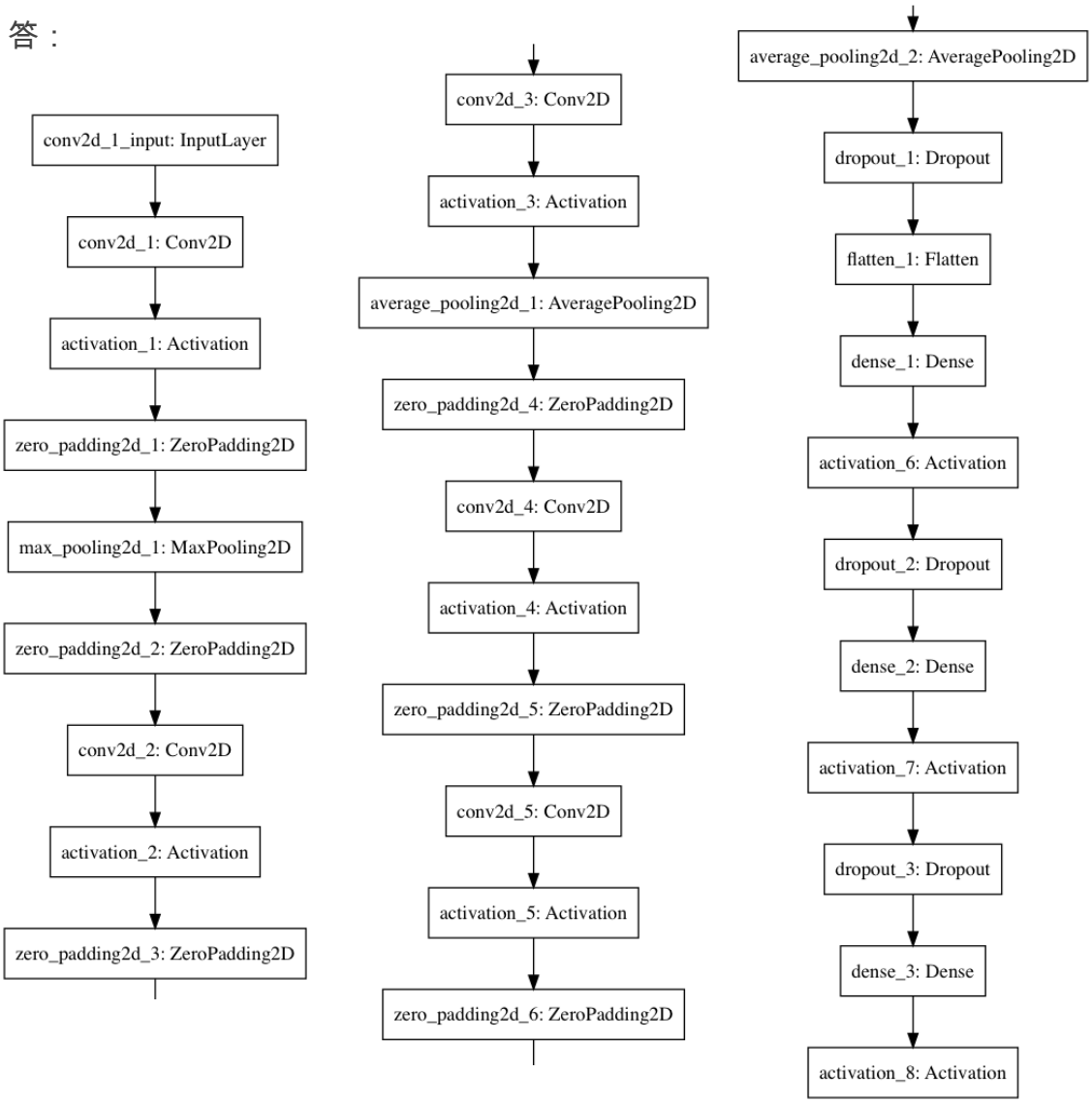
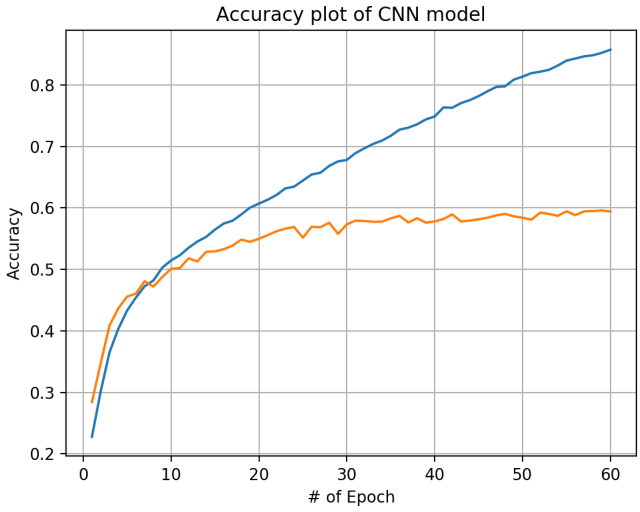
學號：R06922097 系級： 資工碩一 姓名：鄭雅文

