

學號：B030123456 系級：電機四 姓名：刀劍第三季開播了！

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 fc

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

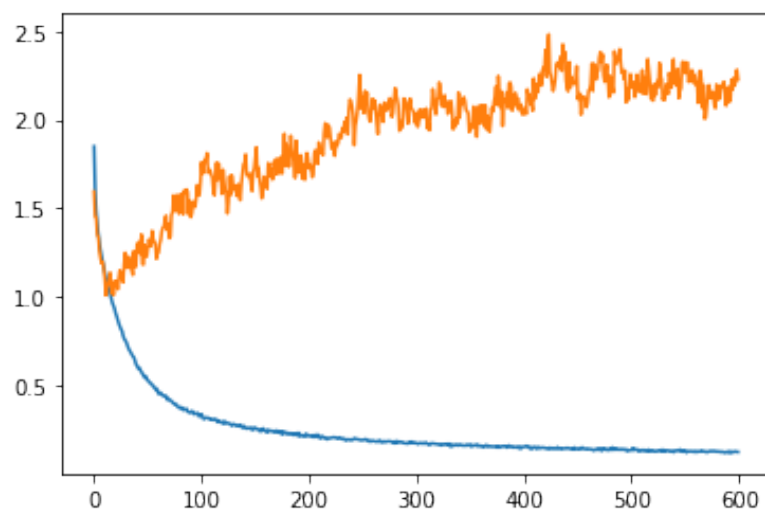
最後在接上 fully connected layer 把 feature 變成 7 維的結果

```
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)  
  )  
)
```

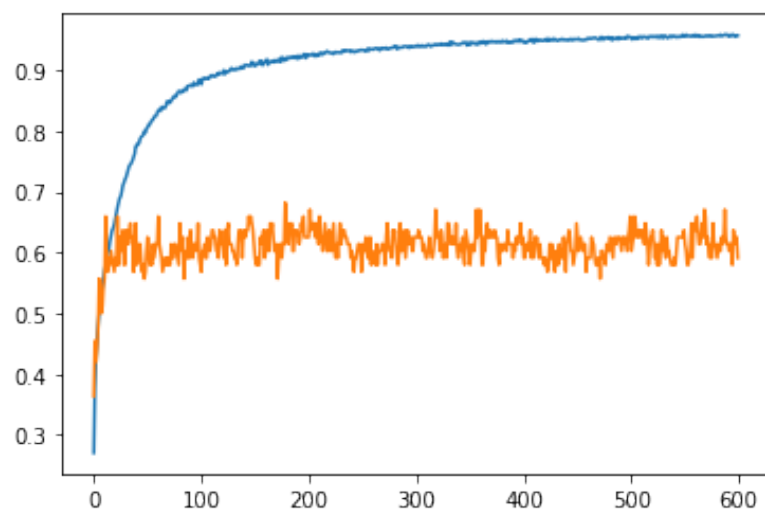
2. (1%) 請附上 model 的 training/validation history (loss and accuracy)。

為下圖分別是我的 training/validation 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

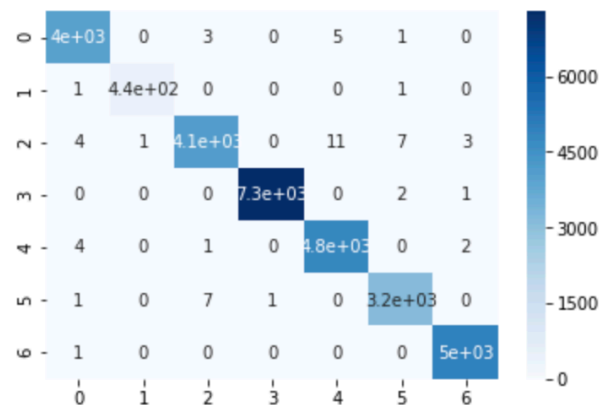
3. (1%) 畫出 **confusion matrix** 分析哪些類別的圖片容易使 **model** 搞混，並簡單說明。

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

confusion matrix 如下

```
[[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]]
```

