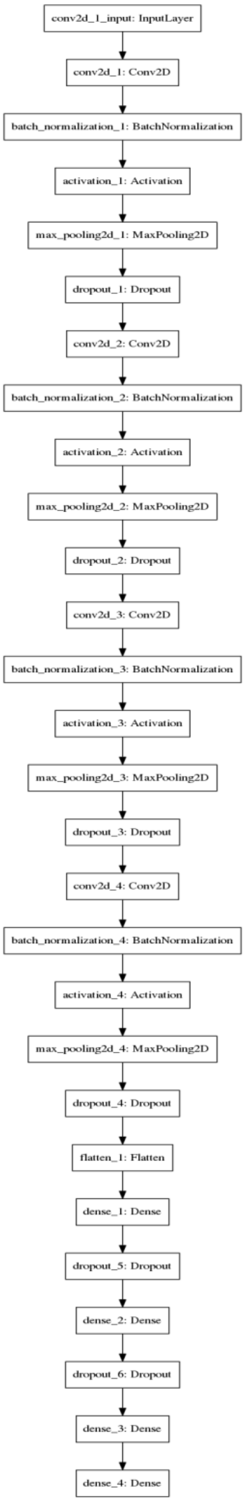
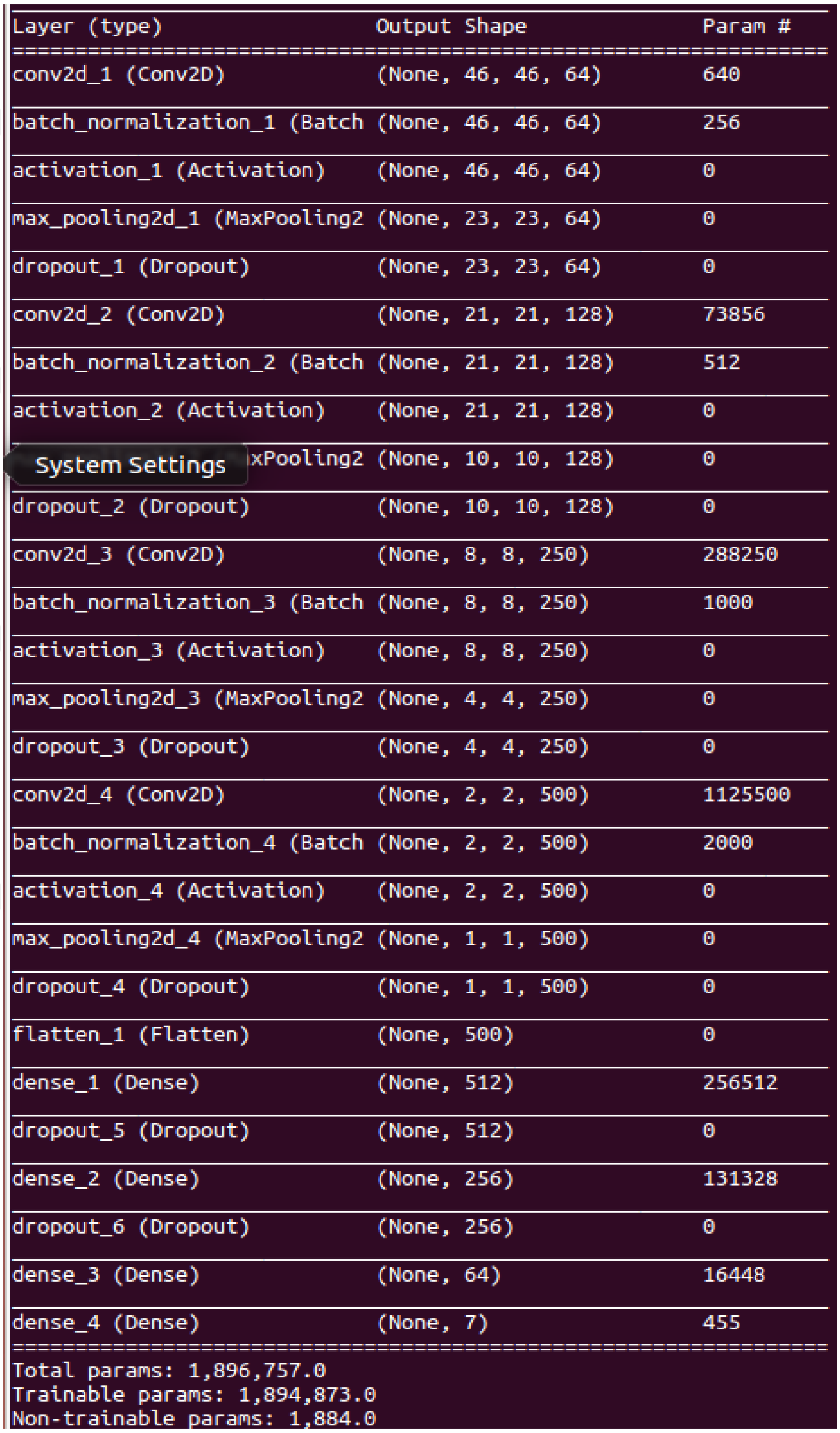
學號：R05921087　系級：　電機碩一　姓名：鍾智堅

