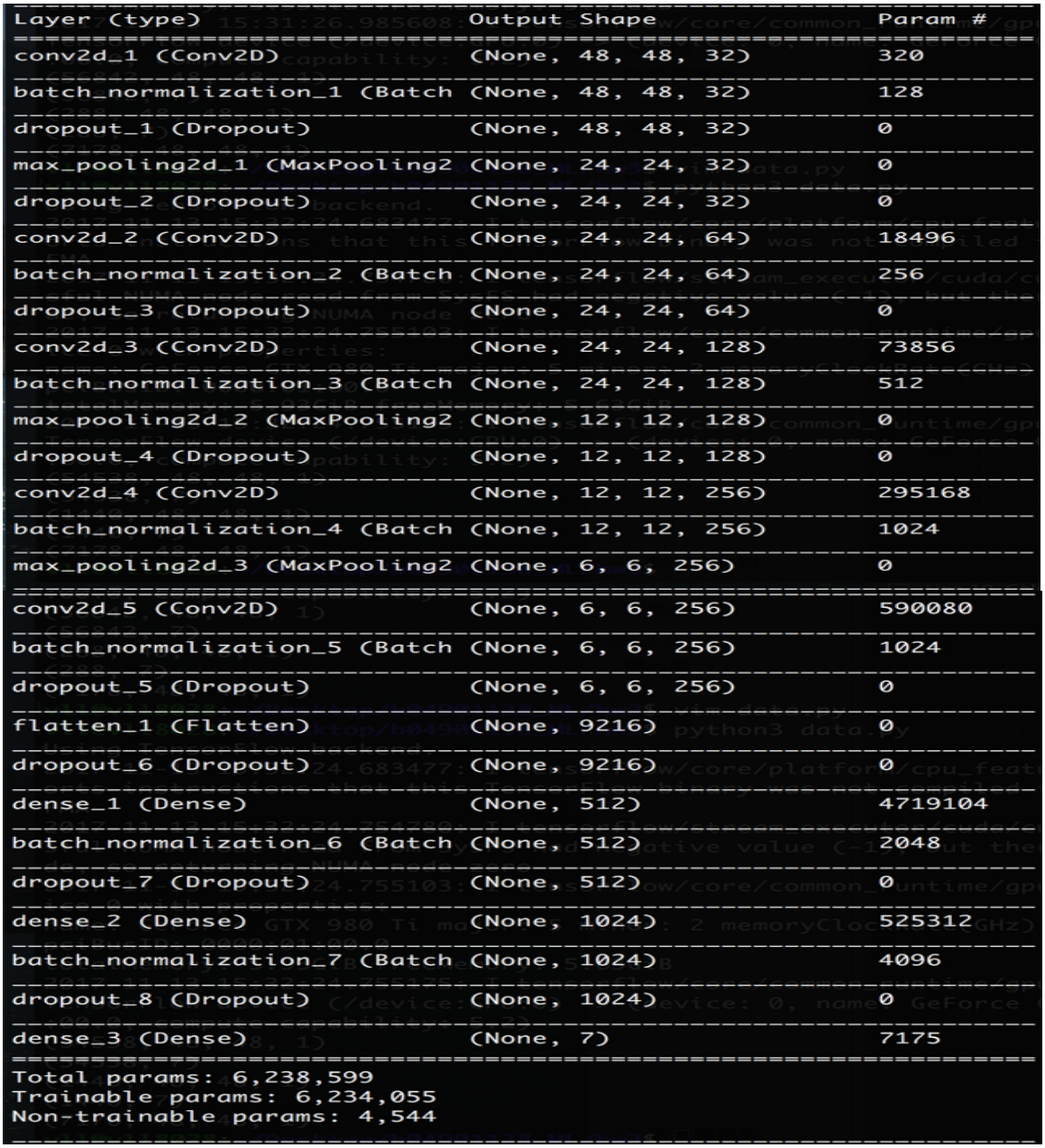
學號：B04901138 系級： 電機三　姓名：張景程

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: 無)



