

1. (1%) 請說明這次使用的**model**架構，包含各層維度及連接方式。

這次我使用的是簡單的cnn model只用了三層convolution 2d 最後再附上三層fcn

便有非常高的正確率了，使用pytorch 實作cnn

詳細的model架構如下，每個conv後都有接有 dropout、maxpooling、relu

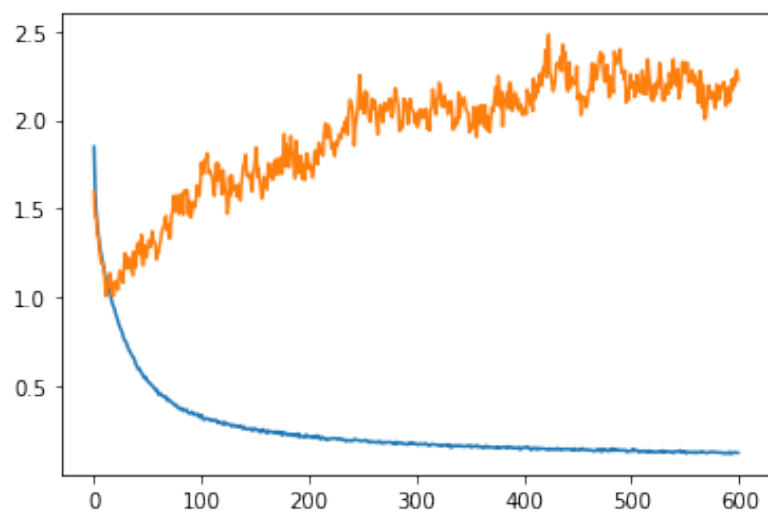
最後在接上fully connected layer 把feature 變成7維的結果

```
ImageNet(  
  (conv1): Sequential(  
    (0): Conv2d(1, 48, kernel_size=(3, 3), stride=(1, 1))  
    (1): Dropout2d(p=0.3)  
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
    (3): ReLU()  
  )  
  (conv2): Sequential(  
    (0): Conv2d(48, 128, kernel_size=(3, 3), stride=(1, 1))  
    (1): Dropout2d(p=0.4)  
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
    (3): ReLU()  
  )  
  (conv3): Sequential(  
    (0): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1))  
    (1): Dropout2d(p=0.4)  
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)  
    (3): ReLU()  
  )  
  (fc1): Linear(in_features=4096, out_features=512, bias=True)  
  (fc2): Linear(in_features=512, out_features=256, bias=True)  
  (fc3): Linear(in_features=256, out_features=7, bias=True)  
)
```

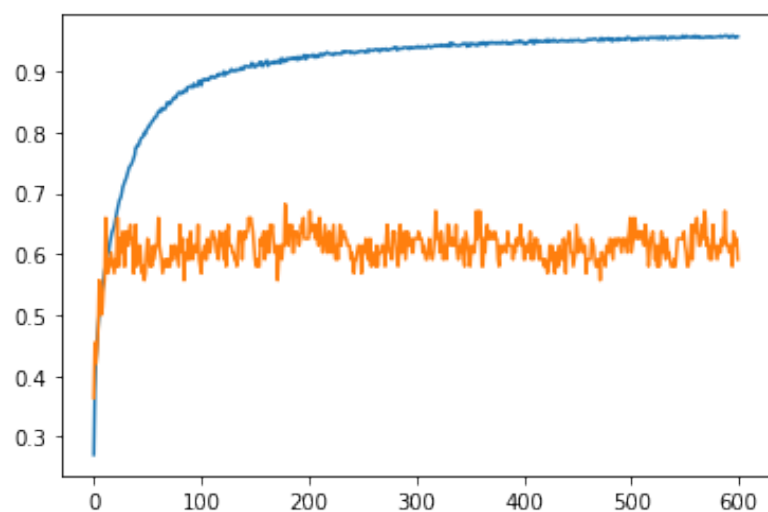
2. (1%) 請附上**model**的**training/validation history (loss and accuracy)**。