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：



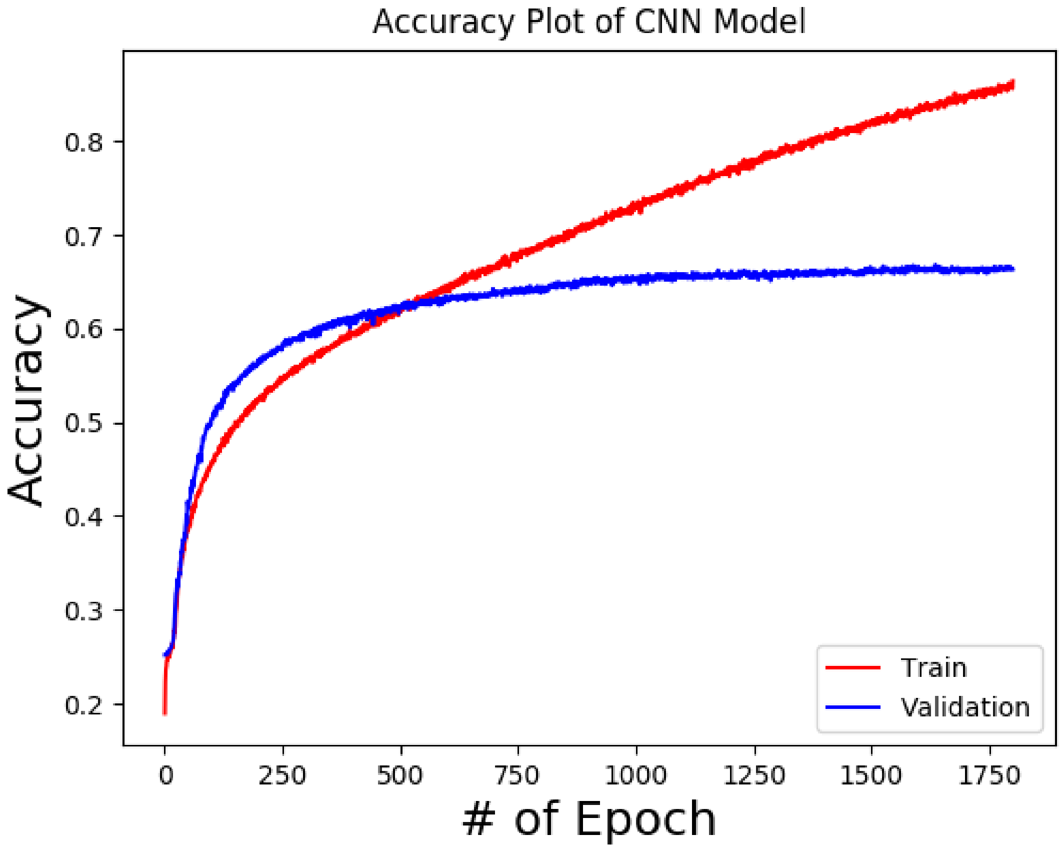


```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 46, 46, 64)        640
_________________________________________________________________
batch_normalization_1 (Batch (None, 46, 46, 64)        256
_________________________________________________________________
activation_1 (Activation)    (None, 46, 46, 64)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 23, 23, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 23, 23, 64)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 21, 21, 128)       73856
_________________________________________________________________
batch_normalization_2 (Batch (None, 21, 21, 128)       512
_________________________________________________________________
activation_2 (Activation)    (None, 21, 21, 128)       0
_________________________________________________________________
 System Settings     xPooling2 (None, 10, 10, 128)       0
_________________________________________________________________
dropout_2 (Dropout)          (None, 10, 10, 128)       0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 8, 8, 250)         288250
_________________________________________________________________
batch_normalization_3 (Batch (None, 8, 8, 250)         1000
_________________________________________________________________
activation_3 (Activation)    (None, 8, 8, 250)         0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 4, 4, 250)         0
_________________________________________________________________
dropout_3 (Dropout)          (None, 4, 4, 250)         0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 2, 2, 500)         1125500
_________________________________________________________________
batch_normalization_4 (Batch (None, 2, 2, 500)         2000
_________________________________________________________________
activation_4 (Activation)    (None, 2, 2, 500)         0
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 1, 1, 500)         0
_________________________________________________________________
dropout_4 (Dropout)          (None, 1, 1, 500)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 500)               0
_________________________________________________________________
dense_1 (Dense)              (None, 512)               256512
_________________________________________________________________
dropout_5 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_2 (Dense)              (None, 256)               131328
_________________________________________________________________
dropout_6 (Dropout)          (None, 256)               0
_________________________________________________________________
dense_3 (Dense)              (None, 64)                16448
_________________________________________________________________
dense_4 (Dense)              (None, 7)                 455
=================================================================
Total params: 1,896,757.0
Trainable params: 1,894,873.0
Non-trainable params: 1,884.0
```

