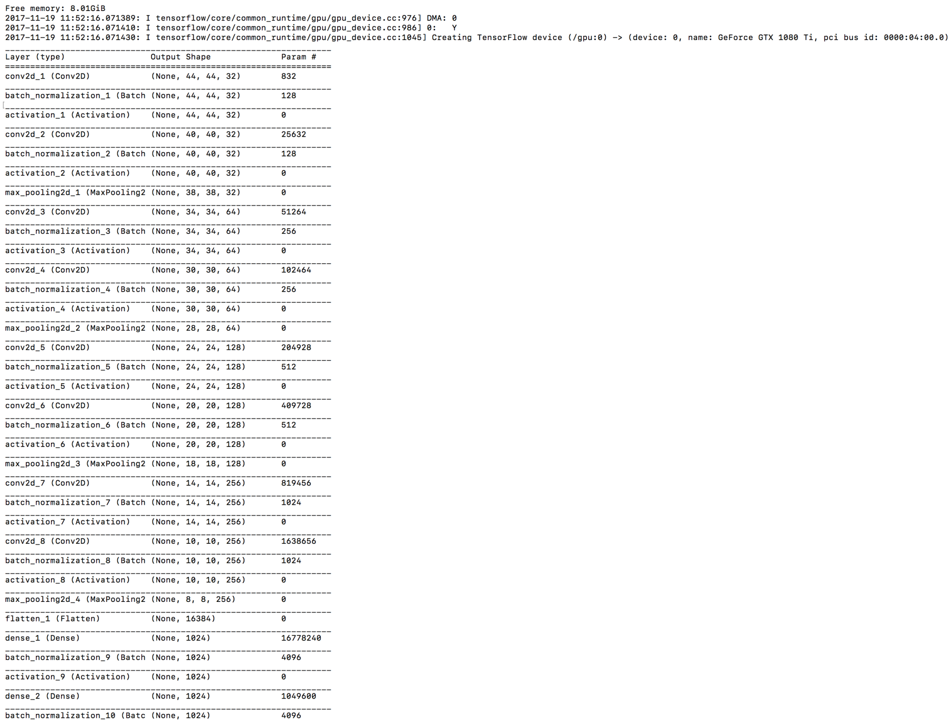
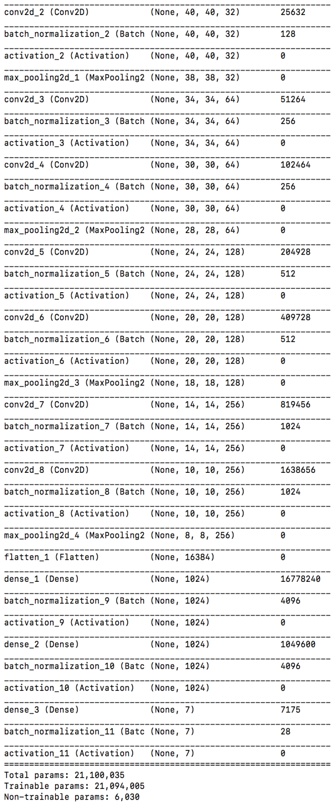
學號：B03902125 系級： 資工四　姓名：林映廷

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:)