為下圖分別是我的 training/validaion loss 與accuracy

Loss



Accuracy



可以看出validation無論是loss 與 accuracy皆在epoch為30左右就差不多了

其中正確率維持在0.6我認為情緒辨識大概能在testing拿到8成正確率就算已經很好了

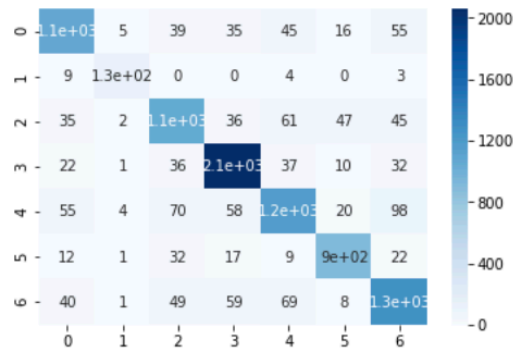
3. (1%) 畫出**confusion matrix**分析哪些類別的圖片容易使**model**搞混，並簡單說明。

(ref: https://en.wikipedia.org/wiki/Confusion_matrix)

confusion matrix如下

```
[[1064    5    39    35    45    16    55]
 [   9  131     0     0     4     0     3]
 [   35    2 1058    36    61    47    45]
 [   22    1   36 2057    37    10    32]
 [   55    4   70   58 1188    20    98]
 [   12    1   32   17    9  901    22]
 [   40    1   49   59   69    8 1290]]
```

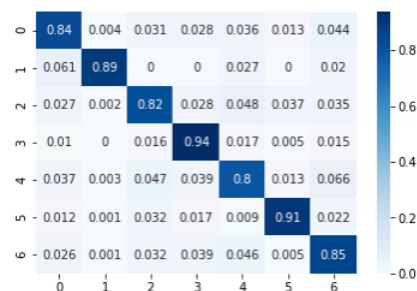
Out[63]: <matplotlib.axes._subplots.AxesSubplot at 0x1381f9ef0>



對row做normalize

```
[[0.845 0.004 0.031 0.028 0.036 0.013 0.044]
 [0.061 0.891 0.    0.    0.027 0.    0.02 ]
 [0.027 0.002 0.824 0.028 0.048 0.037 0.035]
 [0.01  0.    0.016 0.937 0.017 0.005 0.015]
 [0.037 0.003 0.047 0.039 0.796 0.013 0.066]
 [0.012 0.001 0.032 0.017 0.009 0.906 0.022]
 [0.026 0.001 0.032 0.039 0.046 0.005 0.851]]
```