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？

答：



照助教的 sample code，疊一層 64 個 5*5 的 filters 的 CNN，第二層和第三層是 64 個 3*3 的 filters，第四層和第五層是 128 個 3*3 的 filters，最後再疊兩層 1024 個 neural 的 fully connected network。中間也是照助教的 sample code 做 ZeroPadding、Maxpooling、AveragePooling、Dropout。



在 training 的過程中 accuracy 不斷上升，validation set 則漸趨 0.6。在 kaggle public testing set 上的 accuracy 為 **0.61186**。

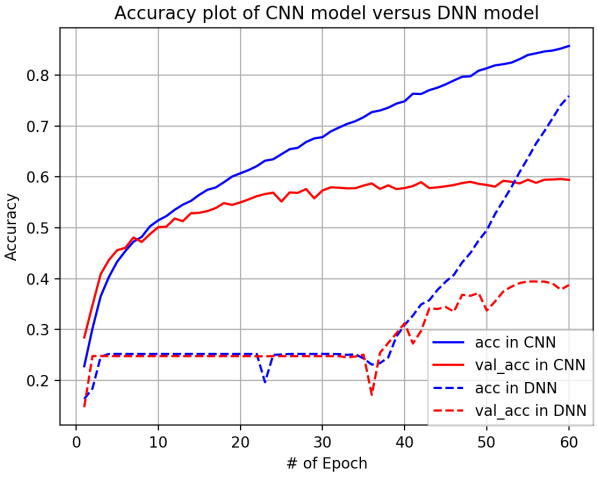
2. **(1%)** 承上題，請用與上述 **CNN** 接近的參數量，實做簡單的 **DNN model**。其模型架構、訓練過程和準確率為何？試與上題結果做比較，並說明你觀察到了什麼？

答：DNN 中的 input 為 48*48=2304 維的陣列，每層的 output 設為 filter 邊長的平方*filter 數，如 64 個 5*5 的 filters，則 output 為 64*5*5 = 1600，則計算出來的 param # = (2304+1)*1600 = 3688000，以此類推。

DNN model summary

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 1600)	3688000
activation_1 (Activation)	(None, 1600)	0
dense_2 (Dense)	(None, 576)	922176
activation_2 (Activation)	(None, 576)	0
dense_3 (Dense)	(None, 576)	332352
activation_3 (Activation)	(None, 576)	0
dense_4 (Dense)	(None, 1152)	664704
activation_4 (Activation)	(None, 1152)	0
dense_5 (Dense)	(None, 1152)	1328256
activation_5 (Activation)	(None, 1152)	0
dense_6 (Dense)	(None, 1024)	1180672
activation_6 (Activation)	(None, 1024)	0
dense_7 (Dense)	(None, 1024)	1049600
activation_7 (Activation)	(None, 1024)	0
dense_8 (Dense)	(None, 7)	7175
activation_8 (Activation)	(None, 7)	0

Total params: 9,172,935
Trainable params: 9,172,935
Non-trainable params: 0



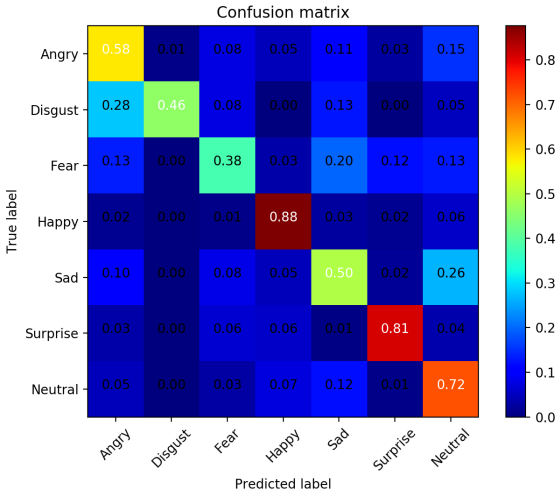
CNN model summary (上題)

