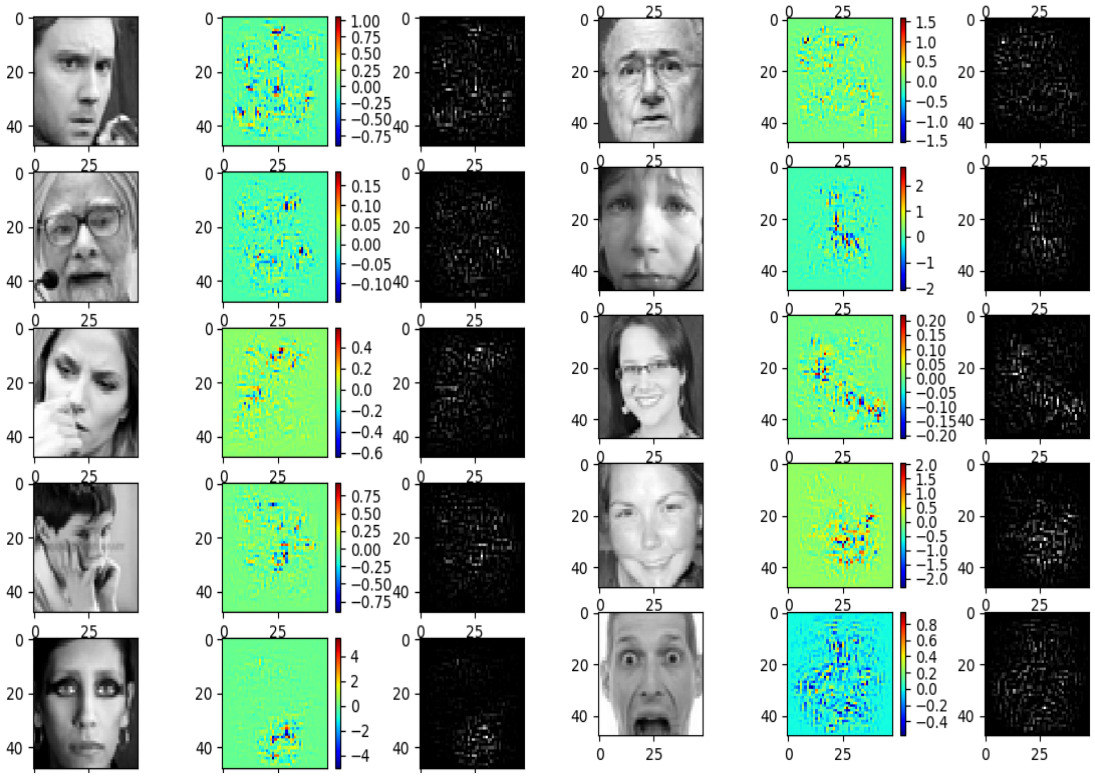
3. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析] (Collaborators:)

答：



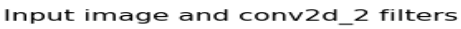
由上表可知以我的 CNN model 來說，Fear 類圖片預測正確率最低(約 57%) ，對 Happy 的辨識度最高(達到 91%)；Sad 類圖片有超過 20%的 input data 被判為 Neutral，而 Fear 類圖片有大約 26%的資料被誤判為 Neutral & Sad ，總結來說，Sad、Neutral、Fear 三類圖片是我的 model 在預測時的主要誤差(混淆)來源。

