

學號：R06942077 系級：電信碩一 姓名：洪健鈞

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？

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 44, 44, 64)	1664
zero_padding2d_1 (ZeroPadding2D)	(None, 48, 48, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 22, 22, 64)	0
zero_padding2d_2 (ZeroPadding2D)	(None, 24, 24, 64)	0
conv2d_2 (Conv2D)	(None, 22, 22, 64)	36928
zero_padding2d_3 (ZeroPadding2D)	(None, 24, 24, 64)	0
conv2d_3 (Conv2D)	(None, 22, 22, 64)	36928
average_pooling2d_1 (AveragePooling2D)	(None, 10, 10, 64)	0
zero_padding2d_4 (ZeroPadding2D)	(None, 12, 12, 64)	0
conv2d_4 (Conv2D)	(None, 10, 10, 128)	73856
zero_padding2d_5 (ZeroPadding2D)	(None, 12, 12, 128)	0
conv2d_5 (Conv2D)	(None, 10, 10, 128)	147584
zero_padding2d_6 (ZeroPadding2D)	(None, 12, 12, 128)	0
average_pooling2d_2 (AveragePooling2D)	(None, 5, 5, 128)	0
flatten_1 (Flatten)	(None, 3200)	0
dense_1 (Dense)	(None, 1024)	3277824
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
dropout_2 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 7)	7175
Total params: 4,631,559		
Trainable params: 4,631,559		
Non-trainable params: 0		

答：Exception ignored in: <bound method BaseSession.\_del\_ of <tensorflow.python...

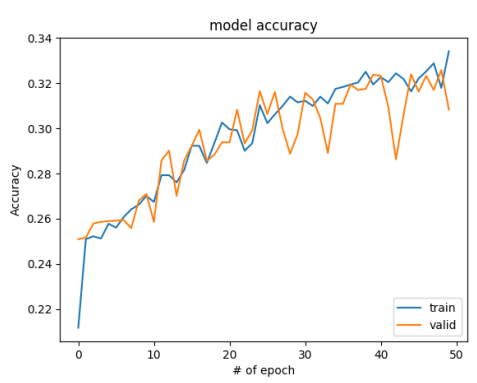
CNN 模型架構參考助教的 **sample code**，並將 **training set** 做一些處理，**optimizer** 用 **Adadelta**，及做 **convolution2D** 和 **maxpooling**，主要目的在強化 **sentimental** 的特徵以及降低維度，可以讓 **model** 的準確率上升，另外使用 **dropout** 來防止 **overfitting**，另外設定 **earlystop** 若 **accuracy** 開始下降即停止 **training**。訓練準確率的部分，在 **training** 的部分有切 5% 的 **validation\_set** 來驗證，準確率大約為 0.7 多，上傳 **kaggle** 準確率大約 0.67 左右。

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？

max_pooling2d_1 (MaxPooling2 (None, 24, 24, 1))	0	
zero_padding2d_2 (ZeroPaddin (None, 26, 26, 1))	0	
zero_padding2d_3 (ZeroPaddin (None, 28, 28, 1))	0	
average_pooling2d_1 (Average (None, 13, 13, 1))	0	
zero_padding2d_4 (ZeroPaddin (None, 15, 15, 1))	0	
zero_padding2d_5 (ZeroPaddin (None, 17, 17, 1))	0	
zero_padding2d_6 (ZeroPaddin (None, 19, 19, 1))	0	
average_pooling2d_2 (Average (None, 9, 9, 1))	0	
flatten_1 (Flatten)	(None, 81)	0
dense_1 (Dense)	(None, 1024)	83968
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
dropout_2 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 7)	7175
activation_1 (Activation)	(None, 7)	0
=====		

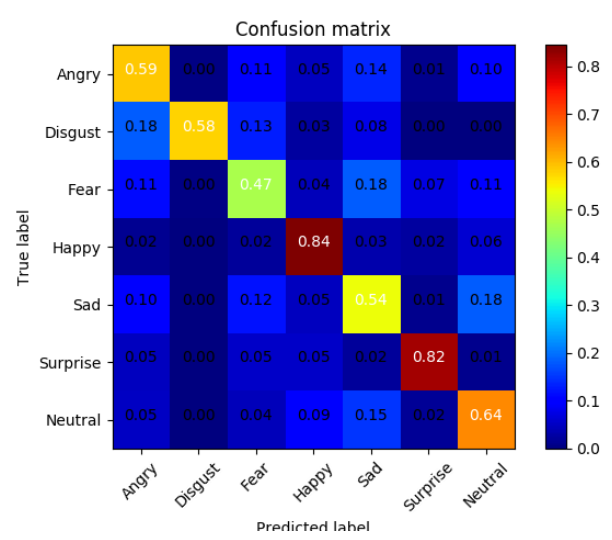
答：

架構類似於第一題的架構，只不過這裡拿掉了 **Convolution2D**，不過準確率也因此大大的下降，因為沒用 **Conv2D** 來強化特徵點，降低了準確率。準確率剩不到 **0.4**。



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]

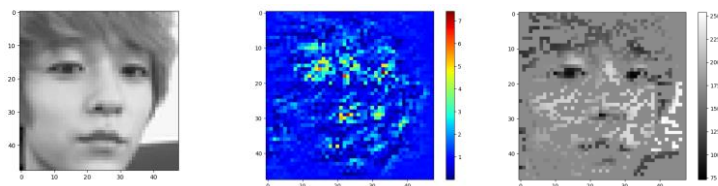
答：



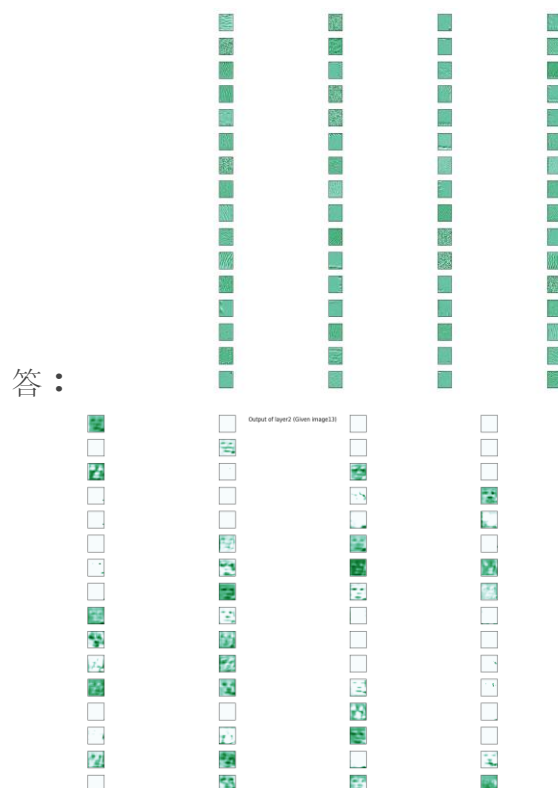
Sad 及 Fear 最容易互相搞混，可能是因為 Sad 和 Fear 可能都是屬於負面情緒，sad 哭時候的表情看起來會很像 fear 因此非常容易搞混，Fear 有 18% 機率被預測成 sad，Sad 則有 12% 機率被分為 Fear。

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: )

答：說到人臉的特徵點，最重要的大概就是五官了，其中又以眼睛與嘴巴更為明顯，最容易以他們來分辨這張圖片的情緒，所以 saliency map 的熱點也都集中在眼睛和嘴巴附近。



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**。



五官明顯的人容易被 activate,因為可以更容易讓 filter 被 activate 產生出一些想要的特徵點,例如眼睛及嘴巴。