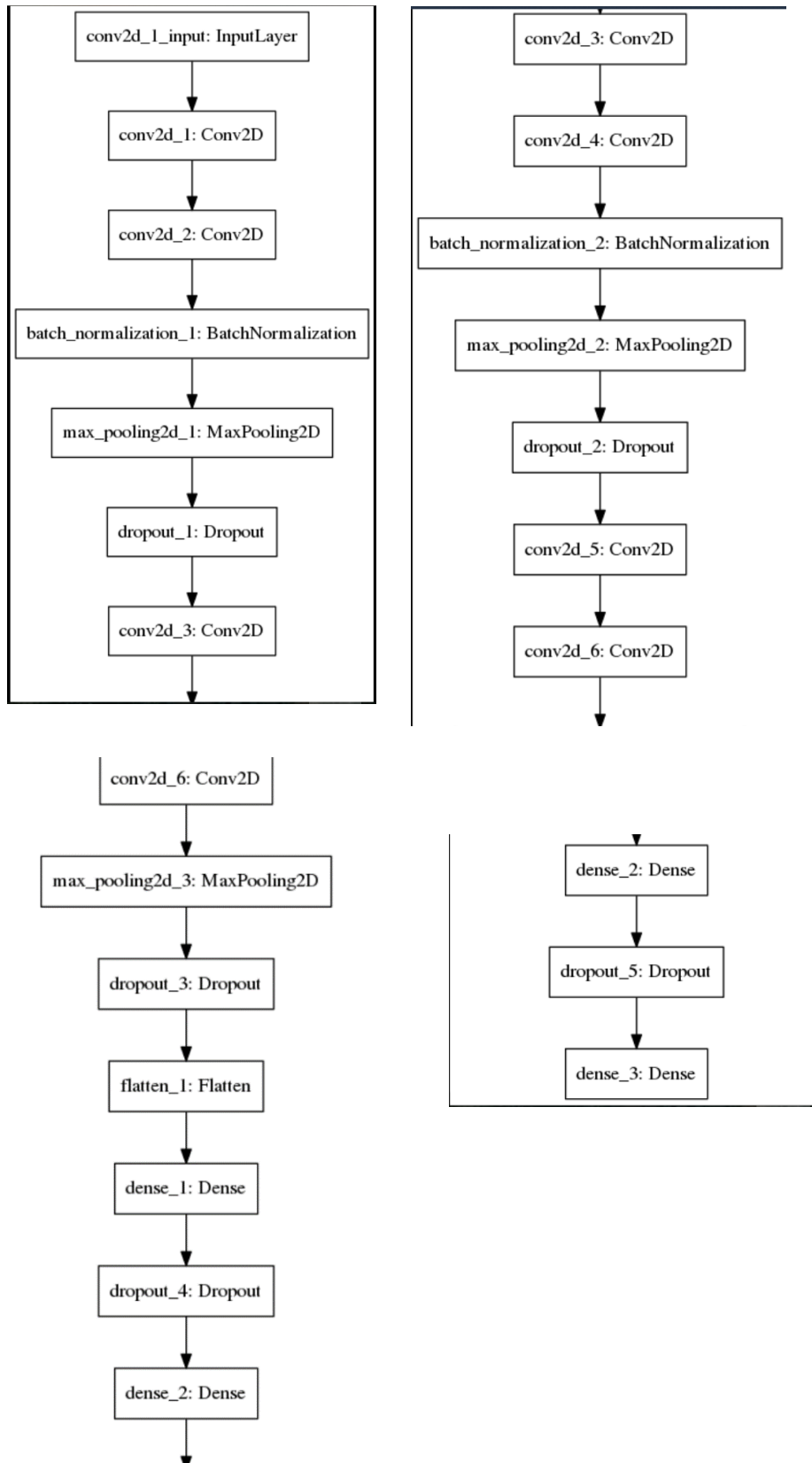


1. (1%) 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？  
(Collaborators: b03901079)

