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: None)

答：

【模型架構】

CNN model：
  Conv2D、ZeroPadding、
  BatchNormalization、
  Maxpooling、Dropout交錯組
 成。

DNN：
  3層Dense和最後一層softmax
  組成。

Total Parameters：
  約255萬。

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 48, 48, 256)       2560
zero_padding2d_1 (ZeroPaddin (None, 52, 52, 256)       0
batch_normalization_1 (Batch (None, 52, 52, 256)       1024
max_pooling2d_1 (MaxPooling2 (None, 26, 26, 256)       0
conv2d_2 (Conv2D)            (None, 24, 24, 128)       295040
batch_normalization_2 (Batch (None, 24, 24, 128)       512
zero_padding2d_2 (ZeroPaddin (None, 26, 26, 128)       0
conv2d_3 (Conv2D)            (None, 24, 24, 128)       147584
batch_normalization_3 (Batch (None, 24, 24, 128)       512
max_pooling2d_2 (MaxPooling2 (None, 12, 12, 128)       0
dropout_1 (Dropout)          (None, 12, 12, 128)       0
conv2d_4 (Conv2D)            (None, 10, 10, 128)       147584
batch_normalization_4 (Batch (None, 10, 10, 128)       512
zero_padding2d_3 (ZeroPaddin (None, 12, 12, 128)       0
conv2d_5 (Conv2D)            (None, 10, 10, 128)       147584
batch_normalization_5 (Batch (None, 10, 10, 128)       512
max_pooling2d_3 (MaxPooling2 (None, 5, 5, 128)         0
dropout_2 (Dropout)          (None, 5, 5, 128)         0
flatten_1 (Flatten)          (None, 3200)              0
dense_1 (Dense)              (None, 512)               1638912
batch_normalization_6 (Batch (None, 512)               2048
dropout_3 (Dropout)          (None, 512)               0
dense_2 (Dense)              (None, 256)               131328
batch_normalization_7 (Batch (None, 256)               1024
dropout_4 (Dropout)          (None, 256)               0
dense_3 (Dense)              (None, 128)               32896
batch_normalization_8 (Batch (None, 128)               512
dropout_5 (Dropout)          (None, 128)               0
dense_4 (Dense)              (None, 7)                 903
=================================================================
Total params: 2,551,047
Trainable params: 2,547,719
Non-trainable params: 3,328
```

【訓練過程】 【準確率】
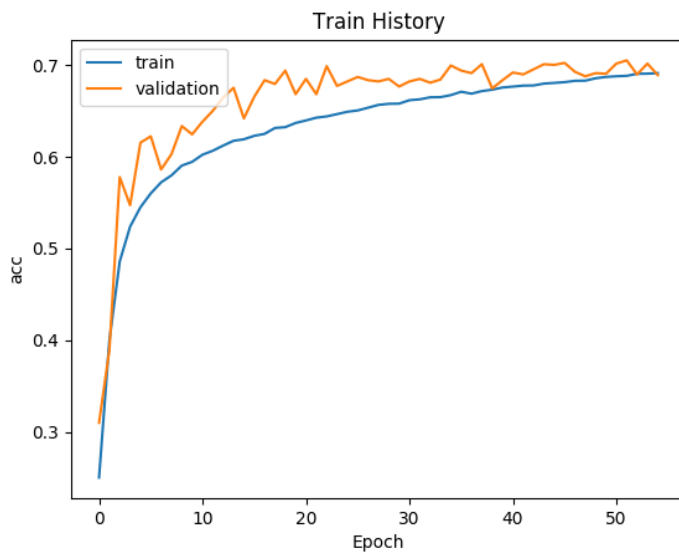
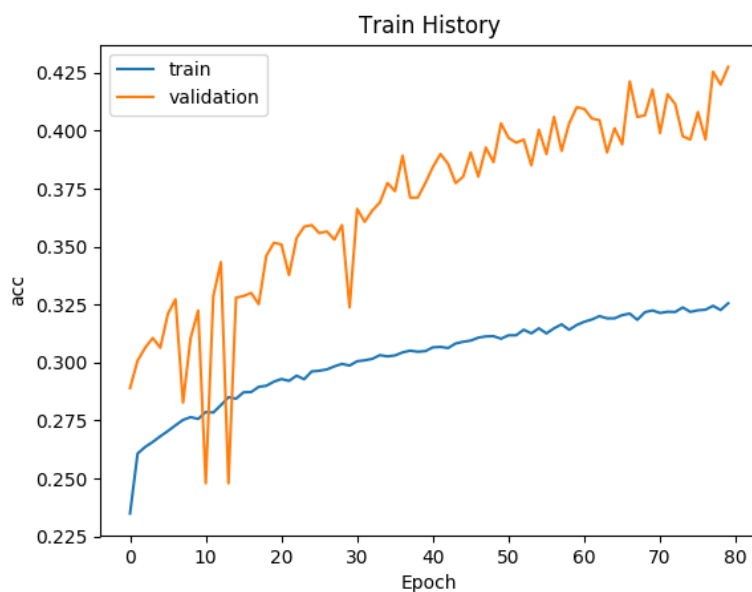|	Accuracy
Private	0.69462
Public	0.68264
Average	0.68863

訓練過程中比較特別的部分是validation accuracy一直都比train accuracy高，可能是validation挑到比較多happy的資料；此外，在第53個epoch有EarlyStopping。

準確率的部分，和同學討論後發現，train單一model達到接近69%的準確率算滿好的，可能是因為model疊很深，而且有用ImageGenerator增加資料量。

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: None)

答：【訓練過程】 【準確率】

	Accuracy
Private	0.38227
Public	0.39314
Average	0.387705

DNN model的結果明顯比CNN model train出來的結果差。同樣的，validation accuracy一直都比train accuracy高。此外，這次沒有EarlyStopping，若增加Epoch數量可能會有更好的結果。

【模型架構】

DNN：

　　3層Dense和最後一層softmax組成

Total Parameters：

　　約263萬，與DNN model的255萬參數量接近。