訓練是使用 Nadam 作為 optimizer，然後 epoch 為 1800，learning rate 為 0.00001，schedule_decay 為 0.001。

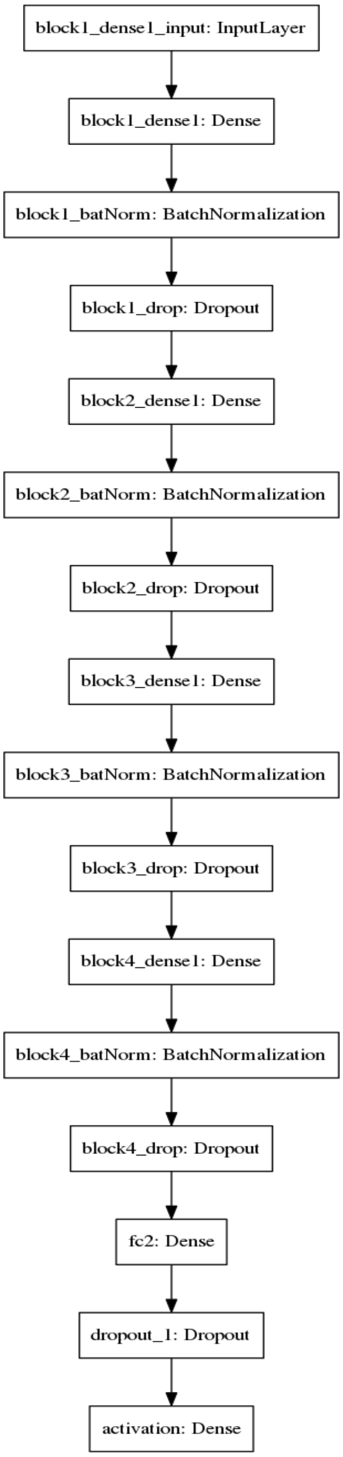
Training data 為 80%，Validation data 為 20%。

