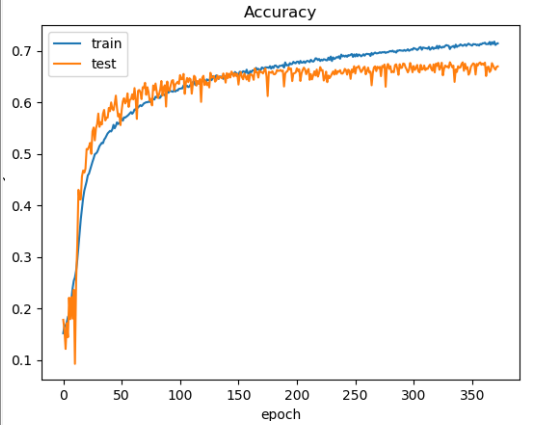
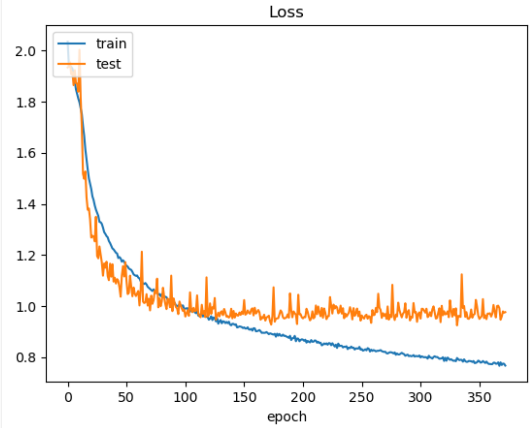
學號：R06942048 系級： 電信碩一 姓名：林彥伯

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, 48, 48, 64)	1664
leaky_re_lu_1 (LeakyReLU)	(None, 48, 48, 64)	0
batch_normalization_1 (Batch	(None, 48, 48, 64)	256
max_pooling2d_1 (MaxPooling2	(None, 24, 24, 64)	0
dropout_1 (Dropout)	(None, 24, 24, 64)	0
conv2d_2 (Conv2D)	(None, 24, 24, 128)	73856
leaky_re_lu_2 (LeakyReLU)	(None, 24, 24, 128)	0
batch_normalization_2 (Batch	(None, 24, 24, 128)	512
max_pooling2d_2 (MaxPooling2	(None, 12, 12, 128)	0
dropout_2 (Dropout)	(None, 12, 12, 128)	0
conv2d_3 (Conv2D)	(None, 12, 12, 512)	590336
leaky_re_lu_3 (LeakyReLU)	(None, 12, 12, 512)	0
batch_normalization_3 (Batch	(None, 12, 12, 512)	2048
max_pooling2d_3 (MaxPooling2	(None, 6, 6, 512)	0
dropout_3 (Dropout)	(None, 6, 6, 512)	0
conv2d_4 (Conv2D)	(None, 6, 6, 512)	2359808
leaky_re_lu_4 (LeakyReLU)	(None, 6, 6, 512)	0
batch_normalization_4 (Batch	(None, 6, 6, 512)	2048
average_pooling2d_1 (Average	(None, 3, 3, 512)	0
dropout_4 (Dropout)	(None, 3, 3, 512)	0
flatten_1 (Flatten)	(None, 4608)	0
dense_1 (Dense)	(None, 1024)	4719616
batch_normalization_5 (Batch	(None, 1024)	4096
dropout_5 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 256)	262400
batch_normalization_6 (Batch	(None, 256)	1024
dropout_6 (Dropout)	(None, 256)	0
dense_3 (Dense)	(None, 7)	1799

Total params: 8,019,463
Trainable params: 8,014,471
Non-trainable params: 4,992





本次 model 不是特別深，主要方式是先進行 data argument，因為 data 分布非常 imbalance，故有隨機 sample 一些類別的資料(同樣的資料)，接下來送入 image generator(會自己翻轉、cropping、rotation)，kaggle 上的分數為:70%左右，但因為要 train 的時間滿久的(當時沒有存下來)，故圖片的 loss 為 early stopping 的結果。

