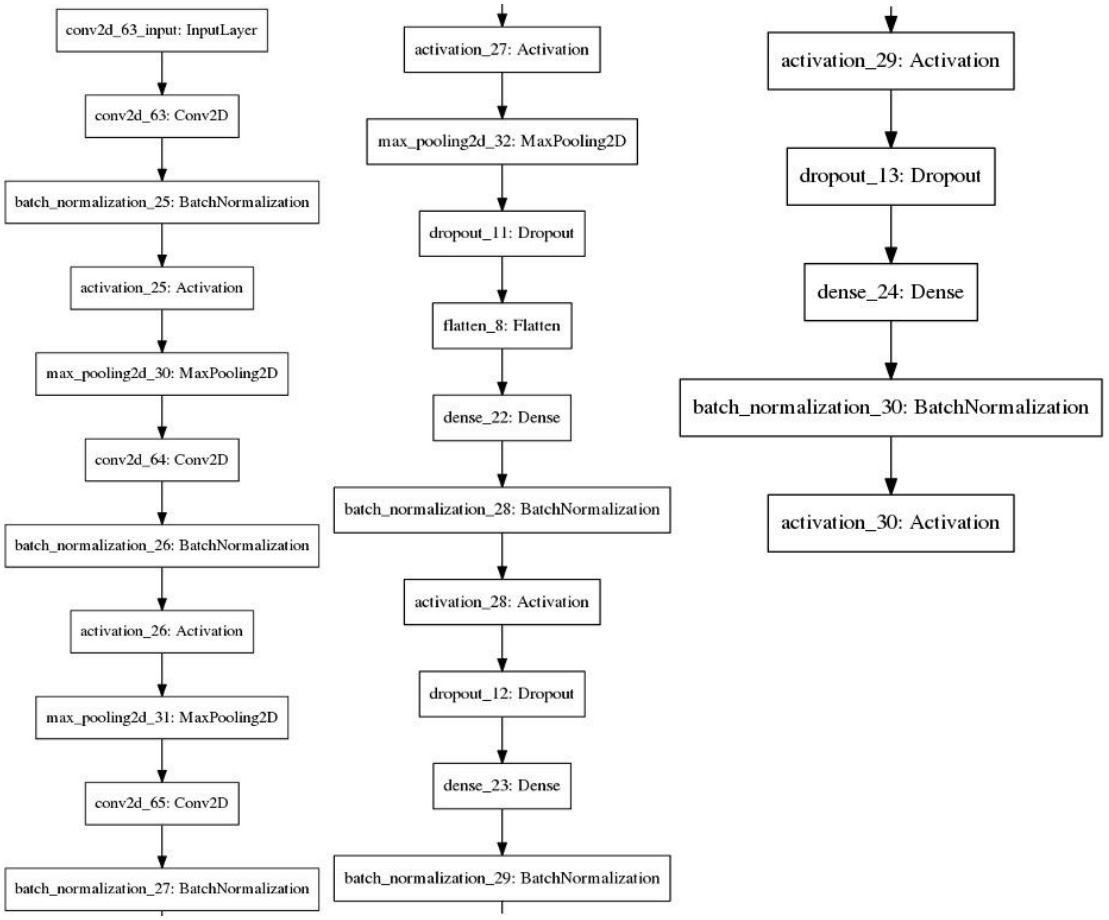
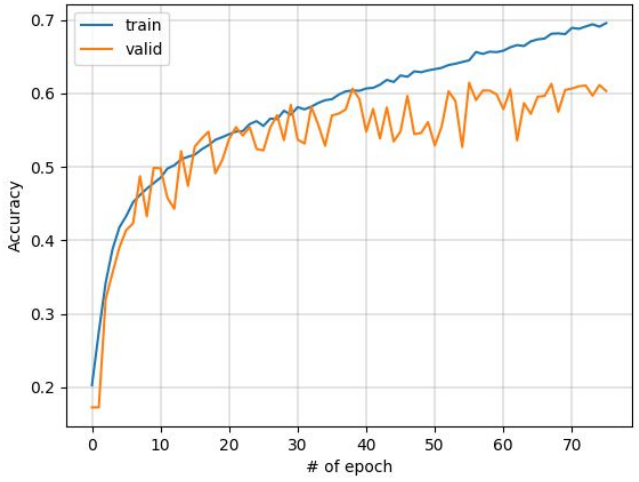
學號：B04501002 系級： 土木二 姓名：謝至宥