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 64)	1664
activation_1 (Activation)	(None, 48, 48, 64)	0
zero_padding2d_1 (ZeroPaddin	(None, 52, 52, 64)	0
max_pooling2d_1 (MaxPooling2	(None, 24, 24, 64)	0
zero_padding2d_2 (ZeroPaddin	(None, 26, 26, 64)	0
conv2d_2 (Conv2D)	(None, 26, 26, 64)	36928
activation_2 (Activation)	(None, 26, 26, 64)	0
zero_padding2d_3 (ZeroPaddin	(None, 28, 28, 64)	0
conv2d_3 (Conv2D)	(None, 28, 28, 64)	36928
activation_3 (Activation)	(None, 28, 28, 64)	0
average_pooling2d_1 (Average	(None, 13, 13, 64)	0
zero_padding2d_4 (ZeroPaddin	(None, 15, 15, 64)	0
conv2d_4 (Conv2D)	(None, 15, 15, 128)	73856
activation_4 (Activation)	(None, 15, 15, 128)	0
zero_padding2d_5 (ZeroPaddin	(None, 17, 17, 128)	0
conv2d_5 (Conv2D)	(None, 17, 17, 128)	147584
activation_5 (Activation)	(None, 17, 17, 128)	0
zero_padding2d_6 (ZeroPaddin	(None, 19, 19, 128)	0
average_pooling2d_2 (Average	(None, 9, 9, 128)	0
dropout_1 (Dropout)	(None, 9, 9, 128)	0
flatten_1 (Flatten)	(None, 10368)	0
dense_1 (Dense)	(None, 1024)	10617856
activation_6 (Activation)	(None, 1024)	0
dropout_2 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
activation_7 (Activation)	(None, 1024)	0
dropout_3 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 7)	7175
activation_8 (Activation)	(None, 7)	0

Total params: 11,971,591
Trainable params: 11,971,591
Non-trainable params: 0

使用相似參數量及相同的層數，此 DNN 模型在 kaggle public testing set 上的 accuracy 是 **0.38506**，且在訓練時，testing set 和 validation set 的 accuracy 都沒有 CNN 的 accuracy 高。可推測 CNN 在偵測圖片的特徵上，效果比 DNN 沒有考慮圖片特徵的相對位置的情況好，所以 CNN 比 DNN 適合用在圖片特徵辨識上。

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]**

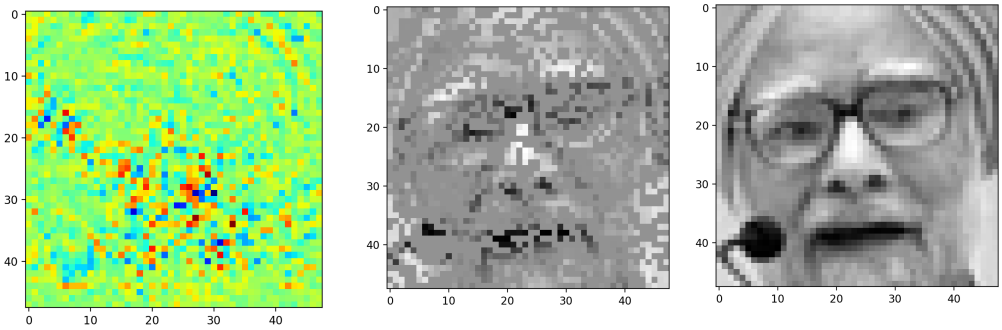


答：

觀察對角線以外的格子，Disgust 會容易被誤認成 Angry，Sad 容易被誤認為 Neural，其次是 Fear 容易被誤認為 Sad。在現實情況中這些表情確實較難被分辨出來。

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？**

答：



根據此張圖(training data #1)，約 focus 在眼睛、嘴巴(深色)的部分，**較容易被凸顯出來，另外其他的作圖如果臉的部分不夠多的話，經過** saliency map **會看不出來** filter **選擇的區域跟表情的關係。**

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: https://gist.github.com/howard1337/f416e5aebd98fbbebdd43a137009757e)

答：

此圖為 training set #1，label 為 0(Angry)，第一排第一個、第二排第一個和第六個和第十個、第三排第七個和第十三個、第四排第二和第四個和八個 filter 過後臉型較明顯，因此這些 filter 可能就是分辨 Angry 的重要 filter。