學號： R06945003 系級： 生醫電資碩一　姓名：林鈺盛

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?

CNN model 架構:

```
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 32, 48, 48)        320
_________________________________________________________________
batch_normalization_1 (Batch (None, 32, 48, 48)        128
_________________________________________________________________
activation_1 (Activation)    (None, 32, 48, 48)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 32, 48, 48)        9248
_________________________________________________________________
batch_normalization_2 (Batch (None, 32, 48, 48)        128
_________________________________________________________________
activation_2 (Activation)    (None, 32, 48, 48)        0
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 32, 48, 48)        9248
_________________________________________________________________
batch_normalization_3 (Batch (None, 32, 48, 48)        128
_________________________________________________________________
activation_3 (Activation)    (None, 32, 48, 48)        0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 16, 24, 48)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 64, 24, 48)        9280
_________________________________________________________________
batch_normalization_4 (Batch (None, 64, 24, 48)        256
_________________________________________________________________
activation_4 (Activation)    (None, 64, 24, 48)        0
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 64, 24, 48)        36928
_________________________________________________________________
batch_normalization_5 (Batch (None, 64, 24, 48)        256
_________________________________________________________________
activation_5 (Activation)    (None, 64, 24, 48)        0
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 64, 24, 48)        36928
_________________________________________________________________
batch_normalization_6 (Batch (None, 64, 24, 48)        256
_________________________________________________________________
activation_6 (Activation)    (None, 64, 24, 48)        0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 32, 12, 48)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 128, 12, 48)       36992
_________________________________________________________________
batch_normalization_7 (Batch (None, 128, 12, 48)       512
_________________________________________________________________
activation_7 (Activation)    (None, 128, 12, 48)       0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 128, 12, 48)       147584
_________________________________________________________________
batch_normalization_8 (Batch (None, 128, 12, 48)       512
_________________________________________________________________
activation_8 (Activation)    (None, 128, 12, 48)       0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 128, 12, 48)       147584
_________________________________________________________________
batch_normalization_9 (Batch (None, 128, 12, 48)       512
_________________________________________________________________
activation_9 (Activation)    (None, 128, 12, 48)       0
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 64, 6, 48)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 18432)             0
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              18875392
_________________________________________________________________
batch_normalization_10 (Batc (None, 1024)              4096
_________________________________________________________________
activation_10 (Activation)   (None, 1024)              0
_________________________________________________________________
dropout_1 (Dropout)          (None, 1024)              0
_________________________________________________________________
dense_2 (Dense)              (None, 7)                 7175
=================================================================
Total params: 19,323,463
Trainable params: 19,320,071
Non-trainable params: 3,392
```

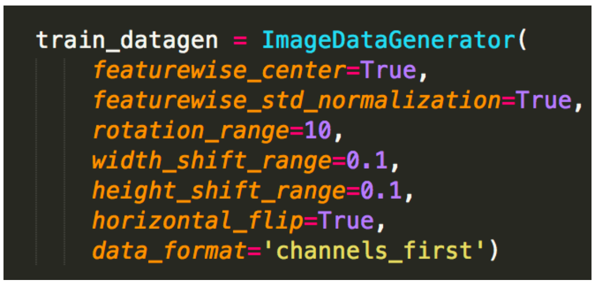
訓練過程:

Epochs:100

Batch size:128

Optimizer: Adam

Augmentation:

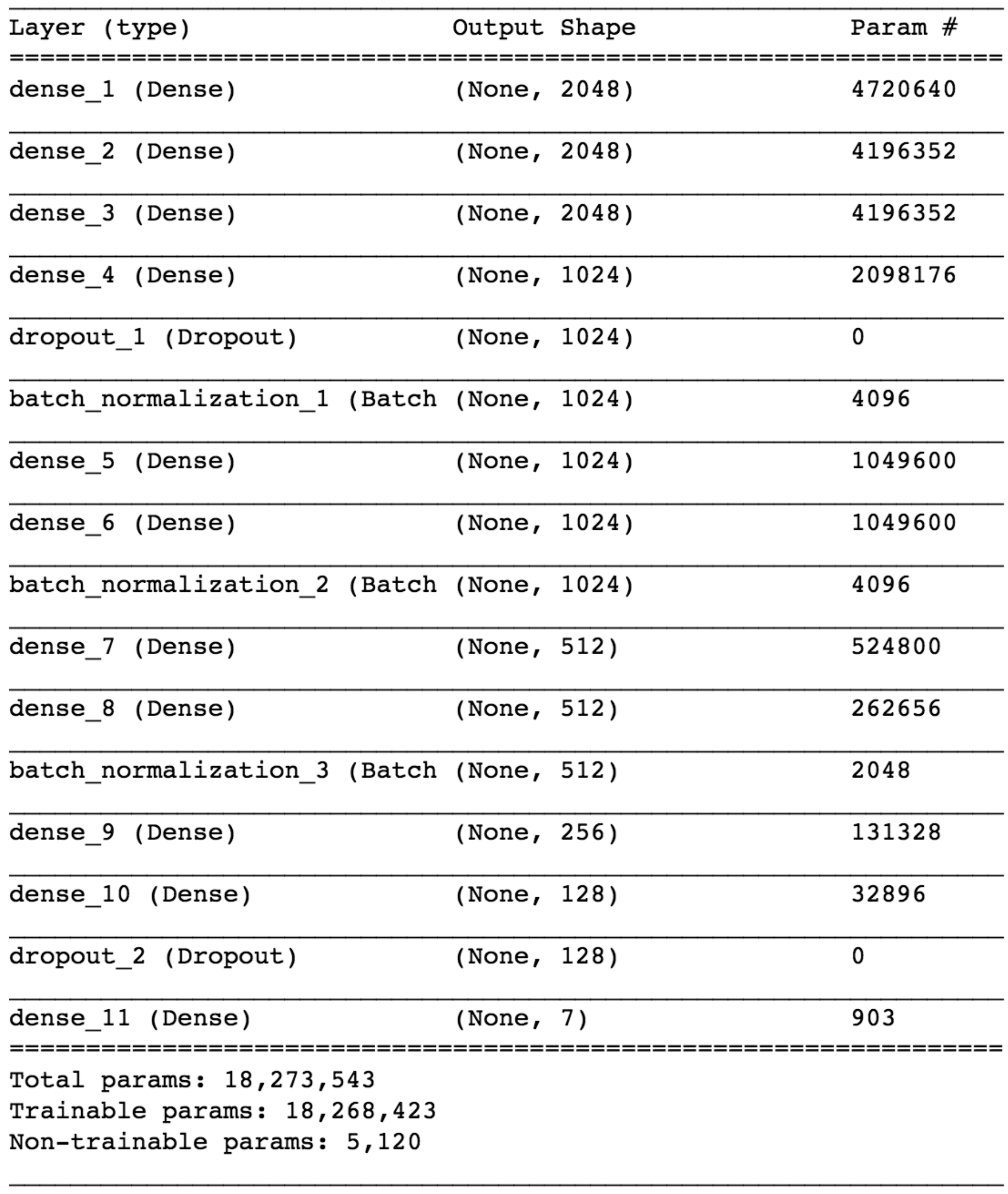


```
train_datagen = ImageDataGenerator(
    featurewise_center=True,
    featurewise_std_normalization=True,
    rotation_range=10,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True,
    data_format='channels_first')
```

我的 CNN model 是用全部的 training data 來訓練，沒有切 validation，因為當時覺得 val_acc 都沒有好的 performance，test data 在 kaggle 上得到的準確率 0.65。