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: )




| Layer (type) | Output Shape | Param # |
|---|---|---|
| conv2d_6 (Conv2D) | (None, 46, 46, 64) | 640 |
| batch_normalization_7 (Batch | (None, 46, 46, 64) | 256 |
| activation_7 (Activation) | (None, 46, 46, 64) | 0 |
| max_pooling2d_4 (MaxPooling2 | (None, 23, 23, 64) | 0 |
| dropout_8 (Dropout) | (None, 23, 23, 64) | 0 |
| conv2d_7 (Conv2D) | (None, 22, 22, 128) | 32896 |
| batch_normalization_8 (Batch | (None, 22, 22, 128) | 512 |
| activation_8 (Activation) | (None, 22, 22, 128) | 0 |
| max_pooling2d_5 (MaxPooling2 | (None, 11, 11, 128) | 0 |
| dropout_9 (Dropout) | (None, 11, 11, 128) | 0 |
| conv2d_8 (Conv2D) | (None, 10, 10, 192) | 98496 |
| dropout_10 (Dropout) | (None, 10, 10, 192) | 0 |
| conv2d_9 (Conv2D) | (None, 9, 9, 192) | 147648 |
| dropout_11 (Dropout) | (None, 9, 9, 192) | 0 |
| conv2d_10 (Conv2D) | (None, 8, 8, 192) | 147648 |
| batch_normalization_9 (Batch | (None, 8, 8, 192) | 768 |
| activation_9 (Activation) | (None, 8, 8, 192) | 0 |
| max_pooling2d_6 (MaxPooling2 | (None, 4, 4, 192) | 0 |
| dropout_12 (Dropout) | (None, 4, 4, 192) | 0 |
| flatten_2 (Flatten) | (None, 3072) | 0 |
| dense_4 (Dense) | (None, 512) | 1573376 |
| batch_normalization_10 (Batc | (None, 512) | 2048 |
| activation_10 (Activation) | (None, 512) | 0 |
| dropout_13 (Dropout) | (None, 512) | 0 |
| dense_5 (Dense) | (None, 512) | 262656 |
| batch_normalization_11 (Batc | (None, 512) | 2048 |
| activation_11 (Activation) | (None, 512) | 0 |
| dropout_14 (Dropout) | (None, 512) | 0 |
| dense_6 (Dense) | (None, 7) | 3591 |
| batch_normalization_12 (Batc | (None, 7) | 28 |
| activation_12 (Activation) | (None, 7) | 0 |

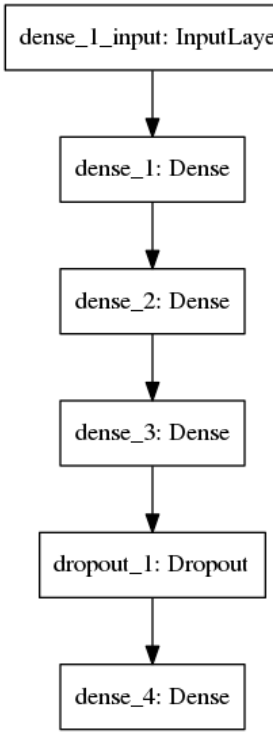
Total params: 2,272,611
Trainable params: 2,269,781
Non-trainable params: 2,830




我最後的模型使用了5層convolution layer，2層的dense(relu)，還有一層的softmax，共有7層的0.5 dropout。
訓練到50幾個epoch時，validation的正確率就已經不太會提升了，training則持續變好，應屬於overfitting，由於dropout已經很多了，若加入其他方式ensemble應該可以改善，可惜沒時間訓練。如果沒有dropout，trainging正確率會到0.8，validation則約0.4~0.5，overfitting現象更加明顯。

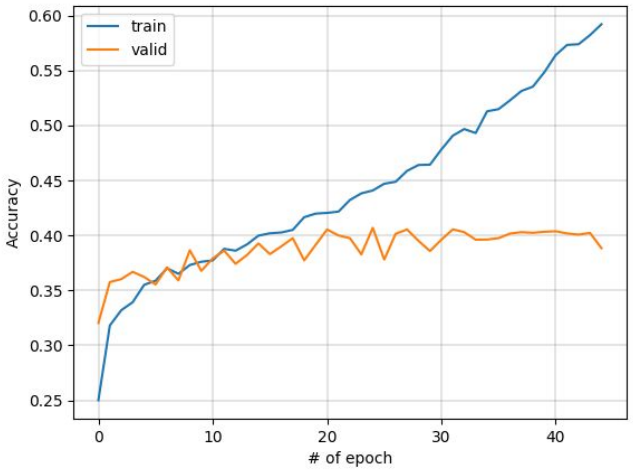
在訓練時也曾經試過13層conv2d加2層dense配relu，再一層softmax，但最終結果是練不起來，可能是在某個local minimum卡住的樣子。

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: )