Out[64]: <matplotlib.axes._subplots.AxesSubplot at 0x1382a3828>



0~7為助教給的各種情緒

厭惡較易被誤判為 生氣。

恐懼和難過 容易相互搞混

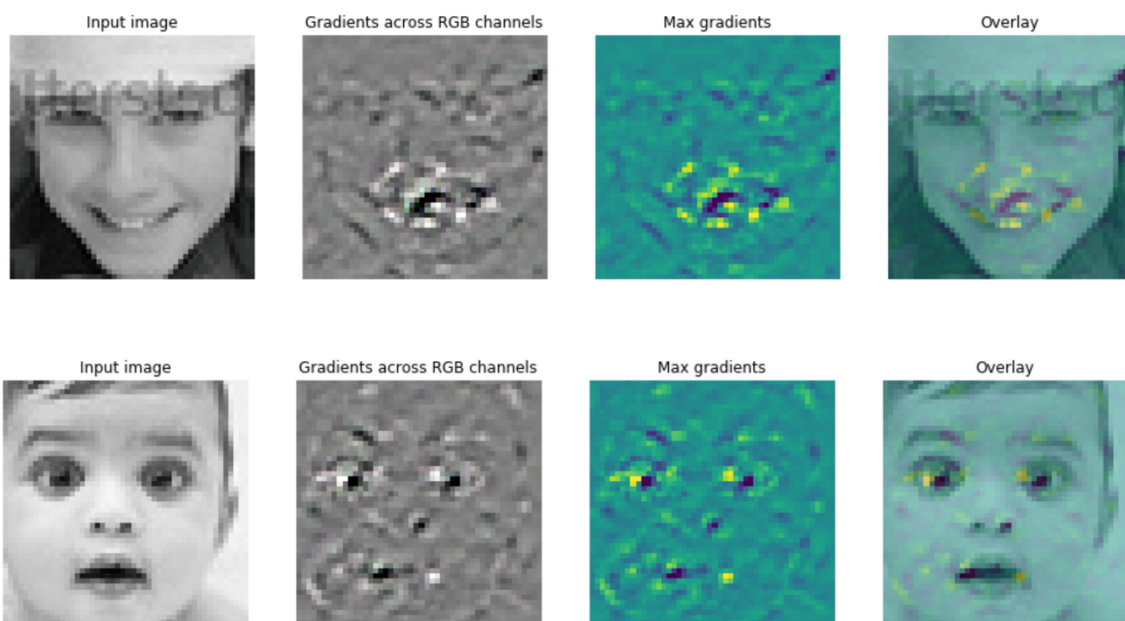
高興(class 3)是最不容易被分錯的類別，因為 3那行那列最多零

【關於第四及第五題】

可以使用簡單的 **3-layer CNN model** [64, 128, 512] 進行實作。

4.(1%) 畫出**CNN model**的**saliency map**，並簡單討論其現象。

(ref: <https://reurl.cc/Qpig8b>)



可以看出第一張是高興的圖，我們所訓練的cnn在嘴巴附近反應最大，可以想成他在偵測嘴角上揚的特徵，而第二張則是在眼睛與嘴巴的地方也很明顯，可能是在判斷圖片是否有張大眼睛的特徵以判定為驚訝。

5(1%) 畫出最後一層的filters最容易被哪些feature activate。

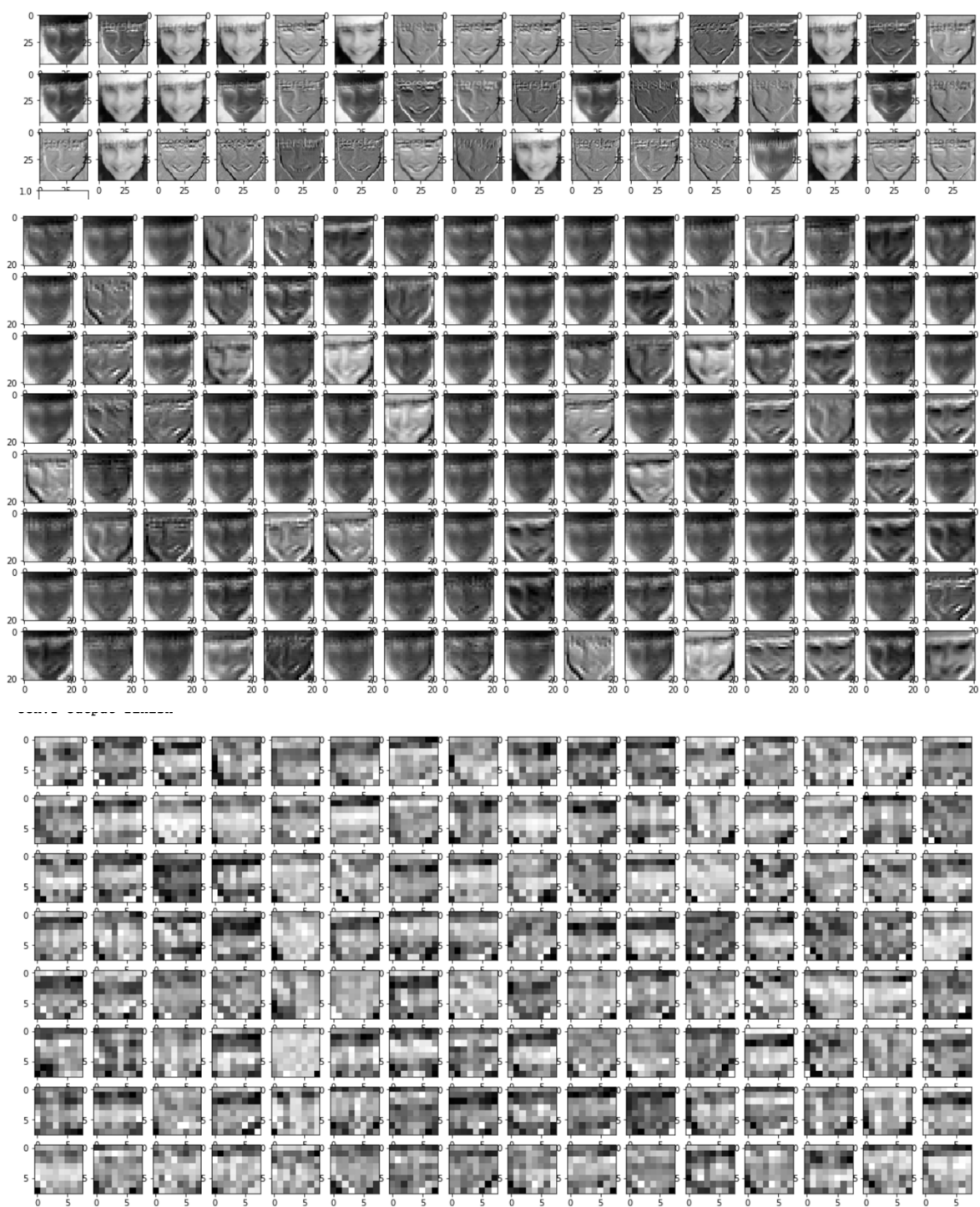
(ref: <https://reurl.cc/ZnrgYg>)