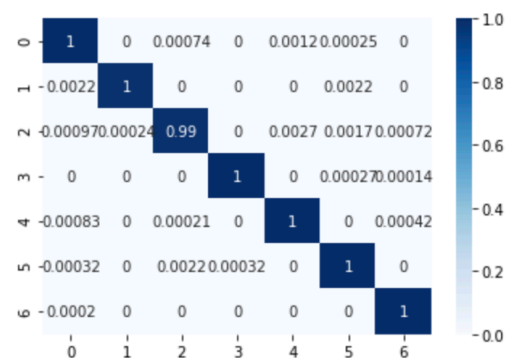
Out[92]: <matplotlib.axes._subplots.AxesSubplot at 0x12fb4fba8>



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

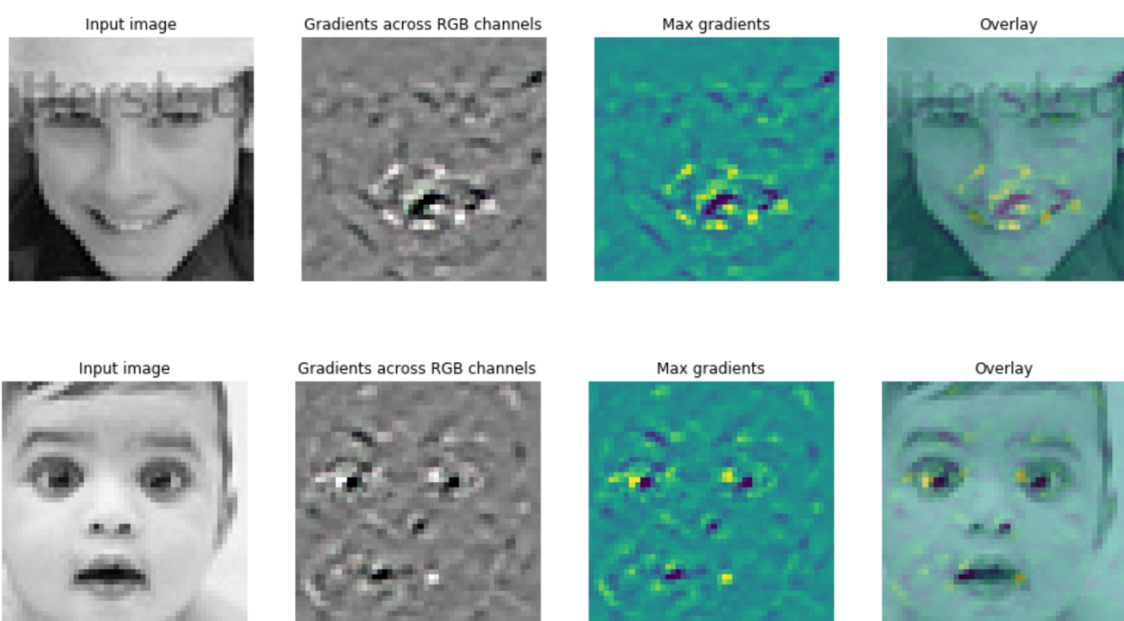
高興(class 3)是最不容易被分錯的類別，因為 3 那行那列最多零

[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/Qpig8b>)



可以看出第一張是高興的圖，我們所訓練的 **cmn** 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**。