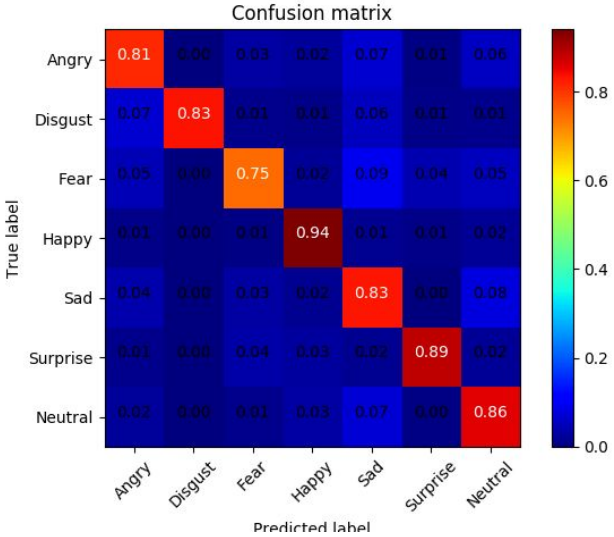

| Layer (type) | Output Shape | Param # |
|---|---|---|
| dense_1 (Dense) | (None, 1024) | 2360320 |
| dense_2 (Dense) | (None, 1024) | 1049600 |
| dense_3 (Dense) | (None, 756) | 774900 |
| dropout_1 (Dropout) | (None, 756) | 0 |
| dense_4 (Dense) | (None, 7) | 5299 |

Total params: 4,190,119
Trainable params: 4,190,119
Non-trainable params: 0



全部使用dense發現validation正確率更難提升，容易overfit，也更早就被early stopping終止，推測是cnn的filter發揮優勢，使相同的特徵可以表現在不同位置。不過全部使用dense的訓練時間更短，應該是少了convolution微分對電腦而言運算較快。

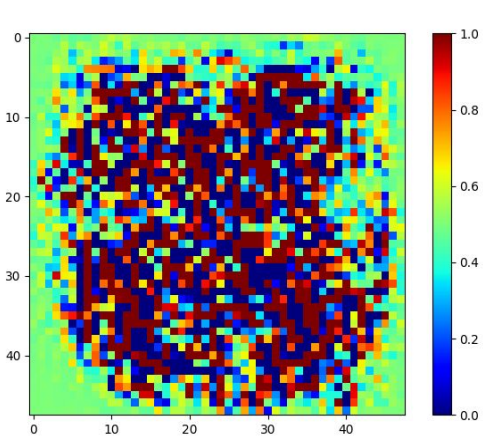
3. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析]
(Collaborators: )

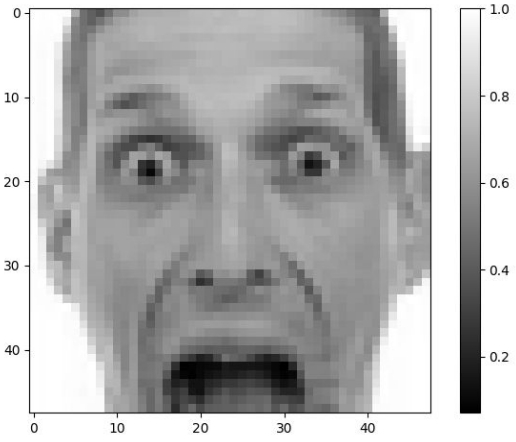


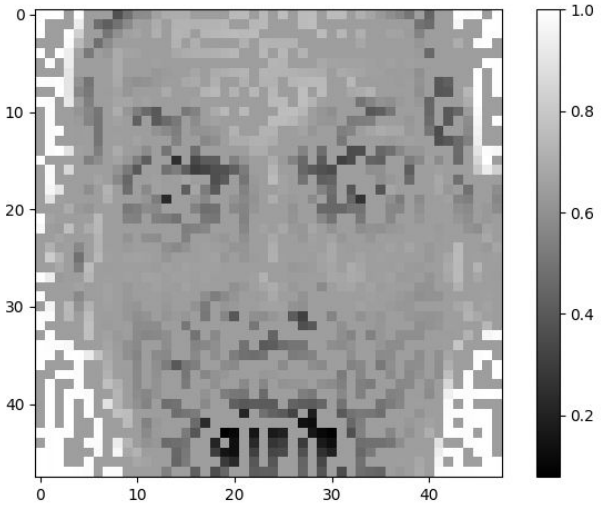

就從非對角線上的錯誤預測來看，恐懼誤判成傷心，傷心誤判成中立是比較多的。

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: )

從結果觀察到眼睛、嘴巴、鼻子是容易被感應的部位








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：