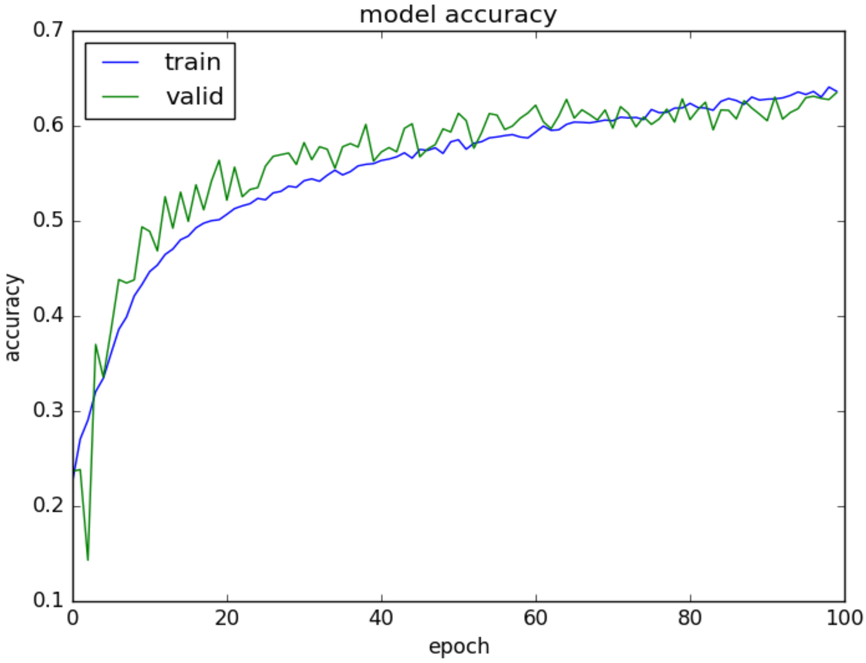
   答：



```
Free memory: 8.01GiB
2017-11-19 11:52:16.071389: I tensorflow/core/common_runtime/gpu/gpu_device.cc:976] DMA: 0
2017-11-19 11:52:16.071410: I tensorflow/core/common_runtime/gpu/gpu_device.cc:986] 0:   Y
2017-11-19 11:52:16.071430: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1045] Creating TensorFlow device (/gpu:0) -> (device: 0, name: GeForce GTX 1080 Ti, pci bus id: 0000:04:00.0)
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
conv2d_1 (Conv2D)            (None, 44, 44, 32)        832       
_________________________________________________________________
batch_normalization_1 (Batch (None, 44, 44, 32)        128       
_________________________________________________________________
activation_1 (Activation)    (None, 44, 44, 32)        0         
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 40, 40, 32)        25632     
_________________________________________________________________
batch_normalization_2 (Batch (None, 40, 40, 32)        128       
_________________________________________________________________
activation_2 (Activation)    (None, 40, 40, 32)        0         
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 38, 38, 32)        0         
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 34, 34, 64)        51264     
_________________________________________________________________
batch_normalization_3 (Batch (None, 34, 34, 64)        256       
_________________________________________________________________
activation_3 (Activation)    (None, 34, 34, 64)        0         
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 30, 30, 64)        102464    
_________________________________________________________________
batch_normalization_4 (Batch (None, 30, 30, 64)        256       
_________________________________________________________________
activation_4 (Activation)    (None, 30, 30, 64)        0         
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 28, 28, 64)        0         
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 24, 24, 128)       204928    
_________________________________________________________________
batch_normalization_5 (Batch (None, 24, 24, 128)       512       
_________________________________________________________________
activation_5 (Activation)    (None, 24, 24, 128)       0         
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 20, 20, 128)       409728    
_________________________________________________________________
batch_normalization_6 (Batch (None, 20, 20, 128)       512       
_________________________________________________________________
activation_6 (Activation)    (None, 20, 20, 128)       0         
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 18, 18, 128)       0         
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 14, 14, 256)       819456    
_________________________________________________________________
batch_normalization_7 (Batch (None, 14, 14, 256)       1024      
_________________________________________________________________
activation_7 (Activation)    (None, 14, 14, 256)       0         
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 10, 10, 256)       1638656   
_________________________________________________________________
batch_normalization_8 (Batch (None, 10, 10, 256)       1024      
_________________________________________________________________
activation_8 (Activation)    (None, 10, 10, 256)       0         
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 8, 8, 256)         0         
_________________________________________________________________
flatten_1 (Flatten)          (None, 16384)             0         
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              16778240  
_________________________________________________________________
batch_normalization_9 (Batch (None, 1024)              4096      
_________________________________________________________________
activation_9 (Activation)    (None, 1024)              0         
_________________________________________________________________
dense_2 (Dense)              (None, 1024)              1049600   
_________________________________________________________________
batch_normalization_10 (Batc (None, 1024)              4096      
```