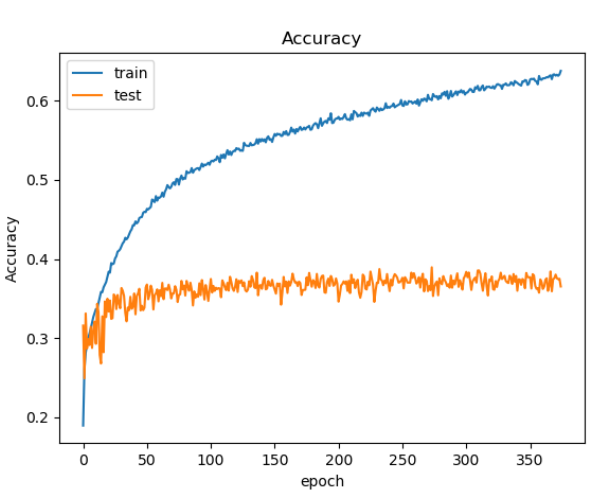
2. (1%) 承上題，請用與上述 CNN 接近的參數量，實做簡單的 DNN model。其模型架構、訓練過程和準確率為何？試與上題結果做比較，並說明你觀察到了

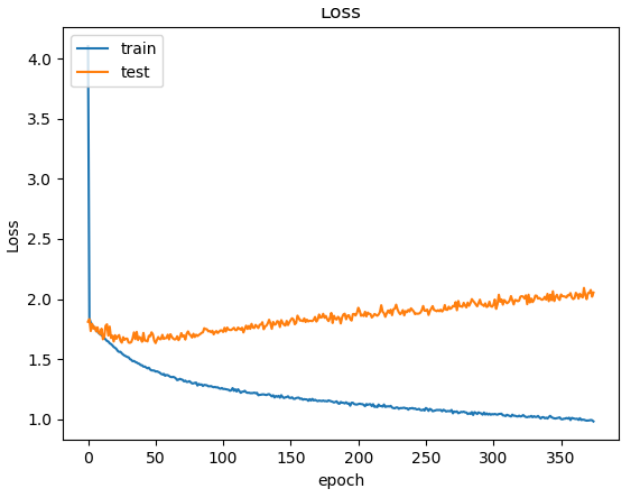
什麼？

(Collaborators:)

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 48, 48, 1)	0
flatten_1 (Flatten)	(None, 2304)	0
dense_1 (Dense)	(None, 4096)	9441280
dense_3 (Dense)	(None, 1024)	4195328
dense_4 (Dense)	(None, 7)	7175
activation_1 (Activation)	(None, 7)	0

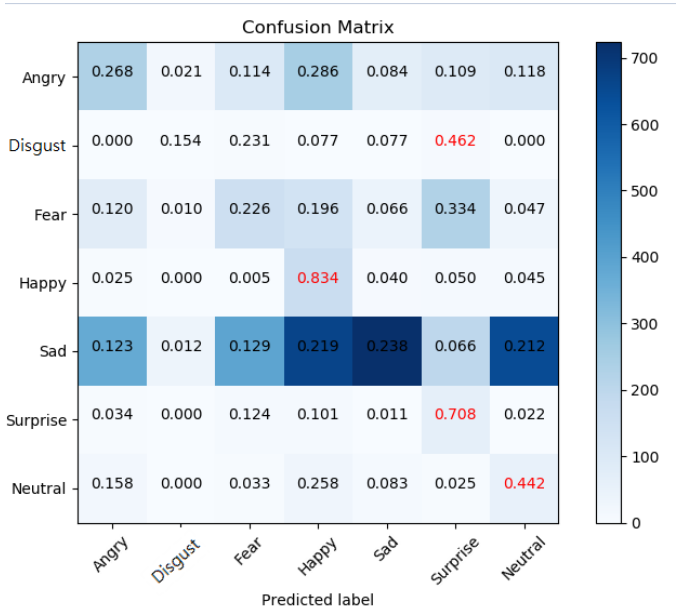
Total params: 13,643,783
Trainable params: 13,643,783
Non-trainable params: 0





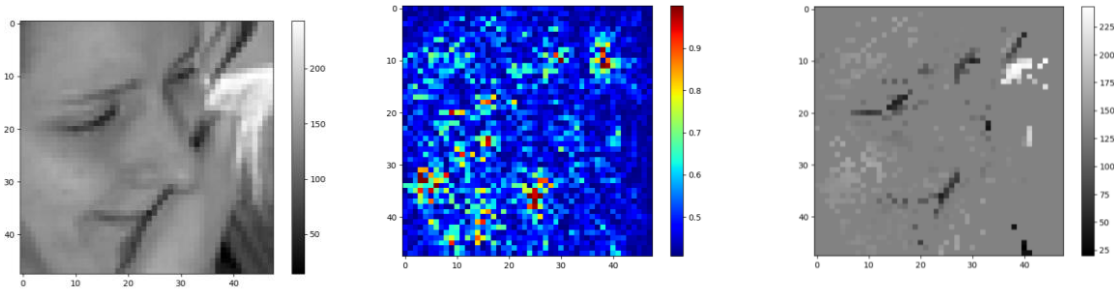
可以看到 DNN 雖然參數比 CNN model 還來得多，但效果不盡理想，因為沒有 convolution 與 pooling 導致網路不會針對特定重點去學習，所以效果很差。大概只有 30 多%