```
Layer (type)                    Output Shape              Param #
=================================================================
reshape_1 (Reshape)             (None, 2304, 1)           0

flatten_1 (Flatten)             (None, 2304)              0

dense_1 (Dense)                 (None, 512)               1180160

batch_normalization_1 (Batch    (None, 512)               2048

dropout_1 (Dropout)             (None, 512)               0

dense_2 (Dense)                 (None, 1024)              525312

batch_normalization_2 (Batch    (None, 1024)              4096

dropout_2 (Dropout)             (None, 1024)              0

dense_3 (Dense)                 (None, 512)               524800

batch_normalization_3 (Batch    (None, 512)               2048

dropout_3 (Dropout)             (None, 512)               0

dense_4 (Dense)                 (None, 256)               131328

batch_normalization_4 (Batch    (None, 256)               1024

dropout_4 (Dropout)             (None, 256)               0

dense_5 (Dense)                 (None, 512)               131584

batch_normalization_5 (Batch    (None, 512)               2048

dropout_5 (Dropout)             (None, 512)               0

dense_6 (Dense)                 (None, 256)               131328

batch_normalization_6 (Batch    (None, 256)               1024

dropout_6 (Dropout)             (None, 256)               0

dense_7 (Dense)                 (None, 7)                 1799
=================================================================
Total params: 2,638,599
Trainable params: 2,632,455
Non-trainable params: 6,144
```

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: None)

答：
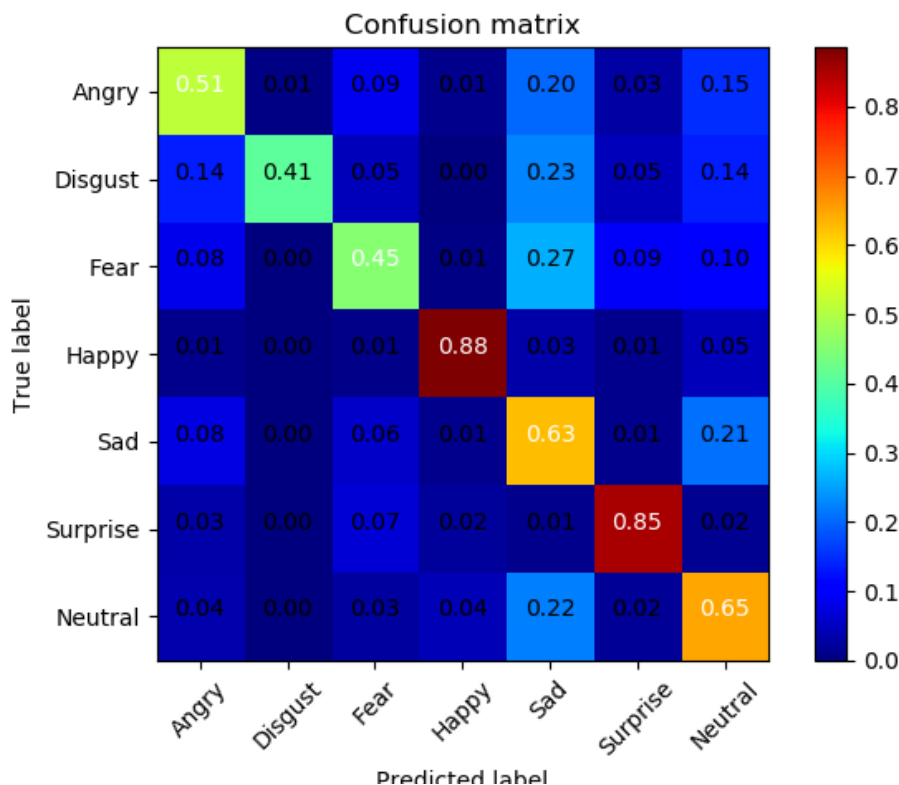