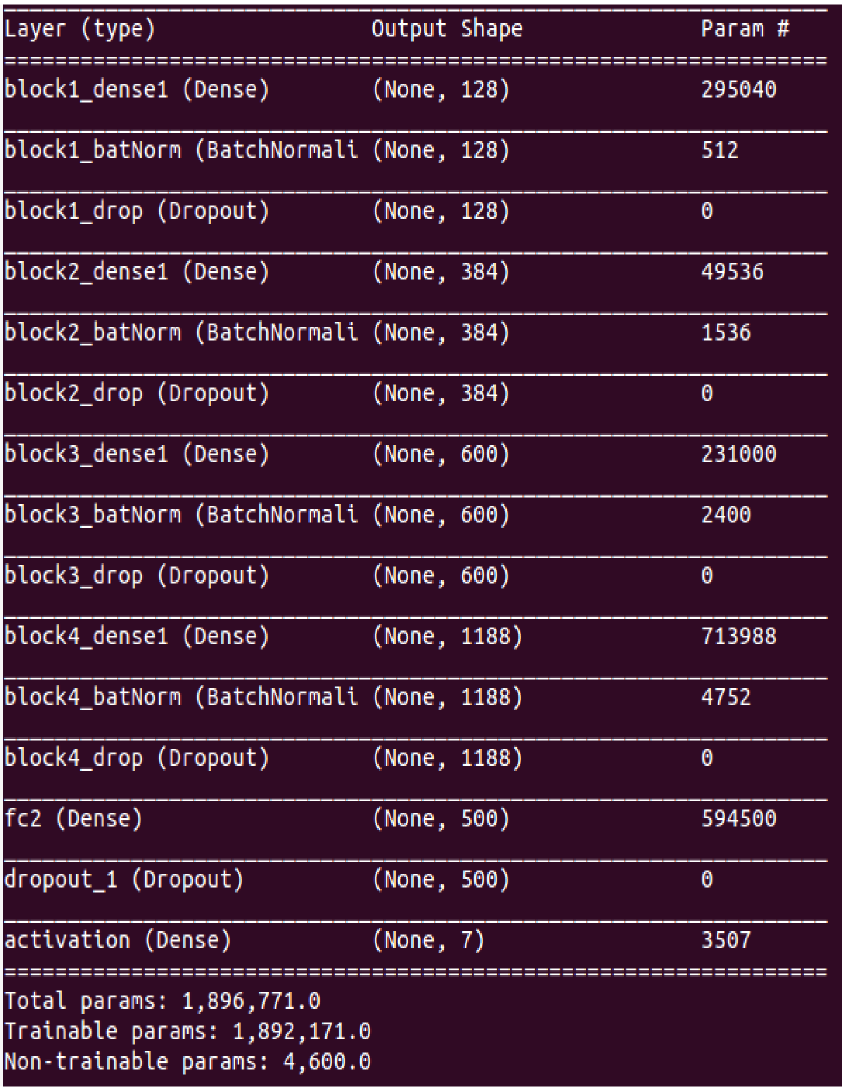
Kaggle 分數為 0.65868。



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？

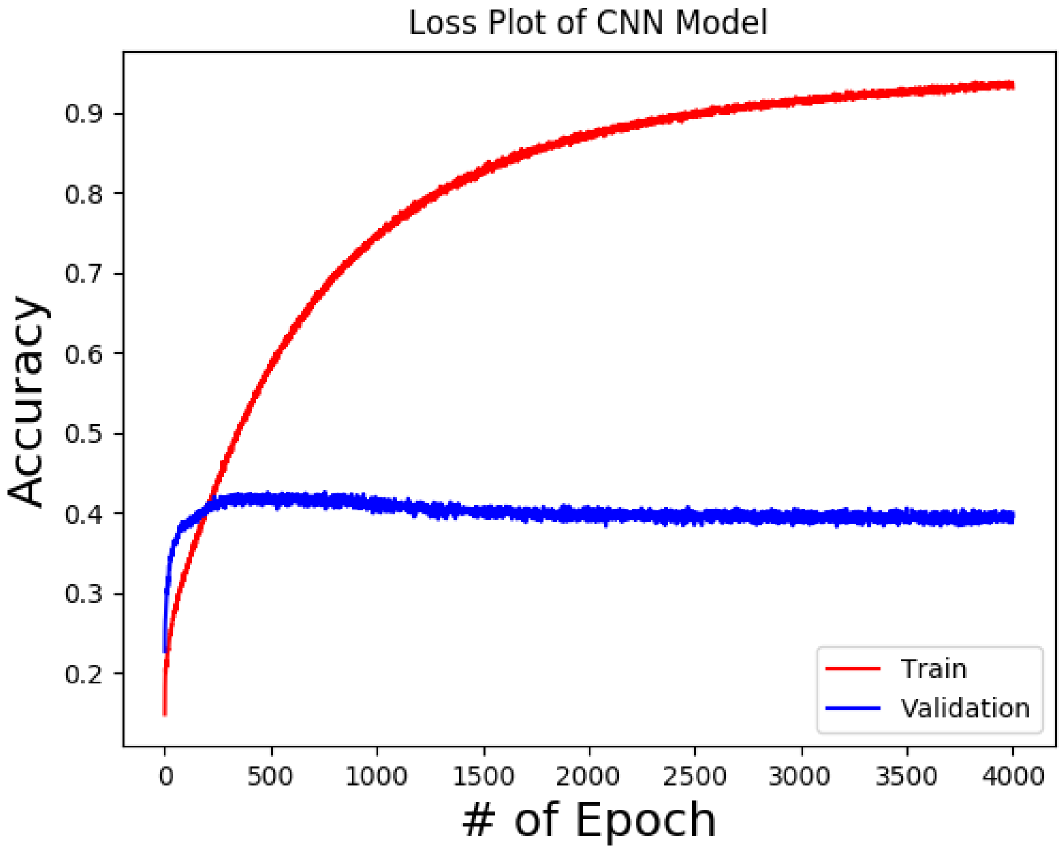
答：





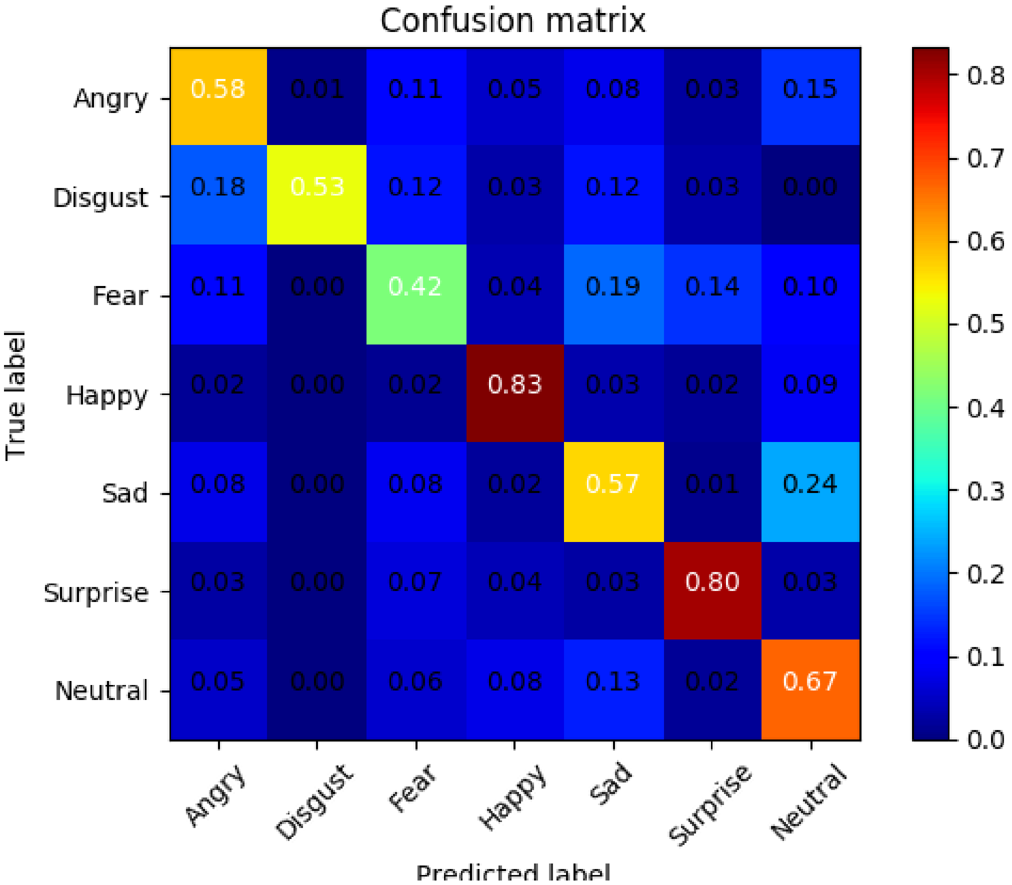
```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
block1_dense1 (Dense)        (None, 128)               295040
_________________________________________________________________
block1_batNorm (BatchNormali (None, 128)               512
_________________________________________________________________
block1_drop (Dropout)        (None, 128)               0
_________________________________________________________________
block2_dense1 (Dense)        (None, 384)               49536
_________________________________________________________________
block2_batNorm (BatchNormali (None, 384)               1536
