我忘記 public 最好的 model 了（0.62970），所以以下報告為目前有的最好的 model（0.62217）

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: 黃敬庭、倪溥辰**)**

   模型架構：

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 44, 44, 64)	1664
batch_normalization_1 (Batch	(None, 44, 44, 64)	256
activation_1 (Activation)	(None, 44, 44, 64)	0
zero_padding2d_1 (ZeroPaddin	(None, 48, 48, 64)	0
conv2d_2 (Conv2D)	(None, 46, 46, 64)	36928
batch_normalization_2 (Batch	(None, 46, 46, 64)	256
activation_2 (Activation)	(None, 46, 46, 64)	0
zero_padding2d_2 (ZeroPaddin	(None, 48, 48, 64)	0
max_pooling2d_1 (MaxPooling2	(None, 22, 22, 64)	0
dropout_1 (Dropout)	(None, 22, 22, 64)	0
zero_padding2d_3 (ZeroPaddin	(None, 24, 24, 64)	0
conv2d_3 (Conv2D)	(None, 22, 22, 64)	36928
batch_normalization_3 (Batch	(None, 22, 22, 64)	256
activation_3 (Activation)	(None, 22, 22, 64)	0
zero_padding2d_4 (ZeroPaddin	(None, 24, 24, 64)	0
conv2d_4 (Conv2D)	(None, 22, 22, 64)	36928
batch_normalization_4 (Batch	(None, 22, 22, 64)	256
activation_4 (Activation)	(None, 22, 22, 64)	0
zero_padding2d_5 (ZeroPaddin	(None, 24, 24, 64)	0
max_pooling2d_2 (MaxPooling2	(None, 11, 11, 64)	0
dropout_2 (Dropout)	(None, 11, 11, 64)	0
zero_padding2d_6 (ZeroPaddin	(None, 13, 13, 64)	0
conv2d_5 (Conv2D)	(None, 11, 11, 64)	36928
batch_normalization_5 (Batch	(None, 11, 11, 64)	256
activation_5 (Activation)	(None, 11, 11, 64)	0
zero_padding2d_7 (ZeroPaddin	(None, 13, 13, 64)	0
conv2d_6 (Conv2D)	(None, 11, 11, 64)	36928
batch_normalization_6 (Batch	(None, 11, 11, 64)	256
activation_6 (Activation)	(None, 11, 11, 64)	0
max_pooling2d_3 (MaxPooling2	(None, 5, 5, 64)	0
dropout_3 (Dropout)	(None, 5, 5, 64)	0
zero_padding2d_8 (ZeroPaddin	(None, 7, 7, 64)	0
conv2d_7 (Conv2D)	(None, 5, 5, 64)	36928
batch_normalization_7 (Batch	(None, 5, 5, 64)	256
activation_7 (Activation)	(None, 5, 5, 64)	0
zero_padding2d_9 (ZeroPaddin	(None, 7, 7, 64)	0
conv2d_8 (Conv2D)	(None, 5, 5, 64)	36928
batch_normalization_8 (Batch	(None, 5, 5, 64)	256
activation_8 (Activation)	(None, 5, 5, 64)	0
max_pooling2d_4 (MaxPooling2	(None, 2, 2, 64)	0
dropout_4 (Dropout)	(None, 2, 2, 64)	0
flatten_1 (Flatten)	(None, 256)	0
dense_1 (Dense)	(None, 1024)	263168
dropout_5 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
dropout_6 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 7)	7175

```
Total params: 1,582,151
Trainable params: 1,581,127
Non-trainable params: 1,024
```