答：



Model 中共有 3,167,079 個參數，以下為使用的 model 說明，

Conv2D(): 使用 kernel\_size 為(3,3)，padding = “same”，activation = “relu”。

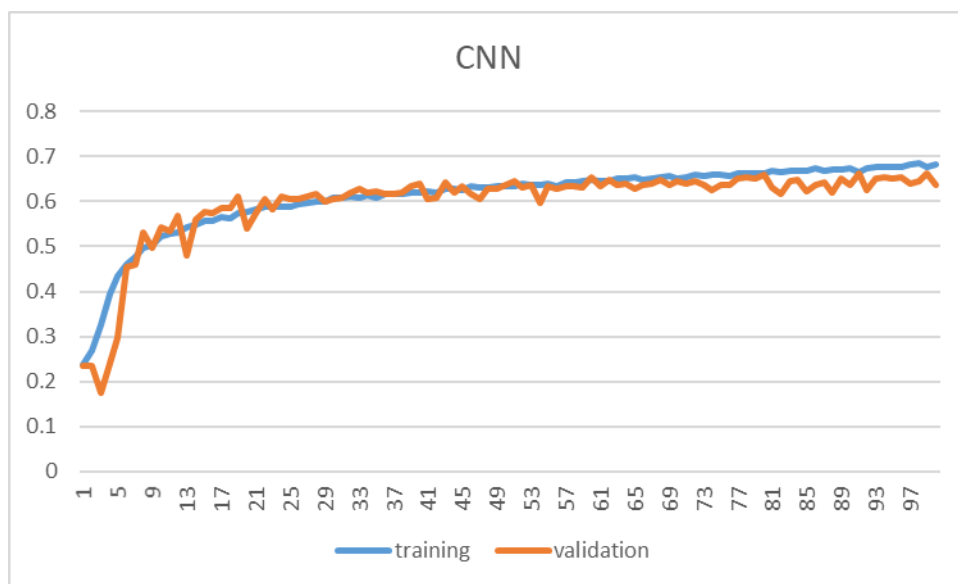
BatchNomalization(): 加快模型收斂的速度。

MaxPooling2D: pooling\_size 為(2,2)。

Dropout(): 避免模型 overfit。

經過三組捲積+池化層後，用 flatten()將特徵值轉為一維的資料讓連結層使用。

期間使用 ImageDataGenerater 將圖片平移或旋轉，產生更多的 training data，訓練模型以提升模型的準確率。



Validation 的 accuracy 在 0.65 左右，training 則靠近 0.7 且有上升趨勢。

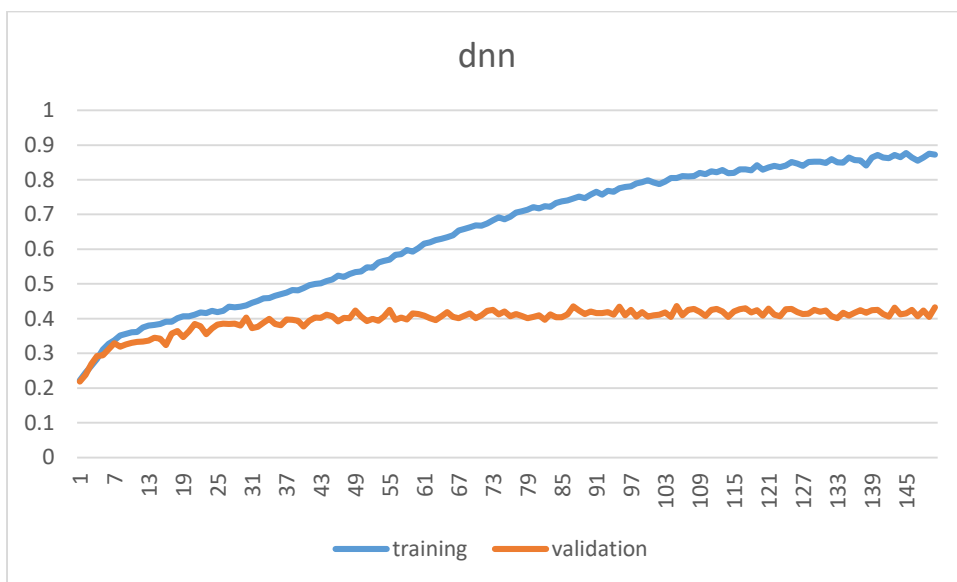
Kaggle 的 best score 是使用此模型產出五個 y\_test 預測值，利用投票的方式選出最好的所產生。

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: b03901079)

