學號：R07922163 系級： 資工所二 姓名：徐浩翔

1. (1%) 請說明這次使用的model架構，包含各層維度及連接方式。

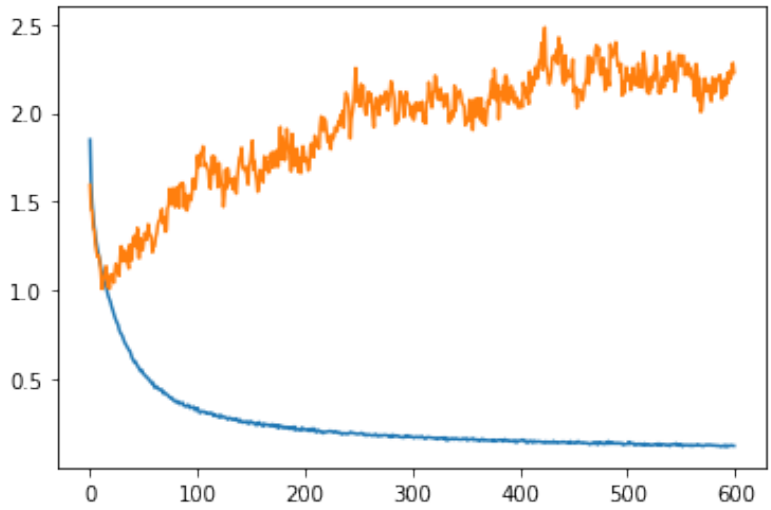
這次我使用的是簡單的cnn model只用了三層convolution 2d 最後再附上三層fcn
便有非常高的正確率了，使用pytorch 實作cnn
詳細的model架構如下，每個conv後都有接有 dropout、maxpooling、relu
最後在接上fully connected layer 把feature 變成7維的結果

```
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()
  )
  (fcn1): Linear(in_features=4096, out_features=512, bias=True)
  (fcn2): Linear(in_features=512, out_features=256, bias=True)
  (fcn3): Linear(in_features=256, out_features=7, bias=True)
)
```

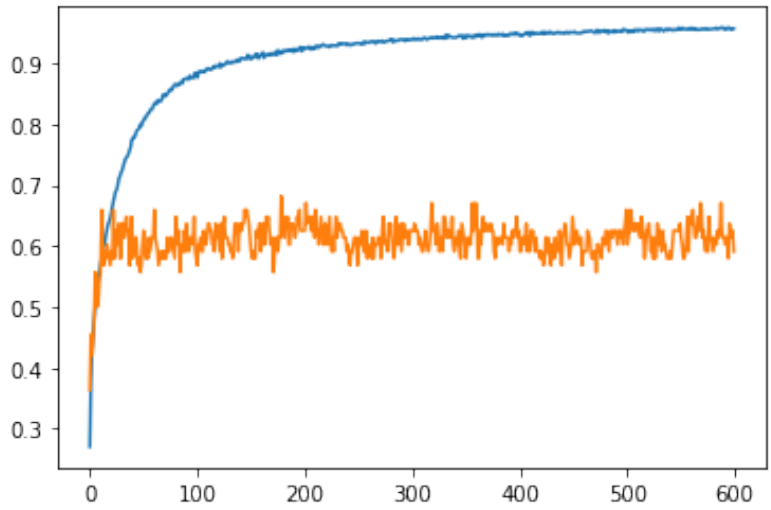
2. (1%) 請附上model的training/validation history (loss and accuracy)。

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我認爲情緒辨識大概能在testing拿到8成正確率就算已經很好了