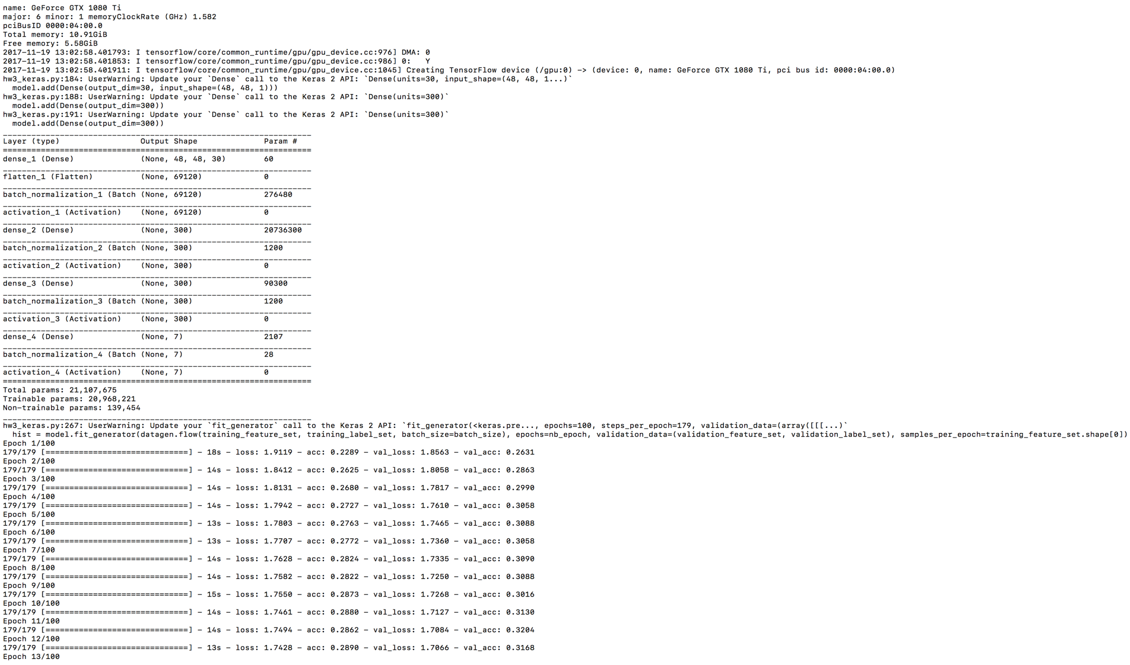


```
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 40, 40, 32)        25632     
_________________________________________________________________
batch_normalization_2 (Batch (None, 40, 40, 32)        128       
_________________________________________________________________
activation_2 (Activation)    (None, 40, 40, 32)        0         
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 38, 38, 32)        0         
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 34, 34, 64)        51264     
_________________________________________________________________
batch_normalization_3 (Batch (None, 34, 34, 64)        256       
_________________________________________________________________
activation_3 (Activation)    (None, 34, 34, 64)        0         
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 30, 30, 64)        102464    
_________________________________________________________________
batch_normalization_4 (Batch (None, 30, 30, 64)        256       
_________________________________________________________________
activation_4 (Activation)    (None, 30, 30, 64)        0         
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 28, 28, 64)        0         
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 24, 24, 128)       204928    
_________________________________________________________________
batch_normalization_5 (Batch (None, 24, 24, 128)       512       
_________________________________________________________________
activation_5 (Activation)    (None, 24, 24, 128)       0         
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 20, 20, 128)       409728    
_________________________________________________________________
batch_normalization_6 (Batch (None, 20, 20, 128)       512       
_________________________________________________________________
activation_6 (Activation)    (None, 20, 20, 128)       0         
_________________________________________________________________
max_pooling2d_3 (MaxPooling2 (None, 18, 18, 128)       0         
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 14, 14, 256)       819456    
_________________________________________________________________
batch_normalization_7 (Batch (None, 14, 14, 256)       1024      
_________________________________________________________________
activation_7 (Activation)    (None, 14, 14, 256)       0         
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 10, 10, 256)       1638656   
_________________________________________________________________
batch_normalization_8 (Batch (None, 10, 10, 256)       1024      
_________________________________________________________________
activation_8 (Activation)    (None, 10, 10, 256)       0         
_________________________________________________________________
max_pooling2d_4 (MaxPooling2 (None, 8, 8, 256)         0         
_________________________________________________________________
flatten_1 (Flatten)          (None, 16384)             0         
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              16778240  
_________________________________________________________________
batch_normalization_9 (Batch (None, 1024)              4096      
_________________________________________________________________
activation_9 (Activation)    (None, 1024)              0         
_________________________________________________________________
dense_2 (Dense)              (None, 1024)              1049600   
_________________________________________________________________
batch_normalization_10 (Batc (None, 1024)              4096      
_________________________________________________________________
activation_10 (Activation)   (None, 1024)              0         
_________________________________________________________________
dense_3 (Dense)              (None, 7)                 7175      
_________________________________________________________________
batch_normalization_11 (Batc (None, 7)                 28        
_________________________________________________________________
activation_11 (Activation)   (None, 7)                 0         
=================================================================
Total params: 21,100,035
Trainable params: 21,094,005
Non-trainable params: 6,030
```