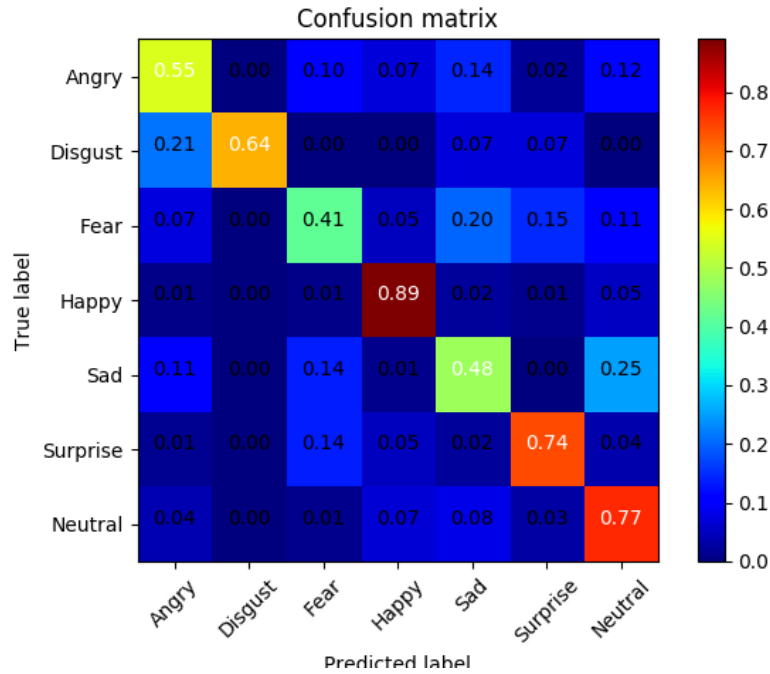
答：model 裡共有 3,674,119 個參數，

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 1024)	2360320
dense_2 (Dense)	(None, 1024)	1049600
dense_3 (Dense)	(None, 256)	262400
dropout_1 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 7)	1799
Total params: 3,674,119		
Trainable params: 3,674,119		
Non-trainable params: 0		



DNN 沒有經過 convolution，對於圖片特徵掌握較不佳，在 accuracy 上，DNN 的 validation 表現比 CNN 差，隨著 epoch 增加，從 training 的 accuracy 可以看出，DNN 容易有 overfit 的問題。但 DNN 的訓練速度比起 CNN 較為快速。

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: b03901079)



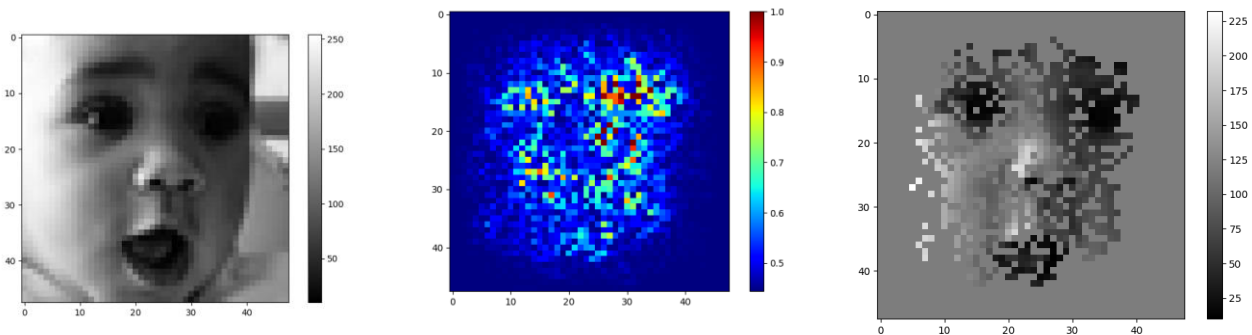
答：

從 confusion matrix 可以看到，最容易辨識錯誤的為將 sad 認為 neutral，其次則是將 disgust 辨識為 angry，以及將 fear 辨識為 sad。

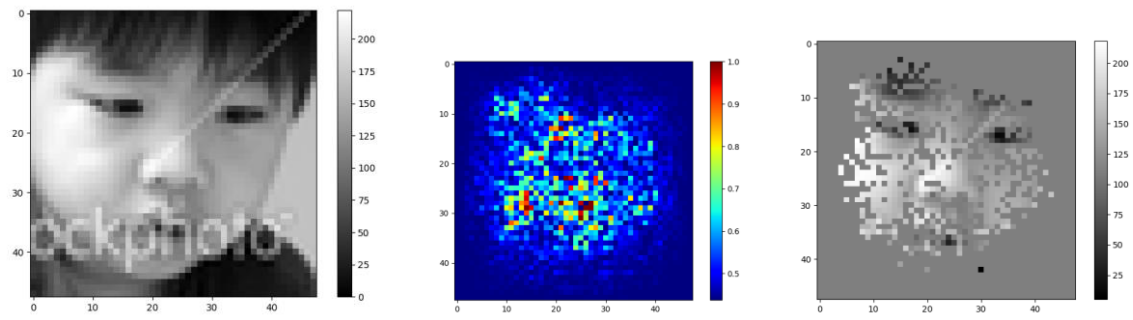
4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？  
(Collaborators: b03901079)

答：

在 saliency maps 中，我將 threshold 設為 0.5，找出模型是透過那些特徵去辨識圖片的，由 surprise 的圖片可以看出，發現 CNN 是透過眼睛及嘴巴的部分去辨別的圖片



而下面這張 sad 的圖片則是透過眉毛及眼睛的部位去辨識出該圖片。



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: b03901079)

答：

我們透過 **gradient ascent** 觀察最容易被 **activate**