_________________________________________________________________
block2_drop (Dropout)        (None, 384)               0
_________________________________________________________________
block3_dense1 (Dense)        (None, 600)               231000
_________________________________________________________________
block3_batNorm (BatchNormali (None, 600)               2400
_________________________________________________________________
block3_drop (Dropout)        (None, 600)               0
_________________________________________________________________
block4_dense1 (Dense)        (None, 1188)              713988
_________________________________________________________________
block4_batNorm (BatchNormali (None, 1188)              4752
_________________________________________________________________
block4_drop (Dropout)        (None, 1188)              0
_________________________________________________________________
fc2 (Dense)                  (None, 500)               594500
_________________________________________________________________
dropout_1 (Dropout)          (None, 500)               0
_________________________________________________________________
activation (Dense)           (None, 7)                 3507
=================================================================
Total params: 1,896,771.0
Trainable params: 1,892,171.0
Non-trainable params: 4,600.0
```

訓練是使用 Nadam 作為 optimizer，然後 epoch 為 4000，learning rate 為 0.00001，schedule_decay 為 0.001。
Training data 為 80%，Validation data 為 20%。
Kaggle’s score 是 0.39593。
DNN 在開始學習非常快，因此參數調整比較大，因此很容易到達瓶頸，相對上比起 CNN，精準度下降許多。



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]
   答：

從 confusion matrix 圖像中，可以看出 Fear 比較難以被辨認，只有 42%成功被辨識。相反，Happy 和 Surprise 是辨識率最高的，成功率分別為 83%和 80%。