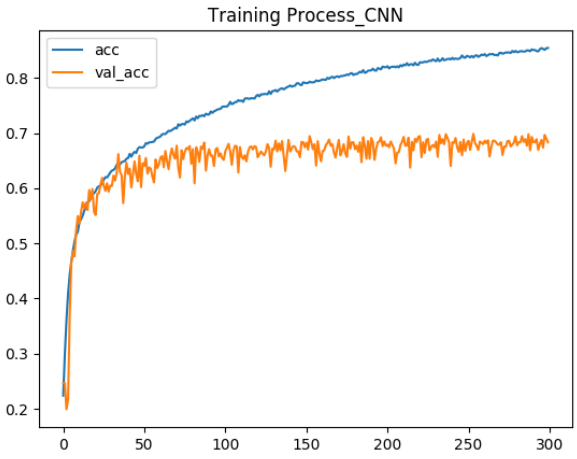
Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 32)	320
batch_normalization_1 (Batch	(None, 48, 48, 32)	128
dropout_1 (Dropout)	(None, 48, 48, 32)	0
max_pooling2d_1 (MaxPooling2	(None, 24, 24, 32)	0
dropout_2 (Dropout)	(None, 24, 24, 32)	0
conv2d_2 (Conv2D)	(None, 24, 24, 64)	18496
batch_normalization_2 (Batch	(None, 24, 24, 64)	256
dropout_3 (Dropout)	(None, 24, 24, 64)	0
conv2d_3 (Conv2D)	(None, 24, 24, 128)	73856
batch_normalization_3 (Batch	(None, 24, 24, 128)	512
max_pooling2d_2 (MaxPooling2	(None, 12, 12, 128)	0
dropout_4 (Dropout)	(None, 12, 12, 128)	0
conv2d_4 (Conv2D)	(None, 12, 12, 256)	295168
batch_normalization_4 (Batch	(None, 12, 12, 256)	1024
max_pooling2d_3 (MaxPooling2	(None, 6, 6, 256)	0
conv2d_5 (Conv2D)	(None, 6, 6, 256)	590080
batch_normalization_5 (Batch	(None, 6, 6, 256)	1024
dropout_5 (Dropout)	(None, 6, 6, 256)	0
flatten_1 (Flatten)	(None, 9216)	0
dropout_6 (Dropout)	(None, 9216)	0
dense_1 (Dense)	(None, 512)	4719104
batch_normalization_6 (Batch	(None, 512)	2048
dropout_7 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 1024)	525312
batch_normalization_7 (Batch	(None, 1024)	4096
dropout_8 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 7)	7175

Total params: 6,238,599
Trainable params: 6,234,055
Non-trainable params: 4,544

我總共用了五層 CNN，五層的 filter 數量分別為 32, 64, 128, 256, 256，每個 filter 大小皆為(3,3)，另外也做了三次 Maxpooling，後面再接三層 DNN，size 分別為 512, 1024, 7，最後再跑 400 個 epoch。

一開始在處理資料的時候，我切了 5%的 data 當作 validation，同時將每張 training image 做一次 flip，這樣可以得到將近兩倍數量的 training data。

另外我還使用了 keras 內建的 Imagedatagenerator，讓每個 training data 稍做一些位移和旋轉在開始 train，這招讓我的準確率提高不少。



準確率：kaggle

public : 0.68487

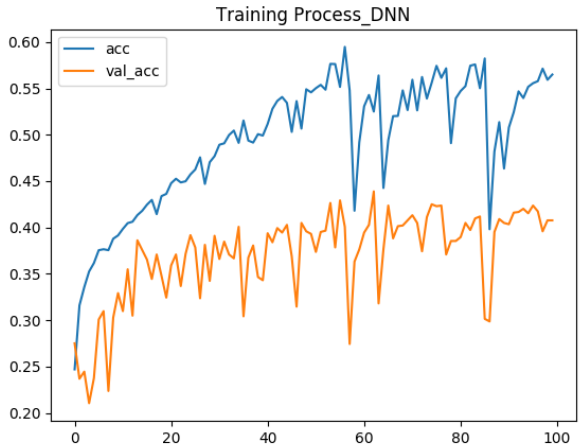
private : 0.70158

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: 無)

