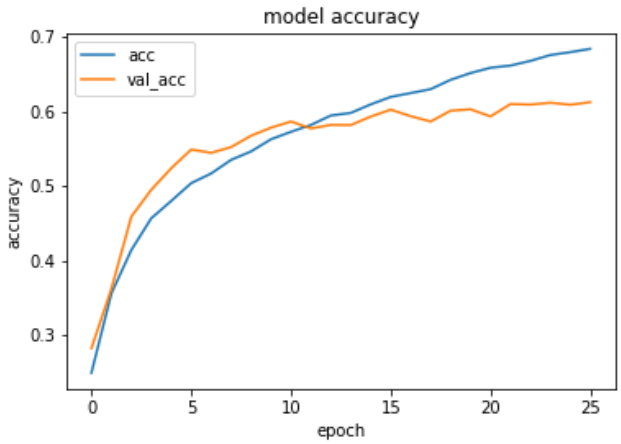
學號：R05229016 系級： 大氣碩二 姓名：羅章碩

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: 鄒適文)**

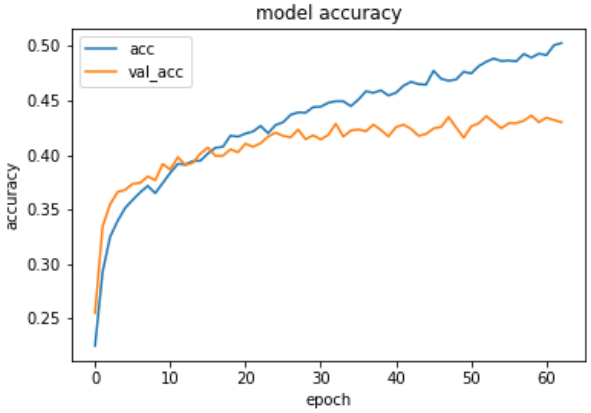
   CNN 模型的架構因為很長所以我放在 report 的最後面，使用了四層卷積層，三層的 Fully-connected layers，其中有加入 dropout 和 batchnormalization 降低 overfitting 的產生。訓練過程的 acc 和 val_acc 的變化如下圖：



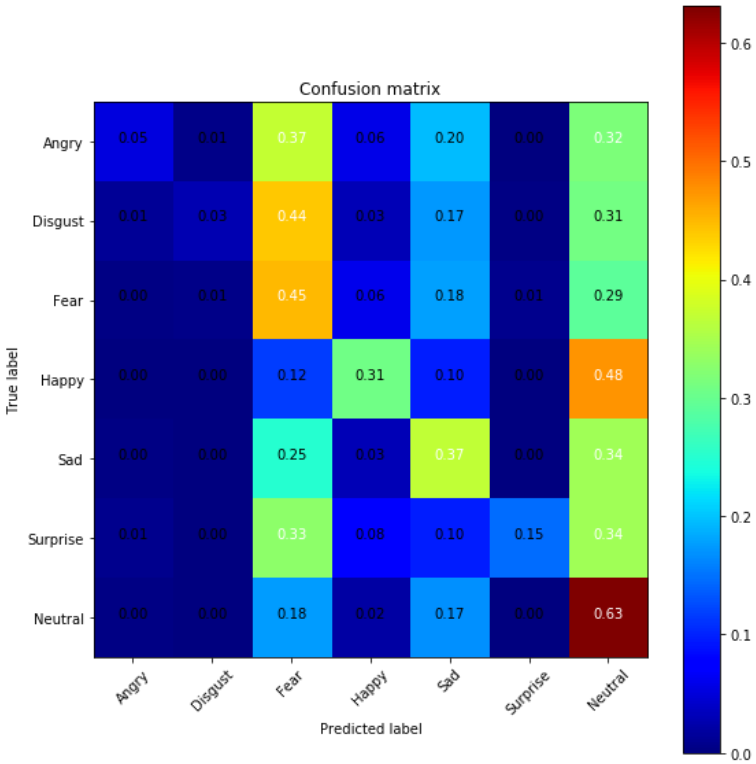
   可以看到 acc 隨著 epoch 越來越高，但是 val_acc 在 5 個 epoch 之後準確率就緩慢的上升，因為我有使用 EarlyStopping 的函數，所以當 val_acc 沒有上升反而下降的時候會停止訓練，以防有 overfitting 的情況產生。而此模型在 kaggle 上的分數有達到接近 0.62 的分數。