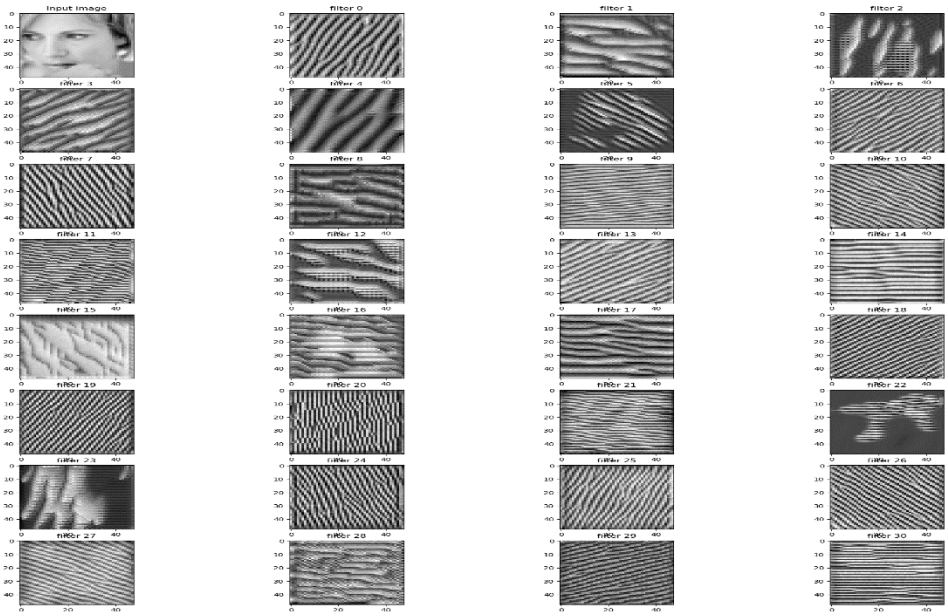
4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？
   (Collaborators:)



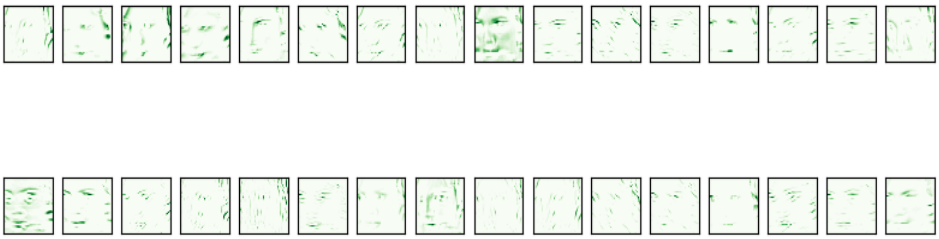
由上圖可以看出，CNN 主要 focus 在眼睛、鼻子、嘴角的輪廓。

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：

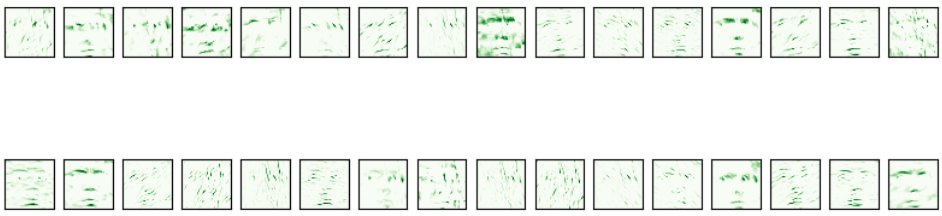




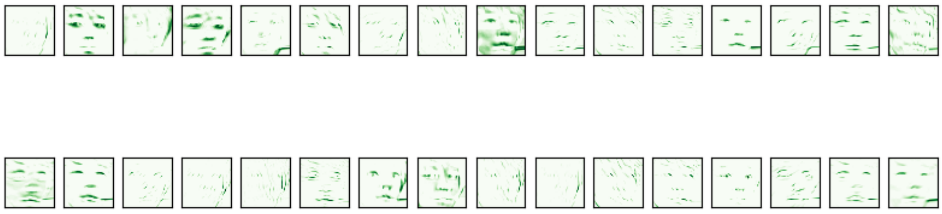
Output of layer conv2d_2 (Given image 48)



Output of layer conv2d_2 (Given image 79)



Output of layer conv2d_2 (Given image 13)



觀察 : 不同的 filter 會有不同的篩選標準，有些 filter 會把重點放在五官的輪廓，有些則凸顯特定部位，例如：第 12 個 filter 凸顯眉毛及嘴巴，第 23 個 filter 凸顯鼻樑……等。
由上面幾張圖可以發現第 3、8、12、16、23 個 filter 特別容易被 activate