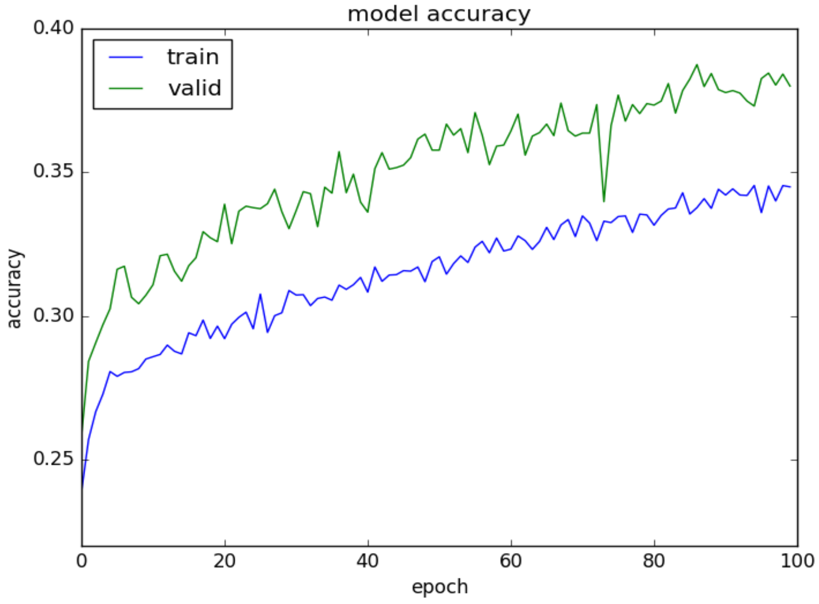


說明:經過 100epochs，得到的準確率為(public+private)/2 = (0.60100+0.61604)/2 = 0.60852

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:)
答：



```
name: GeForce GTX 1080 Ti
major: 6 minor: 1 memoryClockRate (GHz) 1.582
pciBusID 0000:04:00.0
Total memory: 10.91GiB
Free memory: 5.58GiB
2017-11-19 13:02:58.401793: I tensorflow/core/common_runtime/gpu/gpu_device.cc:976] DMA: 0
2017-11-19 13:02:58.401853: I tensorflow/core/common_runtime/gpu/gpu_device.cc:986] 0:   Y
2017-11-19 13:02:58.401911: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1045] Creating TensorFlow device (/gpu:0) -> (device: 0, name: GeForce GTX 1080 Ti, pci bus id: 0000:04:00.0)
hw3_keras.py:184: UserWarning: Update your `Dense` call to the Keras 2 API: `Dense(units=30, input_shape=(48, 48, 1...)`
  model.add(Dense(output_dim=30, input_shape=(48, 48, 1)))
hw3_keras.py:188: UserWarning: Update your `Dense` call to the Keras 2 API: `Dense(units=300)`
  model.add(Dense(output_dim=300))
hw3_keras.py:191: UserWarning: Update your `Dense` call to the Keras 2 API: `Dense(units=300)`
  model.add(Dense(output_dim=300))
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 48, 48, 30)        60
_________________________________________________________________
flatten_1 (Flatten)          (None, 69120)             0
_________________________________________________________________
batch_normalization_1 (Batch (None, 69120)             276480
_________________________________________________________________
activation_1 (Activation)    (None, 69120)             0
_________________________________________________________________
dense_2 (Dense)              (None, 300)               20736300
_________________________________________________________________
batch_normalization_2 (Batch (None, 300)               1200
_________________________________________________________________
activation_2 (Activation)    (None, 300)               0
_________________________________________________________________
dense_3 (Dense)              (None, 300)               90300
_________________________________________________________________
batch_normalization_3 (Batch (None, 300)               1200
_________________________________________________________________
activation_3 (Activation)    (None, 300)               0
_________________________________________________________________
dense_4 (Dense)              (None, 7)                 2107
_________________________________________________________________
batch_normalization_4 (Batch (None, 7)                 28
_________________________________________________________________
activation_4 (Activation)    (None, 7)                 0
=================================================================
Total params: 21,107,675
Trainable params: 20,968,221
Non-trainable params: 139,454
_________________________________________________________________
hw3_keras.py:267: UserWarning: Update your `fit_generator` call to the Keras 2 API: `fit_generator(<keras.pre..., epochs=100, steps_per_epoch=179, validation_data=(array([[[...)`
  hist = model.fit_generator(datagen.flow(training_feature_set, training_label_set, batch_size=batch_size), epochs=nb_epoch, validation_data=(validation_feature_set, validation_label_set), samples_per_epoch=training_feature_set.shape[0])
Epoch 1/100
179/179 [==============================] - 18s - loss: 1.9119 - acc: 0.2289 - val_loss: 1.8563 - val_acc: 0.2631
Epoch 2/100
179/179 [==============================] - 14s - loss: 1.8412 - acc: 0.2625 - val_loss: 1.8058 - val_acc: 0.2863
Epoch 3/100
179/179 [==============================] - 14s - loss: 1.8131 - acc: 0.2680 - val_loss: 1.7817 - val_acc: 0.2990
Epoch 4/100
179/179 [==============================] - 14s - loss: 1.7942 - acc: 0.2727 - val_loss: 1.7610 - val_acc: 0.3058
Epoch 5/100
179/179 [==============================] - 13s - loss: 1.7803 - acc: 0.2763 - val_loss: 1.7465 - val_acc: 0.3088
Epoch 6/100
179/179 [==============================] - 13s - loss: 1.7707 - acc: 0.2772 - val_loss: 1.7360 - val_acc: 0.3058
Epoch 7/100
179/179 [==============================] - 14s - loss: 1.7628 - acc: 0.2824 - val_loss: 1.7335 - val_acc: 0.3090
Epoch 8/100
179/179 [==============================] - 14s - loss: 1.7582 - acc: 0.2822 - val_loss: 1.7250 - val_acc: 0.3088
Epoch 9/100
179/179 [==============================] - 15s - loss: 1.7550 - acc: 0.2873 - val_loss: 1.7268 - val_acc: 0.3016
Epoch 10/100
179/179 [==============================] - 14s - loss: 1.7461 - acc: 0.2880 - val_loss: 1.7127 - val_acc: 0.3130
Epoch 11/100
179/179 [==============================] - 14s - loss: 1.7494 - acc: 0.2862 - val_loss: 1.7084 - val_acc: 0.3204
Epoch 12/100
179/179 [==============================] - 13s - loss: 1.7428 - acc: 0.2890 - val_loss: 1.7066 - val_acc: 0.3168
Epoch 13/100
```