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:)**

   DNN 模型的架構同樣也很長所以我也放在 report 的最後面。上一題 CNN 的參數量大約為 59 萬，而此 DNN 的模型使用了約 61 萬的參數量，但在預測上的結果只有 0.43 的正確率，明顯比 CNN 低很多。以下是 DNN 在訓練過程中 acc 和 val_acc 的變化，很明顯可以看到 val_acc 在大約 0.4 多的時候已經升不上去了。從此可以看出參數量差不多的 DNN 和 CNN (或是 DNN 數量多一點點)，在人臉表情辨識上 CNN 有很高的優勢，因為多了卷積層，它可以更準確的訓練模型。

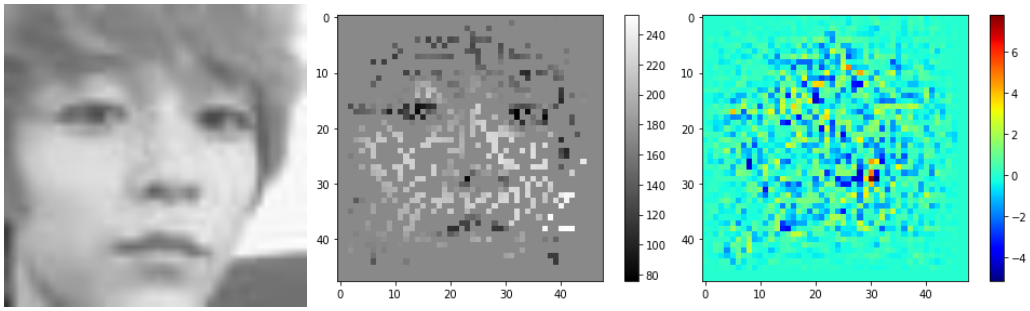


3. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析]



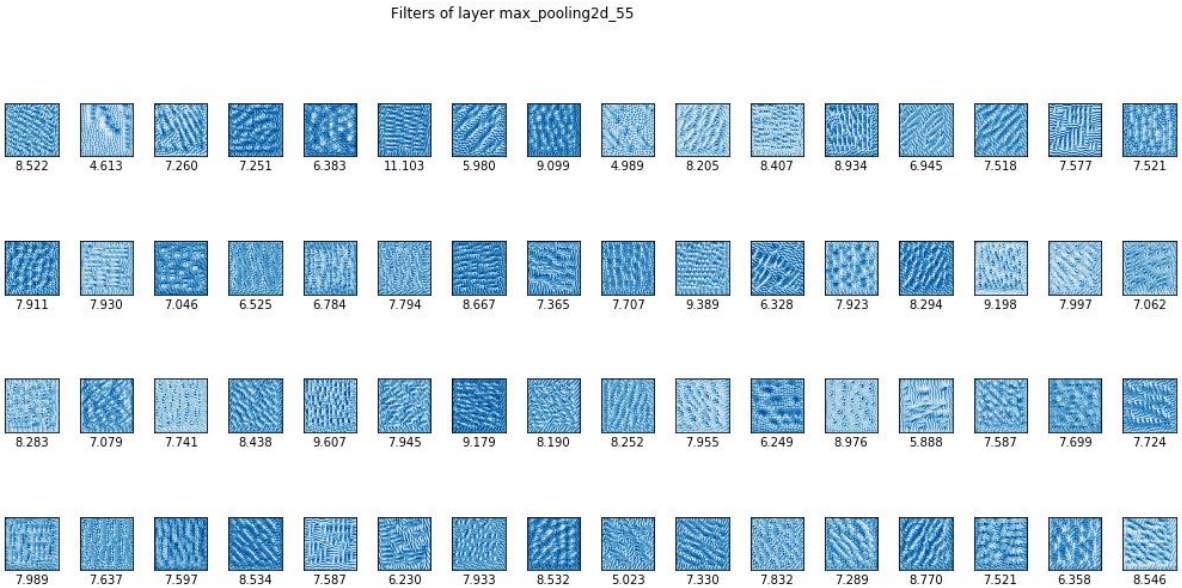
上圖是我用 model 預測 validation set 畫出的結果。其實我有畫了不同 model 的結果比較，發現 model 在預測 Neutral 的表情時最為準確，預測 Disgust 和 Angry 的結果都很差，尤其是 model 幾乎不會預測到 Disgust 的表情。後來去確認了訓練資料發現 Disgust 才 400 多筆資料，跟其他表情來比資料筆數少很多，這有可能造成 model 在辨認 Disgust 表情上表現很差的其中一個原因。除此之外，從上圖可以看到 model 容易將 Disgust、Angry 和 Surprise 的表情誤認成 Fear，有一部分表情也會誤認成 Neutral。