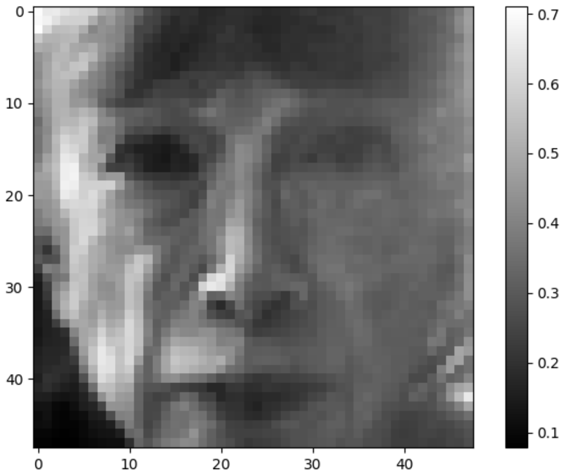


	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	0.58	0.01	0.11	0.05	0.08	0.03	0.15
Disgust	0.18	0.53	0.12	0.03	0.12	0.03	0.00
Fear	0.11	0.00	0.42	0.04	0.19	0.14	0.10
Happy	0.02	0.00	0.02	0.83	0.03	0.02	0.09
Sad	0.08	0.00	0.08	0.02	0.57	0.01	0.24
Surprise	0.03	0.00	0.07	0.04	0.03	0.80	0.03
Neutral	0.05	0.00	0.06	0.08	0.13	0.02	0.67

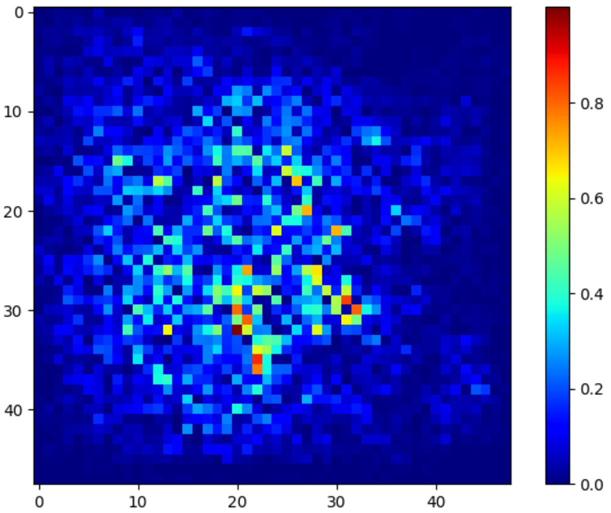
4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？
答：