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]



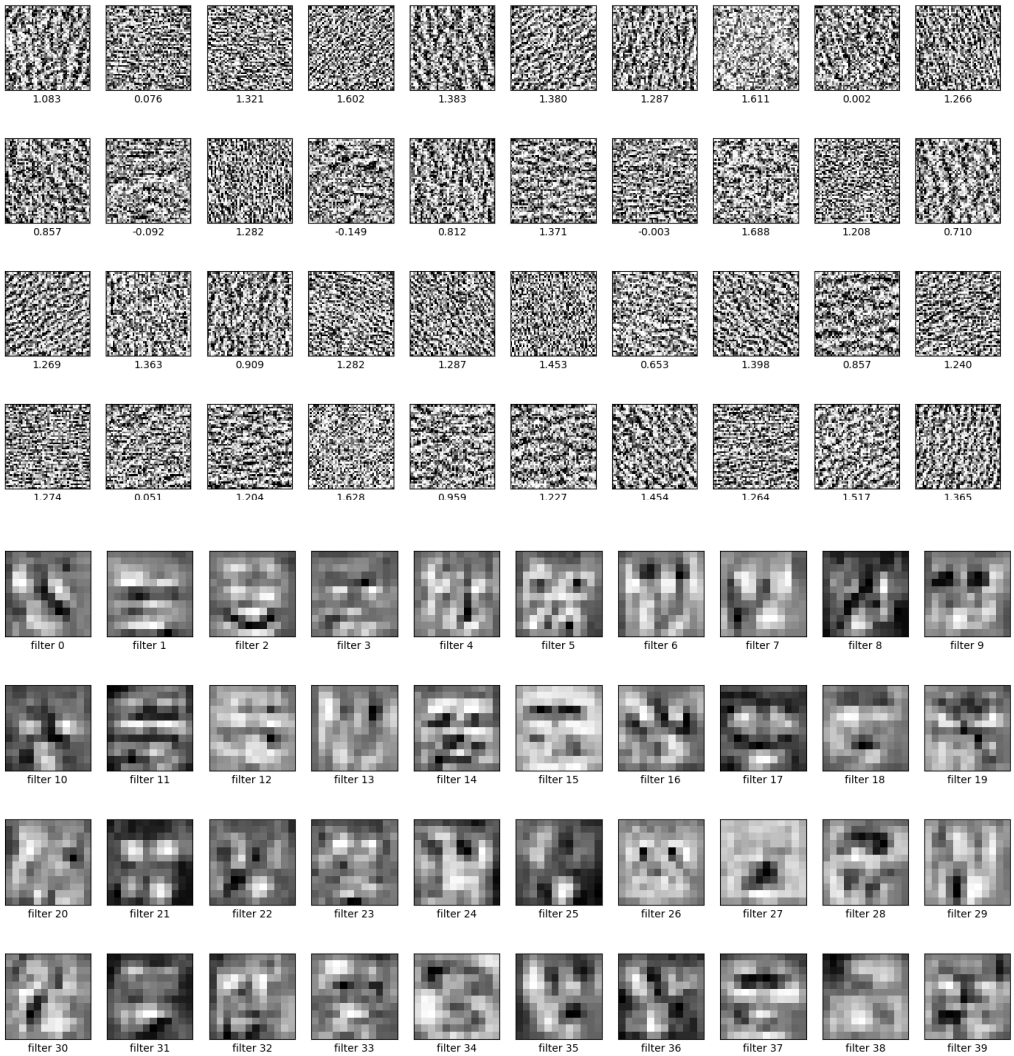
可以發現到非對角線的，然後顏色又較深的(預測為: happy 或 netural 實際卻是: Sad) 然後有一點異常的高: 預測是 Surprise 但 label 是: Disgust，原因是因為 Disgust 異常少數(大約只有 400 多張)導致網路學不太起來。