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: 鄒適文)



從這幾張圖來看感覺 model 會比較 focus 在眼睛、鼻子跟嘴巴上面。如果有張大嘴巴的話，模型更會注意在嘴巴上，像是畫出一個有張嘴或是抿嘴的 Saliency map 時就可以發現嘴巴附近在 heatmap 上有很高的分數。除了嘴巴，model 也會注意到眼睛的樣子，像是皺眉或是挑眉等會讓眼睛形狀比較不一樣，不過這一點也讓 model 在辨認時會出錯，像是我 model 中就很容易把 angry 和 disgust 的表情誤認成 fear，因為這些表情中眼睛都是有一種厭惡的樣子。

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: 鄒適文)
   下兩張圖是測試上一題中圖片的結果，感覺是測點點狀的 filter 和斜線紋路的 filter 最容易被 activate，而直的紋路則比較還好。



Output of layer0 (Given image13)

CNN model:

Layer (type)	Output Shape	Param #
conv2d_57 (Conv2D)	(None, 46, 46, 32)	320
activation_54 (Activation)	(None, 46, 46, 32)	0
max_pooling2d_54 (MaxPooling	(None, 23, 23, 32)	0
dropout_72 (Dropout)	(None, 23, 23, 32)	0
batch_normalization_69 (Batc	(None, 23, 23, 32)	128
conv2d_58 (Conv2D)	(None, 21, 21, 64)	18496
activation_55 (Activation)	(None, 21, 21, 64)	0
max_pooling2d_55 (MaxPooling	(None, 10, 10, 64)	0
dropout_73 (Dropout)	(None, 10, 10, 64)	0
batch_normalization_70 (Batc	(None, 10, 10, 64)	256
conv2d_59 (Conv2D)	(None, 8, 8, 128)	73856
activation_56 (Activation)	(None, 8, 8, 128)	0
max_pooling2d_56 (MaxPooling	(None, 4, 4, 128)	0
dropout_74 (Dropout)	(None, 4, 4, 128)	0
batch_normalization_71 (Batc	(None, 4, 4, 128)	512
conv2d_60 (Conv2D)	(None, 2, 2, 256)	295168
activation_57 (Activation)	(None, 2, 2, 256)	0
max_pooling2d_57 (MaxPooling	(None, 1, 1, 256)	0
dropout_75 (Dropout)	(None, 1, 1, 256)	0
flatten_11 (Flatten)	(None, 256)	0
batch_normalization_72 (Batc	(None, 256)	1024
dense_37 (Dense)	(None, 128)	32896
dropout_76 (Dropout)	(None, 128)	0
batch_normalization_73 (Batc	(None, 128)	512
dense_38 (Dense)	(None, 256)	33024
dropout_77 (Dropout)	(None, 256)	0
batch_normalization_74 (Batc	(None, 256)	1024
dense_39 (Dense)	(None, 512)	131584

```
_________________________________________________________________
dropout_78 (Dropout)         (None, 512)               0
_________________________________________________________________
dense_40 (Dense)             (None, 7)                 3591
=================================================================
Total params: 592,391
Trainable params: 590,663
Non-trainable params: 1,728
_________________________________________________________________
```

DNN model:

```
Layer (type)                 Output Shape              Param #
=================================================================
flatten_16 (Flatten)         (None, 2304)              0
_________________________________________________________________
dense_84 (Dense)             (None, 128)               295040
_________________________________________________________________
dropout_29 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_85 (Dense)             (None, 128)               16512
_________________________________________________________________
batch_normalization_50 (Batc (None, 128)               512
_________________________________________________________________
dense_86 (Dense)             (None, 256)               33024
_________________________________________________________________
dropout_30 (Dropout)         (None, 256)               0
_________________________________________________________________
dense_87 (Dense)             (None, 256)               65792
_________________________________________________________________
batch_normalization_51 (Batc (None, 256)               1024
_________________________________________________________________
dense_88 (Dense)             (None, 256)               65792
_________________________________________________________________
dropout_31 (Dropout)         (None, 256)               0
_________________________________________________________________
batch_normalization_52 (Batc (None, 256)               1024
_________________________________________________________________
dense_89 (Dense)             (None, 512)               131584
_________________________________________________________________
batch_normalization_53 (Batc (None, 512)               2048
_________________________________________________________________
dense_90 (Dense)             (None, 7)                 3591
=================================================================
Total params: 615,943
Trainable params: 613,639
Non-trainable params: 2,304
```