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？

DNN model 架構：



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
dense_1 (Dense)              (None, 2048)              4720640   
_________________________________________________________________
dense_2 (Dense)              (None, 2048)              4196352   
_________________________________________________________________
dense_3 (Dense)              (None, 2048)              4196352   
_________________________________________________________________
dense_4 (Dense)              (None, 1024)              2098176   
_________________________________________________________________
dropout_1 (Dropout)          (None, 1024)              0         
_________________________________________________________________
batch_normalization_1 (Batch (None, 1024)              4096      
_________________________________________________________________
dense_5 (Dense)              (None, 1024)              1049600   
_________________________________________________________________
dense_6 (Dense)              (None, 1024)              1049600   
_________________________________________________________________
batch_normalization_2 (Batch (None, 1024)              4096      
_________________________________________________________________
dense_7 (Dense)              (None, 512)               524800    
_________________________________________________________________
dense_8 (Dense)              (None, 512)               262656    
_________________________________________________________________
batch_normalization_3 (Batch (None, 512)               2048      
_________________________________________________________________
dense_9 (Dense)              (None, 256)               131328    
_________________________________________________________________
dense_10 (Dense)             (None, 128)               32896     
_________________________________________________________________
dropout_2 (Dropout)          (None, 128)               0         
_________________________________________________________________
dense_11 (Dense)             (None, 7)                 903       
=================================================================
Total params: 18,273,543
Trainable params: 18,268,423
Non-trainable params: 5,120
_________________________________________________________________
```

訓練過程：

Epochs：80

Batch size：128

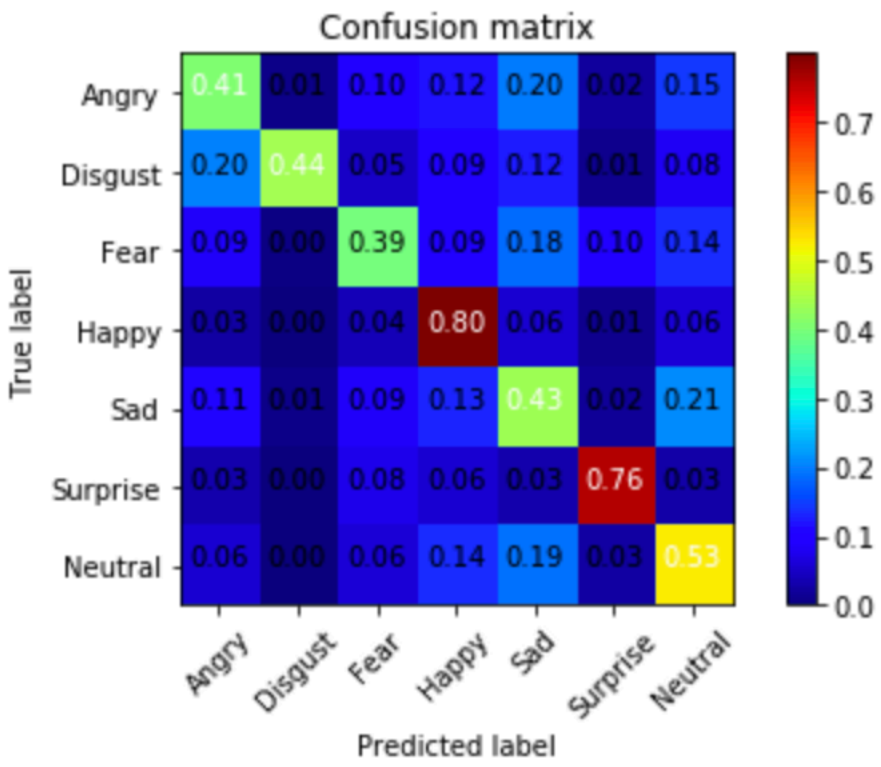
Validation set：20% training data

Optimizer：Adam

Accuracy：val_acc：0.33　kaggle：032

很明顯可以觀察到用一般的 DNN 對於影像辨識的準確率非常差，相較於 CNN

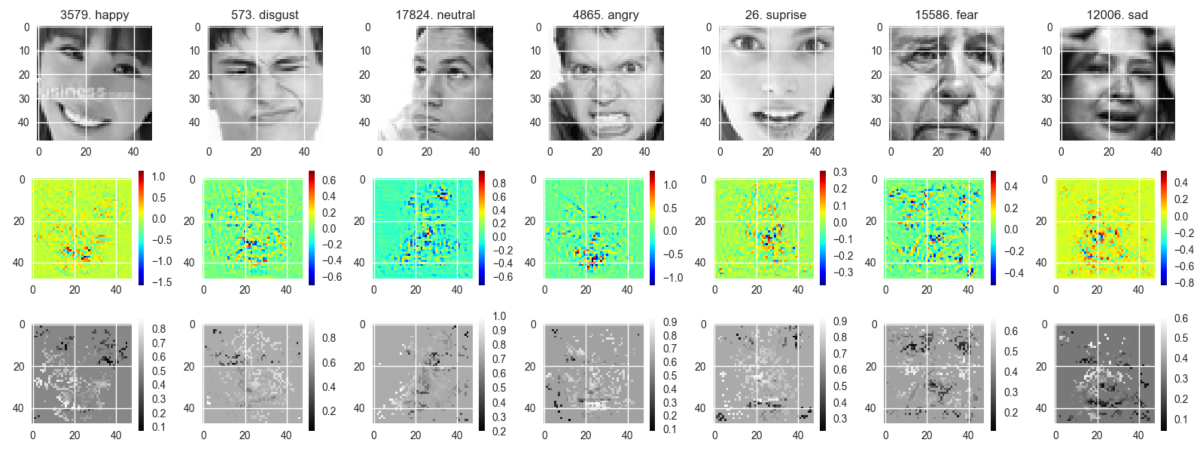
3.(1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析]



True \ Predicted	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	0.41	0.01	0.10	0.12	0.20	0.02	0.15
Disgust	0.20	0.44	0.05	0.09	0.12	0.01	0.08
Fear	0.09	0.00	0.39	0.09	0.18	0.10	0.14
Happy	0.03	0.00	0.04	0.80	0.06	0.01	0.06
Sad	0.11	0.01	0.09	0.13	0.43	0.02	0.21
Surprise	0.03	0.00	0.08	0.06	0.03	0.76	0.03
Neutral	0.06	0.00	0.06	0.14	0.19	0.03	0.53

Happy 以及 Surprise 的準確率最高，其餘容易搞混的有 Angry-Sad、Disgust-Angry、Fear-Sad、Sad-Neutral。(ps.因為我較好的 model 並沒有切 validation，所以用了一個約 55% accuracy 的 model 來 predict validation(20% training data))

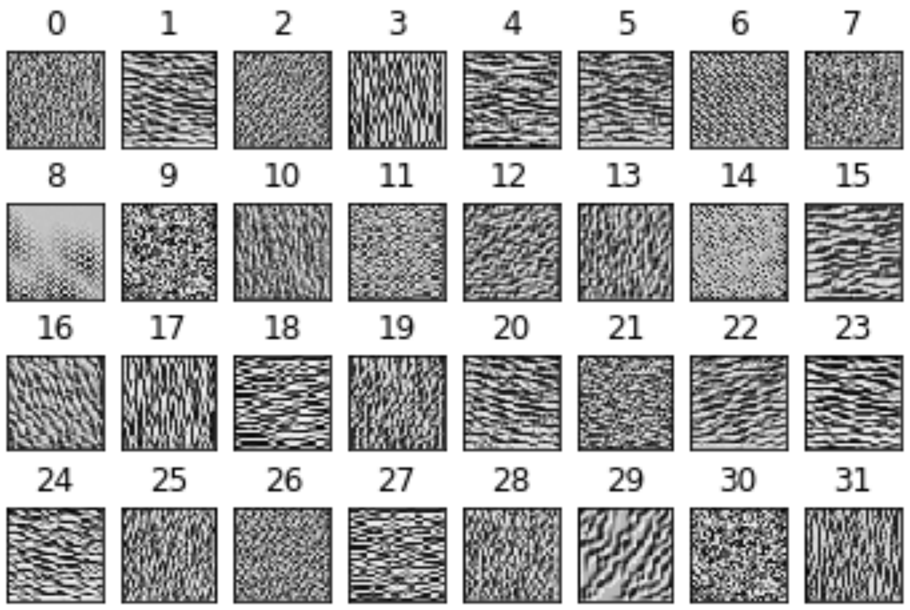
4.(1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？

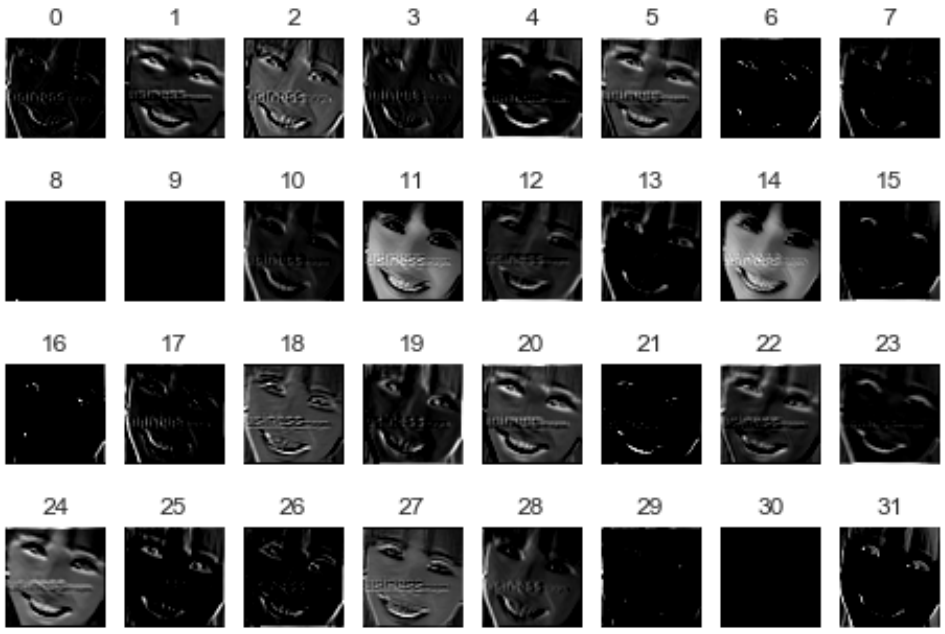


每種 class 皆挑選一張圖片來分析，可以發現 model 主要 focus 在五官的輪廓，尤其是嘴巴延伸至臉頰的部分

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。

First layer：conv2D_1(32 filters with kernel size = 3)





選擇的原圖為上一題的 happy