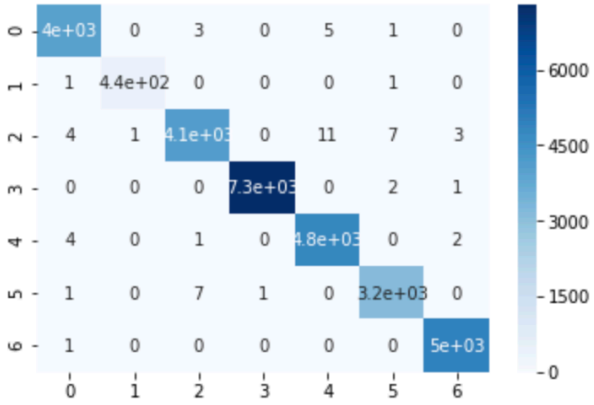
3. (1%) 畫出confusion matrix分析哪些類別的圖片容易使model搞混，並簡單說明。

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

confusion matrix如下

```
[[4032    0    3    0    5    1    0]
 [   1  445    0    0    0    1    0]
 [   4    1 4119    0   11    7    3]
 [   0    0    0 7281    0    2    1]
 [   4    0    1    0 4807    0    2]
 [   1    0    7    1    0 3163    0]
 [   1    0    0    0    0    0 4984]]
```

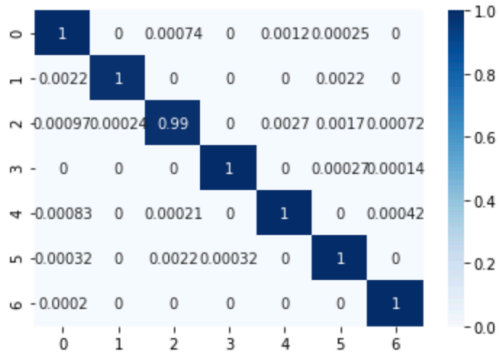
```
Out[92]: <matplotlib.axes._subplots.AxesSubplot at 0x12fb4fba8>
```



對row做normalize

```
[[9.98e-01 0.00e+00 7.42e-04 0.00e+00 1.24e-03 2.47e-04 0.00e+00]
 [2.24e-03 9.96e-01 0.00e+00 0.00e+00 0.00e+00 2.24e-03 0.00e+00]
 [9.65e-04 2.41e-04 9.94e-01 0.00e+00 2.65e-03 1.69e-03 7.24e-04]
 [0.00e+00 0.00e+00 0.00e+00 1.00e+00 0.00e+00 2.75e-04 1.37e-04]
 [8.31e-04 0.00e+00 2.08e-04 0.00e+00 9.99e-01 0.00e+00 4.15e-04]
 [3.15e-04 0.00e+00 2.21e-03 3.15e-04 0.00e+00 9.97e-01 0.00e+00]
 [2.01e-04 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 1.00e+00]]
```

```
Out[93]: <matplotlib.axes._subplots.AxesSubplot at 0x12ff1a128>
```