數據顯示，判斷'Happy'和'Surprise'兩種情緒的結果最準確，'Sad'和'Neutral'次之，而'Angry'、'Disgust'和'Fear'則最不準確。
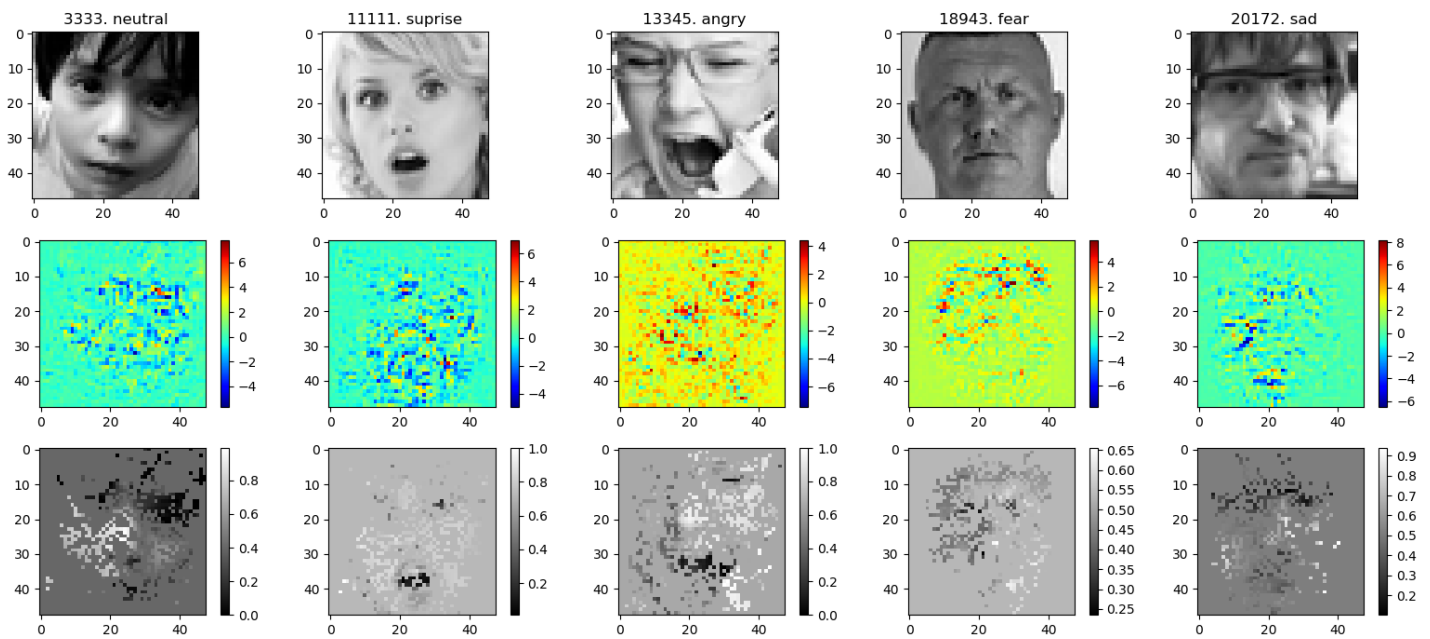
其中，'Angry'、'Disgust'和'Fear'都容易被誤判為'Sad'和'Neutral'，而其中誤判為'Sad'的機率比較高；同時，'Sad'容易被判斷為'Neutral'，'Neutral'也容易被判斷為'Sad'。
推測因為負面情緒牽動的臉部肌肉類似，機器容易混淆，又'Sad'和'Neutral'常常只有一線之隔，可能是微微皺眉、眼神向下等等，因此被誤判為彼此的機率較高。

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: None)

答：

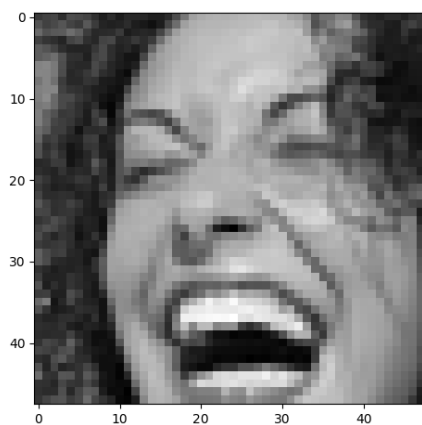Figures selected: #3333、#11111、#13345、#18943、#20172

由圖中可得知，做classification時主要focus在五官，其中又以眼睛、眉毛、嘴巴最為重要。頭髮和四周圍則相對不重要。
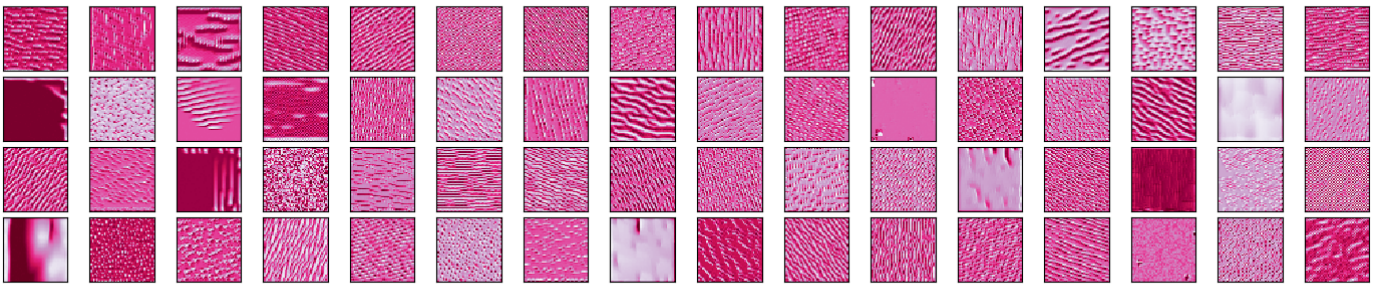
5. (1%) 承(1)(2)，利用上課所提到的 gradient ascent 方法，觀察特定層的filter最容易被哪種圖片 activate。

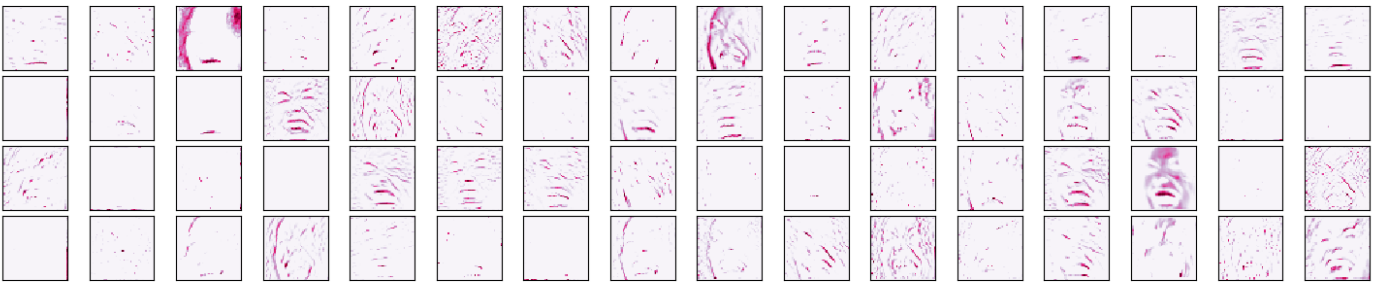   (Collaborators: b04901060黃文璁)

   答：Figure Selected : #6543

Filters of the layer

Outputs of the layer

從兩張圖片中可觀察到，的確有特定的臉部輪廓和filter的圖樣類似。