透過 heatmap，可以看出 focus 的部分是在鼻子附近和眼睛，尤其是表情做出來后有皺紋的部分也是觀察重點之一。

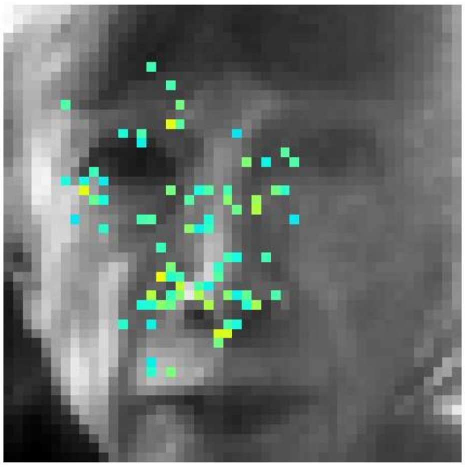
原圖



Saliency Map



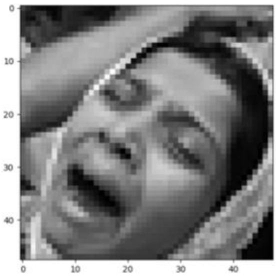
Heat 的重點部分

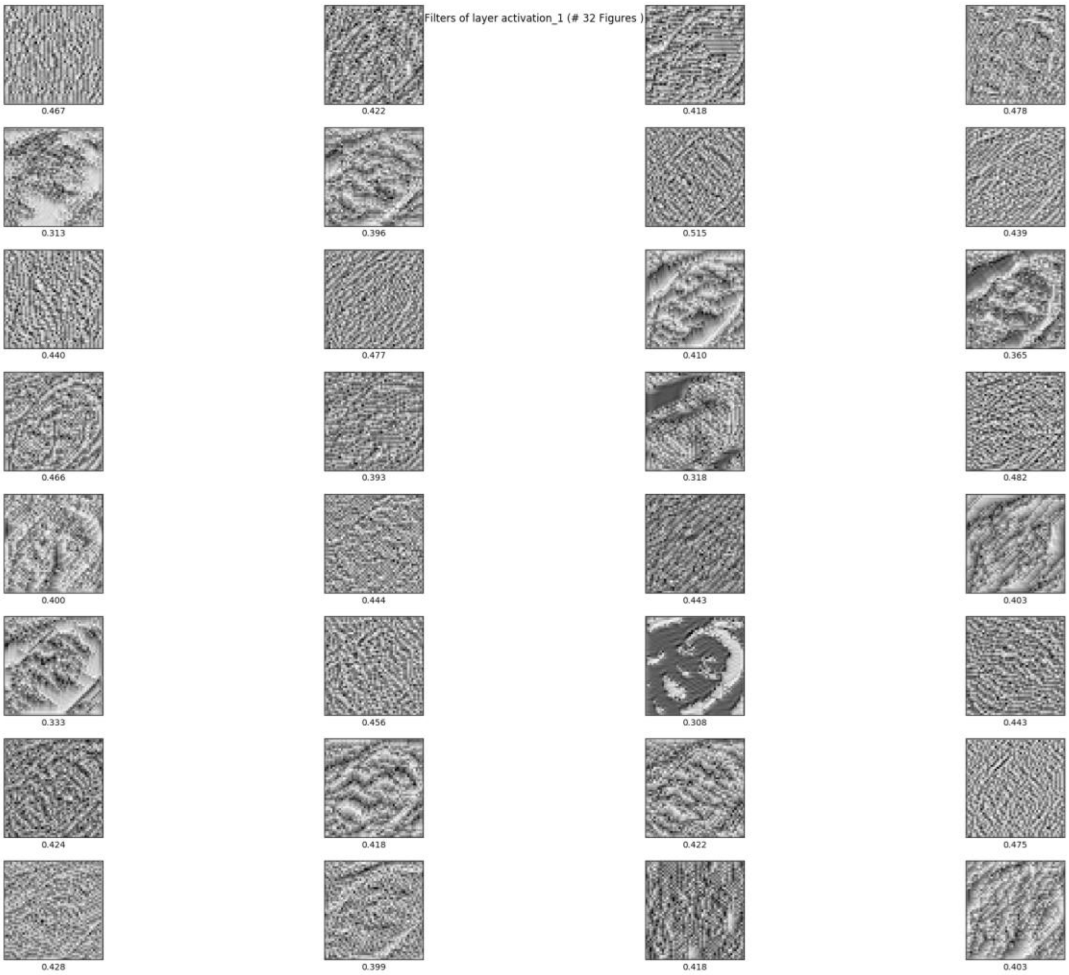


5. (1%) 承(1)(2)，利用上課所提到的 gradient ascent 方法，觀察特定層的 filter 最容易被哪種圖片 activate。

答：

觀察 batch_normalization_2，即第二 block 的 batch normalization，通過觀察 loss，發現 loss 越低的圖片，越容易被這個圖片 activate。Row6 的 column3 是擁有最低 loss，即 0.308





[Bonus]（1%）從 training data 中移除部份 label，實做 semi-supervised learning

[Bonus]（1%）在 Problem 5 中，提供了 3 個 hint，可以嘗試實作及觀察（但也可以不限於 hint 所提到的方向，也可以自己去研究更多關於 CNN 細節的資料)，並說明你做了些什麼？ [完成 1 個: +0.4%, 完成 2 個: +0.7%, 完成 3 個: +1%]

答：

在程式里我把 training data 分成 train 和 validation 兩個部分，再把 test data 作為 unlabel data 做 semi-supervised learning。