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 600)	1383000
dense_2 (Dense)	(None, 1024)	615424
batch_normalization_1 (Batch	(None, 1024)	4096
dropout_1 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 1024)	1049600
batch_normalization_2 (Batch	(None, 1024)	4096
dropout_2 (Dropout)	(None, 1024)	0
dense_4 (Dense)	(None, 1024)	1049600
batch_normalization_3 (Batch	(None, 1024)	4096
dropout_3 (Dropout)	(None, 1024)	0
dense_5 (Dense)	(None, 1024)	1049600
batch_normalization_4 (Batch	(None, 1024)	4096
dropout_4 (Dropout)	(None, 1024)	0
dense_6 (Dense)	(None, 1024)	1049600
batch_normalization_5 (Batch	(None, 1024)	4096
dropout_5 (Dropout)	(None, 1024)	0
dense_7 (Dense)	(None, 7)	7175

Total params: 6,224,479
Trainable params: 6,214,239
Non-trainable params: 10,240



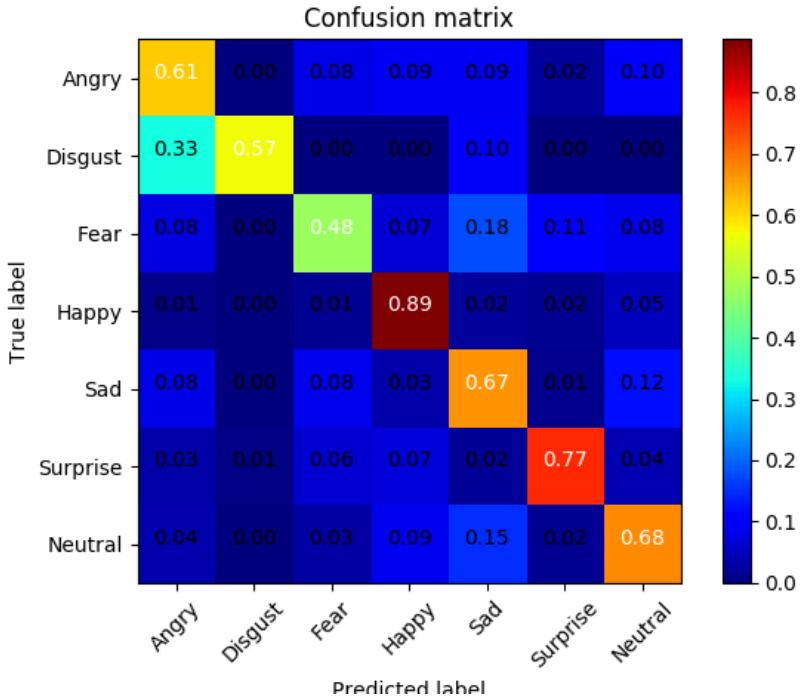
我疊了一個參數和第一小題差不多的 DNN model，準確率只有 40%多一點點，而且有 overfit 的現象，因為 DNN 沒辦法像 CNN 辨認兩個點合在一起的差異，這樣子 train 下去只會讓 model 學到一些奇奇怪怪的判斷原則，沒有太大的意義。

準確率：kaggle

public : 0.43103

private : 0.41404

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: 無)



由圖中可觀察出以下結果：

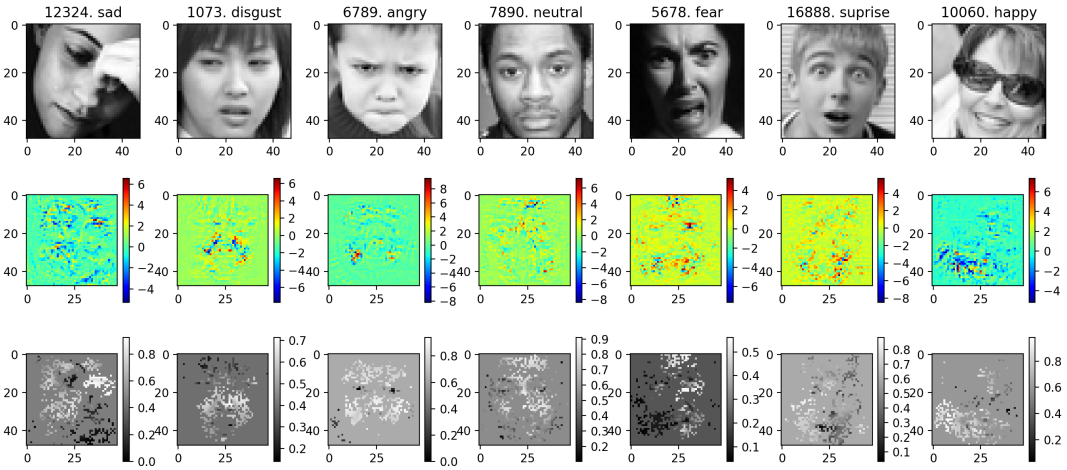
(1) Happy 的準確率最高(0.89)，Fear 最低(0.48)