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

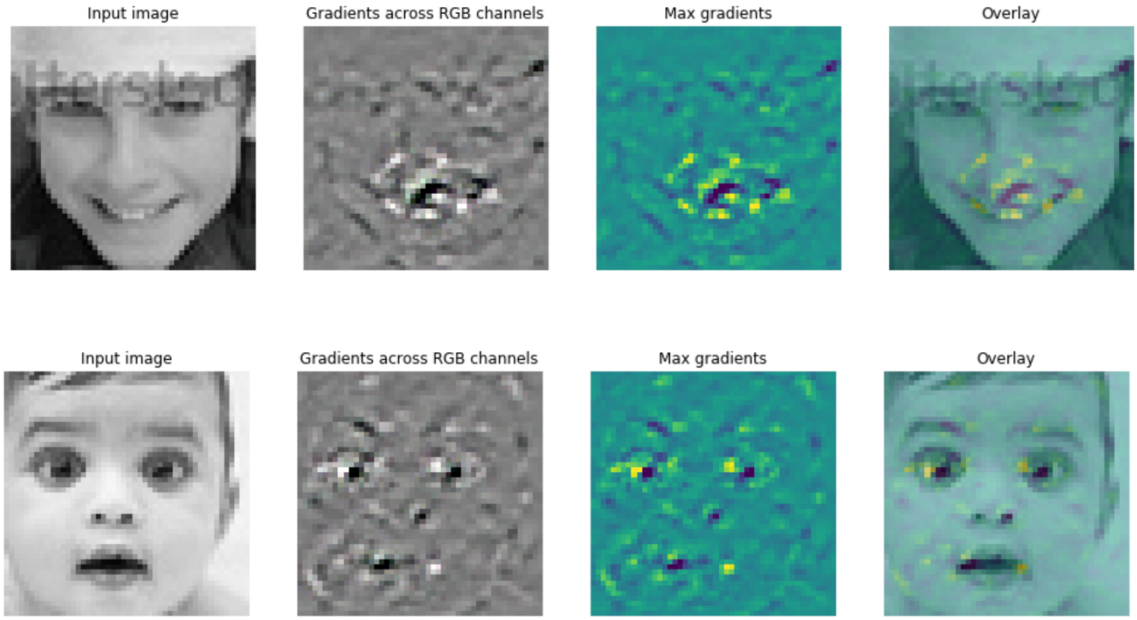
高興(class 3)是最不容易被分錯的類別，因為 3那行那列最多零

[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/Qpjg8b)



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**。

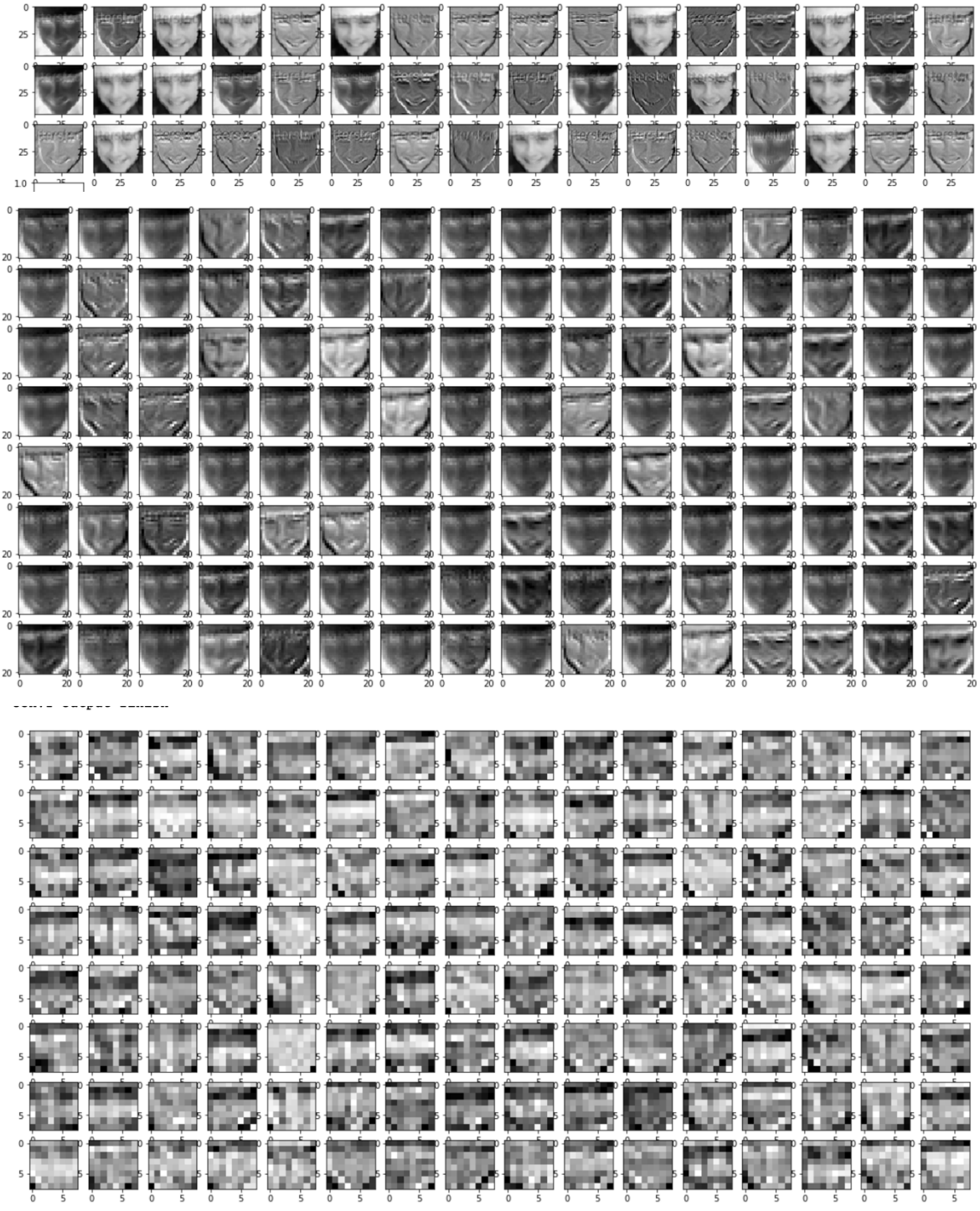
(ref: https://reurl.cc/ZnrgYg)