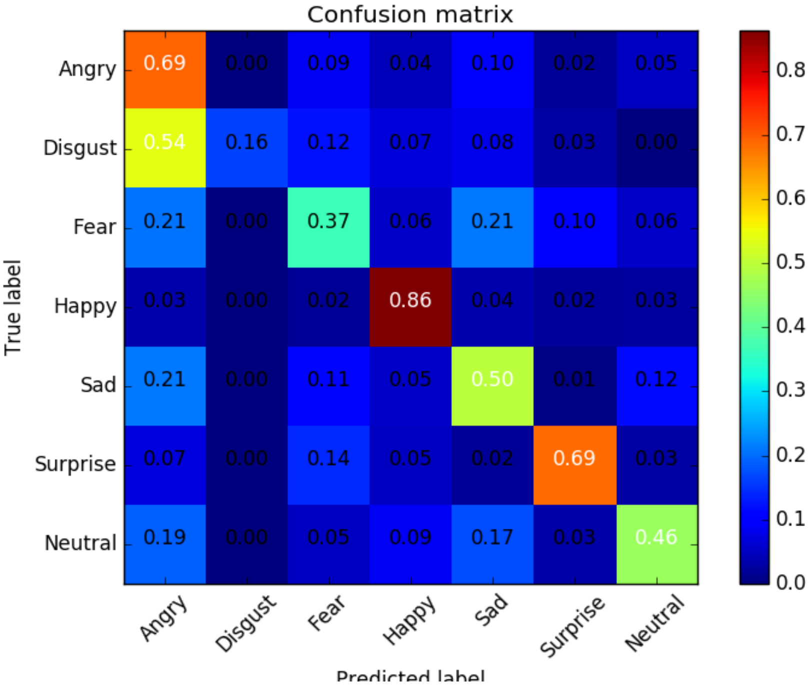


說明:經過 100epochs，得到的準確率為(public+private)/2 = (0.35107+0.35692)/2 = 0.353995

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]