(2) Disgust 最容易被混淆為 Angry (0.33)

(3) Fear 最容易被混淆為 Sad (0.18)

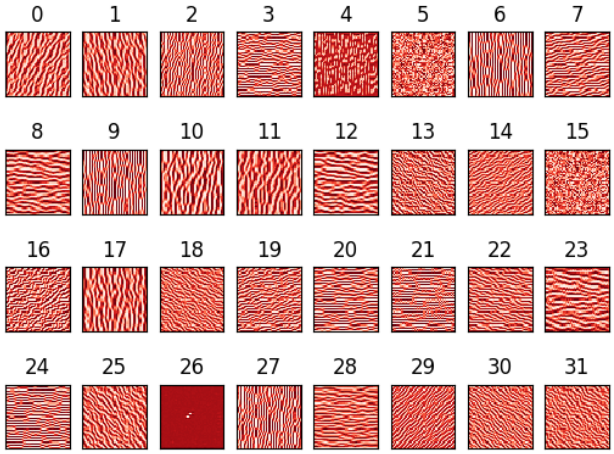
(4) Neutral 最容易被混淆為 Sad (0.15)

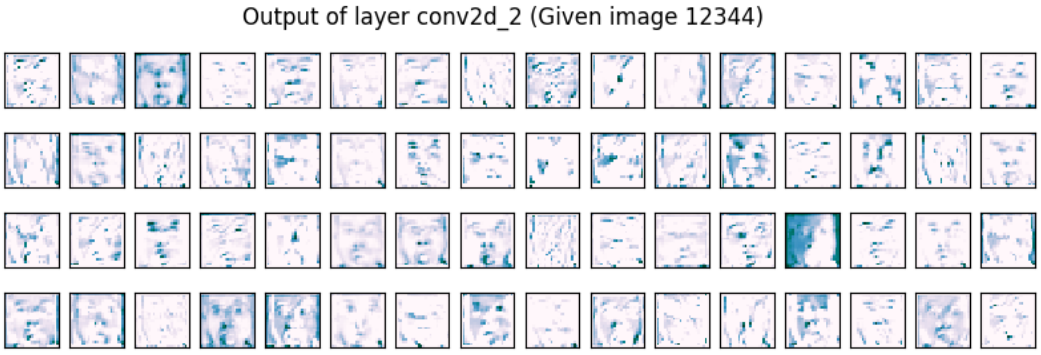
4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？ (Collaborators: B04901136 張家銘 / B04901019 梁書哲)



由 saliency map 可看出我的 model 在分辨影像時會 focus 在人臉的五官，尤其是眼睛和嘴巴的部分。以上圖最右邊的人臉為例，即使圖片中的人戴了墨鏡，model 仍可透過嘴巴的部分判斷出人露齒而笑，因而將之區分為 happy

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: B04901056 張承洋)





第一張圖是由我 model 中第一層 Conv2D 所繪出來，雖然每一個 filter 的工作都是辨別不一樣的 pattern，且從第四題的討論中我們知道 model 主要 focus 在眼睛和嘴巴的部分，但我們仍不容易看出這些東西和人臉之間的直接關聯。
第二張圖則是第二層 Conv2D 的 output 結果，我們可以觀察到，幾乎只剩下五官、臉型的部分還存在，頭髮、四周的東西幾乎已經被過濾掉。