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

1.

$$W_{new} = \left[\frac{W_{old} + 2P_1 - k_1}{S_1} \right] + 1$$

$$H_{new} = \left[\frac{H_{old} + 2P_2 - k_2}{S} \right] + 1$$

2

$$\frac{\partial l}{\partial x_i} = \frac{\partial l}{\partial y_i} \cdot \gamma$$

$$\frac{\partial l}{\partial \sigma_B^2} = \sum_{i=1}^{m} \frac{\partial l}{\partial \hat{x}_i} (x_i - \mu_B) \frac{-1}{2} (\sigma_B^2 + \varepsilon)^{-\frac{3}{2}}$$

$$\frac{\partial l}{\partial \mu_B} = \left(\sum_{i=1}^{m} \frac{\partial l}{\partial \hat{x}_i} \frac{-1}{\sqrt{\sigma_B^2 + \varepsilon}} \right) + \frac{\partial l}{\partial \sigma_B^2} \cdot \frac{\sum_{i=1}^{m} -2(x_i - \mu_B)}{m}$$

$$\frac{\partial l}{\partial x_i} = \frac{\partial l}{\partial \hat{x}_i} \cdot \frac{1}{\sqrt{\sigma_B^2 + \varepsilon}} + \frac{\partial l}{\partial \sigma_B^2} \cdot \frac{2(x_i - \mu_B)}{m} + \frac{\partial l}{\partial \mu_B} \cdot \frac{1}{m}$$

$$\frac{\partial l}{\partial \gamma} = \sum_{i=1}^{m} \frac{\partial l}{\partial y_i} \cdot \hat{x}_i$$

$$\frac{\partial l}{\partial \beta} = \sum_{i=1}^{m} \frac{\partial l}{\partial y_i}$$

3

$$\frac{\partial L_t}{\partial z_t} = -\sum_i y_i \frac{\partial \log(\hat{y}_i)}{\partial z_t} = \sum_i y_i \frac{1}{(\hat{y}_i)} \frac{\partial (\hat{y}_i)}{\partial z_t}$$

$$= -y_t(1-\hat{y}_t) - \sum_{i \neq t} y_i \frac{1}{\hat{y}_i}(-\hat{y}_i \hat{y}_t)$$

$$= -y_t(1-\hat{y}_t) + \sum_{i \neq t} y_i \hat{y}_t$$

$$= -y_t + y_t \hat{y}_t + \sum_{i \neq t} y_i \hat{y}_t$$

$$= \hat{y}_t \left(\sum_i y_i \right) - y_t = \hat{y}_t - y_t$$