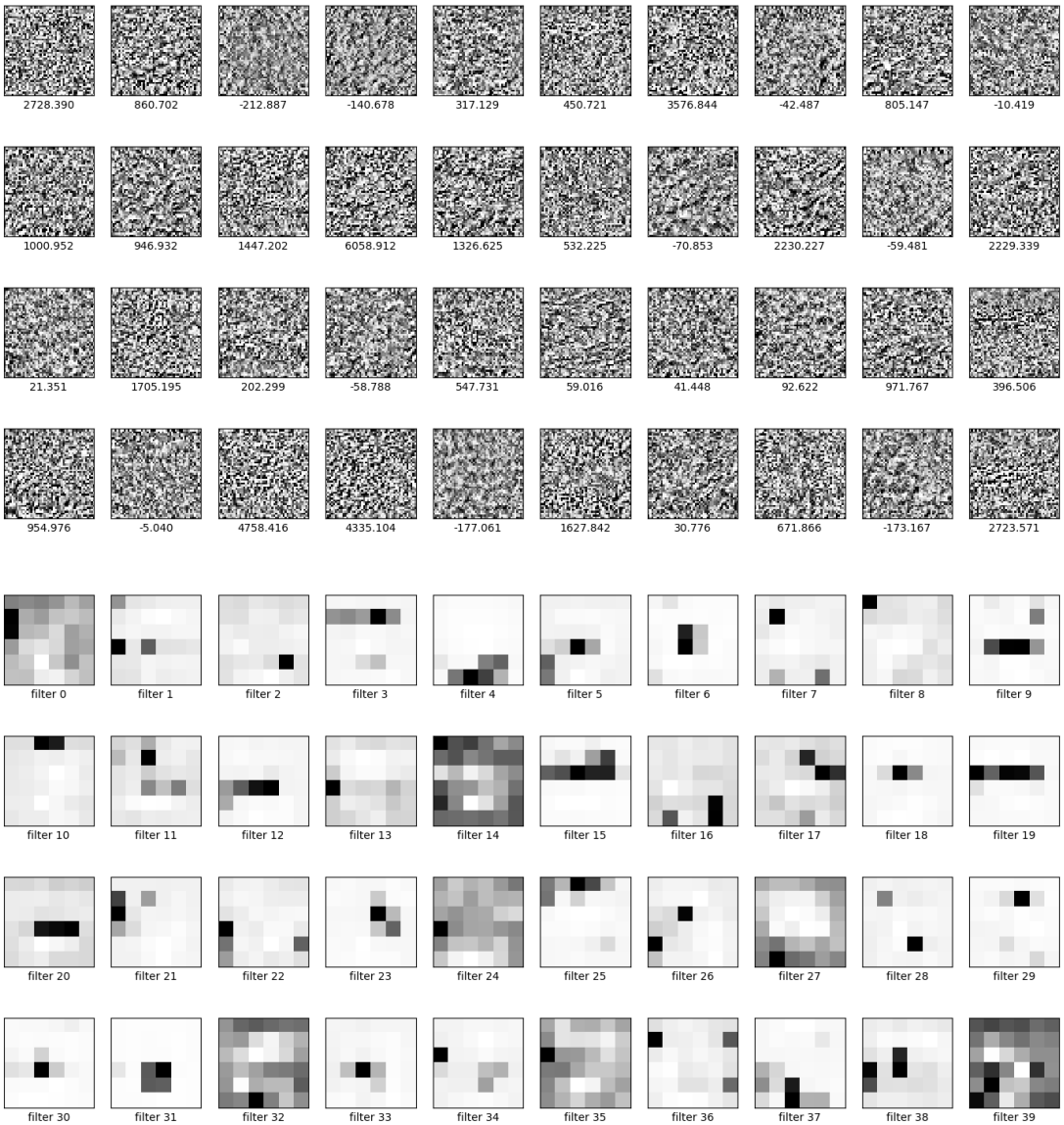
4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？



可以發現到主要注重於五官的部分，filter 對於五官的特徵會比較注重。

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。





前兩張是較前面的 layer，基本上是一坨看不太清楚，不過可以順觀察到應該是臉型的部分。後面兩張則是較後面的 layer，發現到只有特定一些地方有黑點，而且位於圖片中間較多，應該是五官的部分，說明到後面的 layer 可以過濾到五官當作重要的 feature。