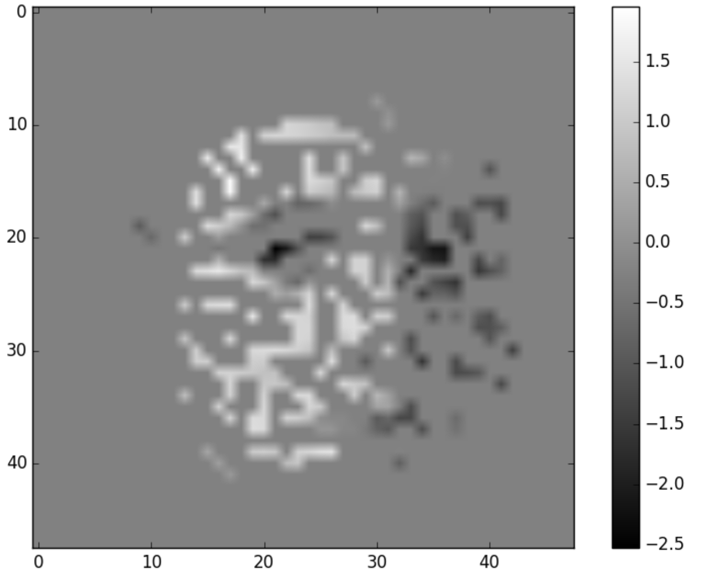
(Collaborators:)

答：

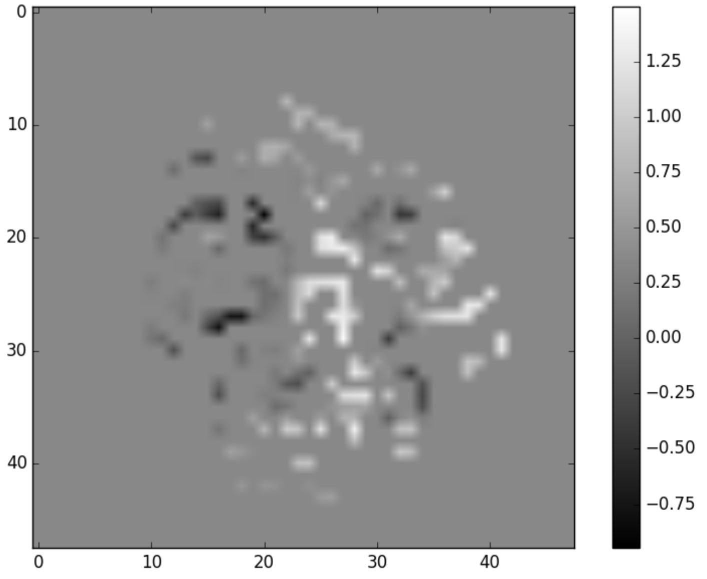


說明:由 confusion matrix 得知，angry 和 disgust 會很容易混肴。

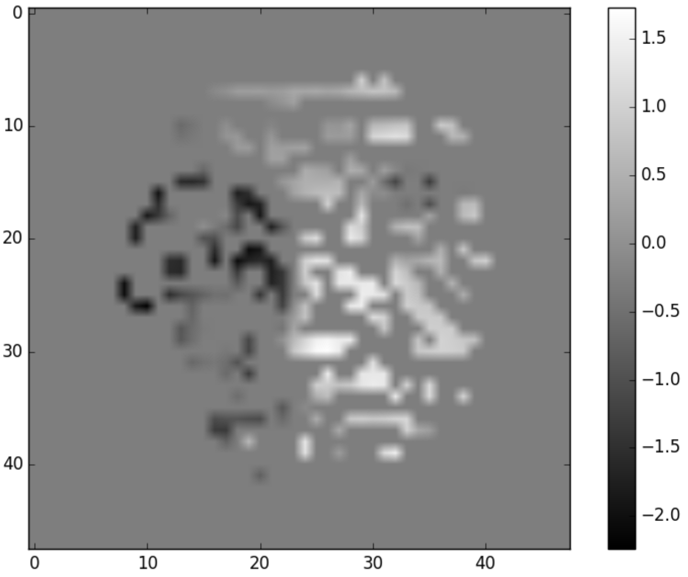
4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？
(Collaborators:)
答：