訓練過程：　　　　　　　　　　　　平均準確率：0.62816
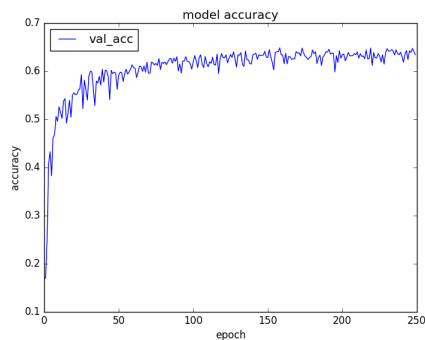
2. **(1%)** 承上題，請用與上述 **CNN** 接近的參數量，實做簡單的 **DNN model**。其模型架構、訓練過程和準確率為何？試與上題結果做比較，並說明你觀察到了什麼？
   (Collaborators: 黃敬庭、倪溥辰**)**
   模型架構：

```
Layer (type)                 Output Shape         Param #
=================================================================
dense_1 (Dense)              (None, 512)          1180160
_____
batch_normalization_1 (Batch (None, 512)          2048
_____
dense_2 (Dense)              (None, 512)          262656
_____
batch_normalization_2 (Batch (None, 512)          2048
_____
dense_3 (Dense)              (None, 7)            3591
=================================================================
Total params: 1,450,503
Trainable params: 1,448,455
Non-trainable params: 2,048
```
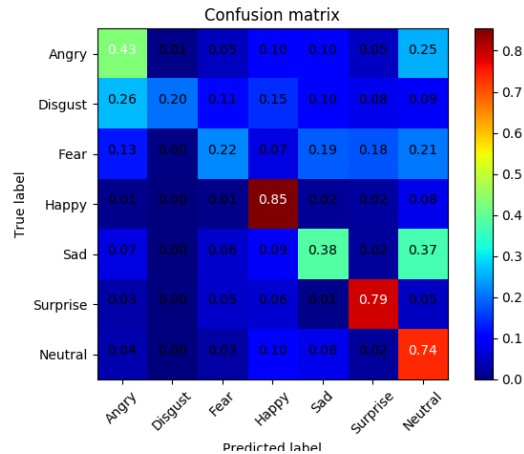
訓練過程：　　　　　　　　　　　　　　　平均準確率：0.34632ls
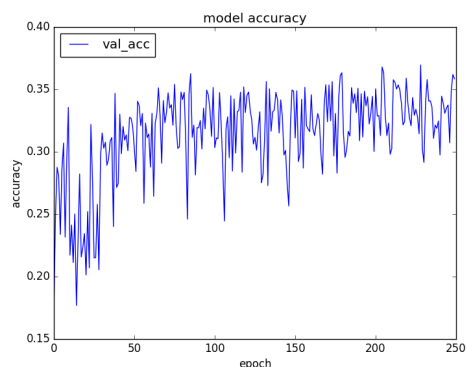
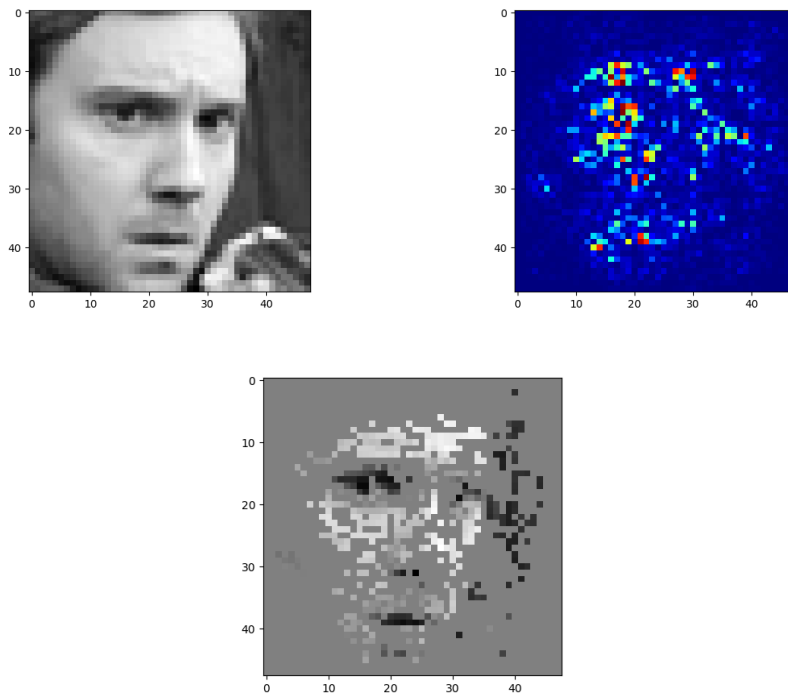觀察：跑完後發現忘記加 drop out，不過還是可以發現在前段的時候 CNN 很快就學到東西，但是 DNN 還是沒什麼進步，可見在圖面辨識部分還是 CNN 表現較好。

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: 黃敬庭、倪溥辰)

由上圖可知，Happy，Surprise，Neutral 的辨識表現得比較好；而 Disgust，Fear，表現最差，大致上來說負面情緒似乎表現較差。

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: 黃敬庭、倪溥辰)

由圖片可知這個 filter 是 focus 在眼睛上下的皮膚變化以及嘴巴附近的特徵。

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:)
   答：