我的model也只有三層conv layer

每層的filters 所activate 的特徵可由下面我輸出的圖觀察

可以看出臉的輪廓跟微笑是filters所在意的feature

第一層甚至有好幾層都可以看出原圖



6.(3%)Refer to math problem

https://hackmd.io/JIZ_0Q3dStSw0t0O0w6NdW

$$\begin{aligned}
 1. \quad W_{reg} &= \left[\frac{W_{reg} + 2P_1 - K_1}{S_1} \right] + 1 \\
 H_{reg} &= \left[\frac{H_{reg} + 2P_2 - K_2}{S} \right] + 1 \\
 2. \quad \lambda \cdot \frac{\partial L}{\partial \beta} &= \frac{\partial L}{\partial \beta} \\
 \frac{\partial L}{\partial \sigma_B^2} &= \sum_{i=1}^n \left(\frac{\partial L}{\partial \sigma_B^2} \right) \frac{\partial \sigma_B^2}{\partial \beta} \\
 \frac{\partial L}{\partial \mu_B} &= \sum_{i=1}^n \left(\frac{\partial L}{\partial \mu_B} \right) \frac{\partial \mu_B}{\partial \beta} \\
 \frac{\partial L}{\partial \sigma_B^2} &= \sum_{i=1}^n \left(\frac{\partial L}{\partial \sigma_B^2} \right) \frac{\partial \sigma_B^2}{\partial \beta} \\
 \frac{\partial L}{\partial \mu_B} &= \sum_{i=1}^n \left(\frac{\partial L}{\partial \mu_B} \right) \frac{\partial \mu_B}{\partial \beta} \\
 \frac{\partial L}{\partial \beta} &= \sum_{i=1}^n \left(\frac{\partial L}{\partial \beta} \right) \frac{\partial \beta}{\partial \beta}
 \end{aligned}$$

$$\begin{aligned}
 3. \quad \frac{\partial L_c}{\partial Z_c} &= - \sum_{i=1}^n y_i \frac{\partial L_c}{\partial Z_c} = \sum_{i=1}^n y_i \left(\frac{\partial L_c}{\partial Z_c} \right) \frac{\partial Z_c}{\partial \beta} \\
 &= - y_c (1 - y_c) - \sum_{i=1}^n y_i \frac{\partial L_c}{\partial Z_c} (-y_i y_c) \\
 &= - y_c (1 - y_c) + \sum_{i=1}^n y_i y_c \\
 &= - y_c + y_c + \sum_{i=1}^n y_i y_c \\
 &= \sum_{i=1}^n y_i y_c
 \end{aligned}$$