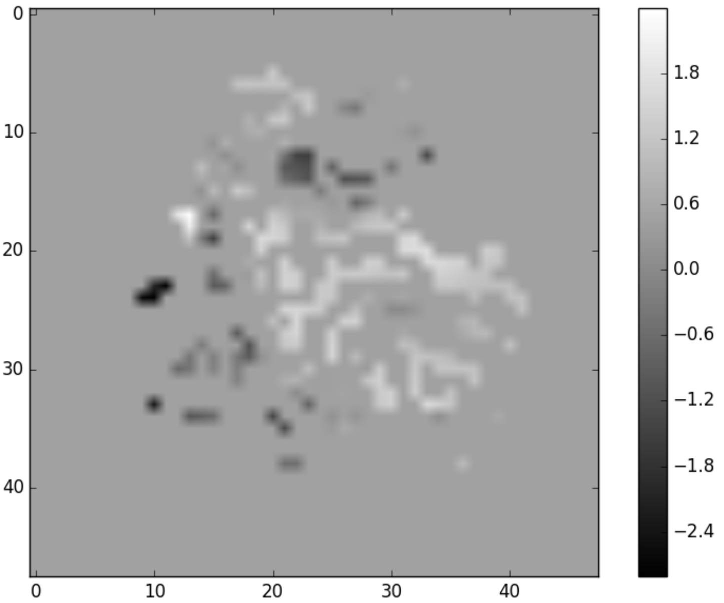
1000.png



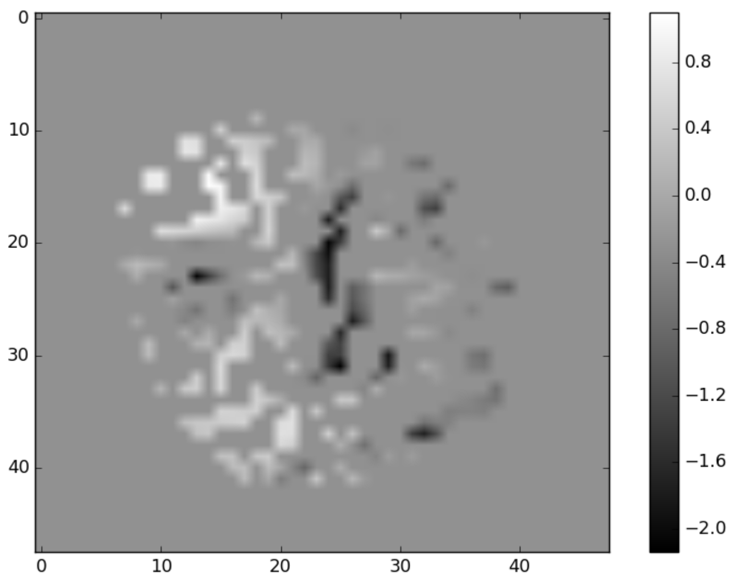
12000.png



19000.png



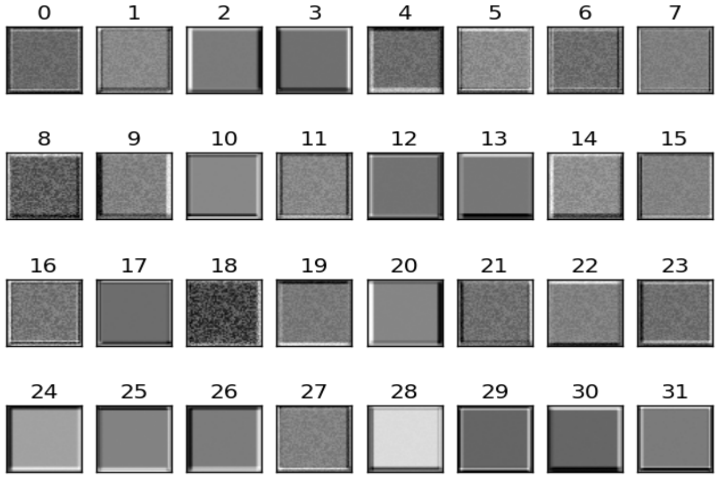
24500.png



27800.png

說明:亮點集中在中間的地方，有時偏左，有時偏右;暗點也集中在中間的地方，亮點的另外一側。很明顯 CNN 有抓到臉部的特徵。

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：



Filters of layer conv2d_1(after 100 epochs)

說明：無法在 conv2d_1 觀察出哪些 class 的圖片會被哪些 filter 給 activate(因為已臨近遲交的 deadline，故無太多時間分析)。