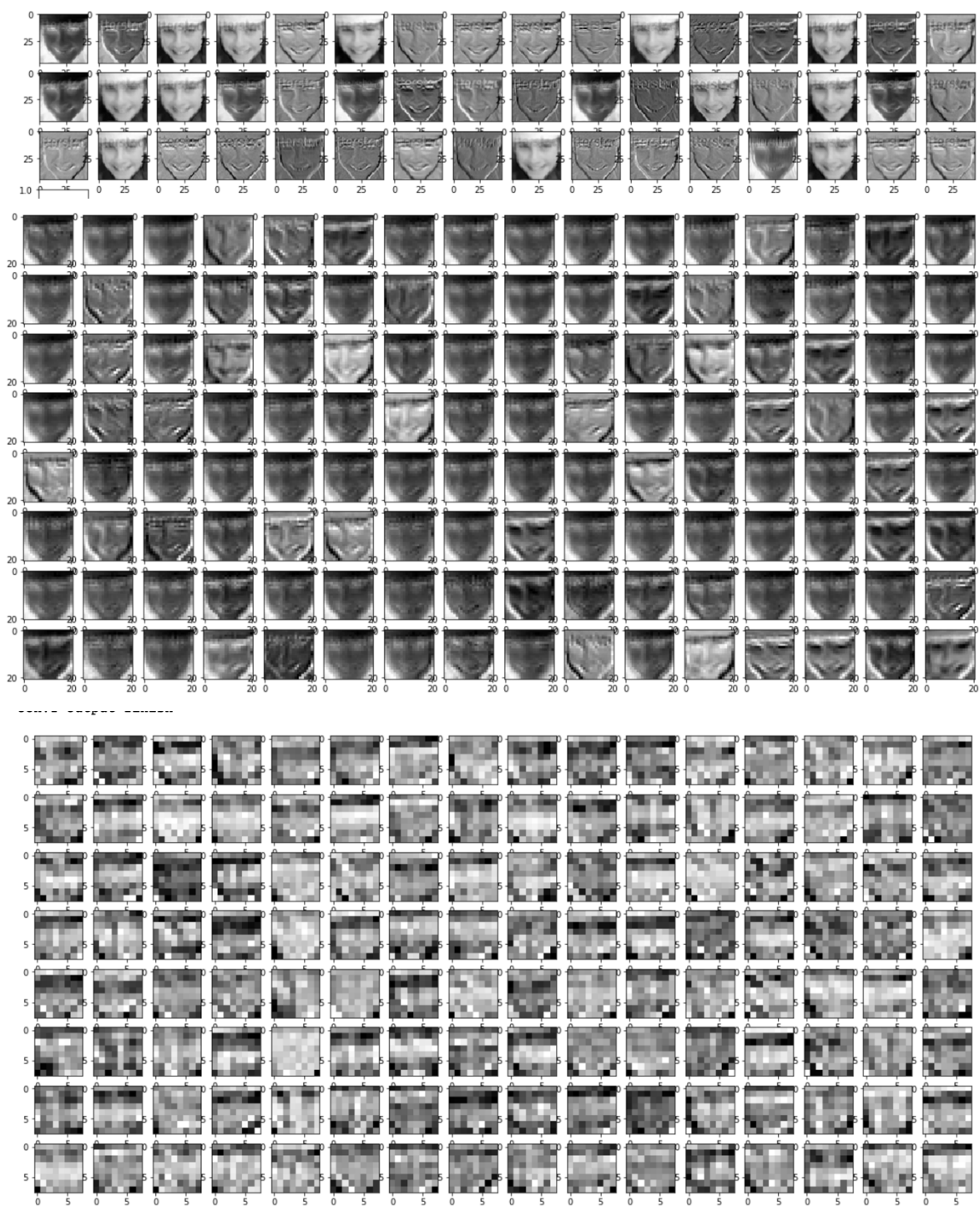
(ref: <https://reurl.cc/ZnrgYg>)

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.

$$W_{\text{new}} = \left[\frac{W_{\text{old}} + 2p_i - k_i}{s_i} \right] + 1$$

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

$$1. \frac{\partial \pi}{\partial e} = \frac{\partial \pi}{\partial e}$$

$$= \frac{\partial \mathcal{L}}{\partial \mathbf{w}} \sum_{i=1}^n \frac{\partial \mathcal{L}}{\partial \mathbf{x}_i} \frac{\partial \mathcal{L}}{\partial \mathbf{y}_i} + \left(\frac{\frac{\partial \mathcal{L}}{\partial \mathbf{y}_i} \frac{\partial \mathcal{L}}{\partial \mathbf{x}_i}}{\frac{\partial \mathcal{L}}{\partial \mathbf{y}_i} \frac{\partial \mathcal{L}}{\partial \mathbf{x}_i} - \frac{\partial \mathcal{L}}{\partial \mathbf{y}_i} \frac{\partial \mathcal{L}}{\partial \mathbf{x}_i}} \right) \frac{\partial \mathcal{L}}{\partial \mathbf{w}} \sum_{i=1}^n \frac{\partial \mathcal{L}}{\partial \mathbf{x}_i} \frac{\partial \mathcal{L}}{\partial \mathbf{y}_i}$$

$$\frac{\partial \ell}{\partial \alpha} = \frac{\partial \ell}{\partial \alpha} \cdot \frac{1}{\sqrt{1+\alpha}} + \frac{\partial \ell}{\partial \alpha} \cdot \frac{2(\chi_i - \mu_B)}{m} + \frac{\partial \ell}{\partial \mu_B} \cdot \frac{1}{m}$$

$$\frac{d}{dt} \sum_{i=1}^n \frac{\partial L}{\partial \dot{x}_i} \cdot \dot{x}_i$$

$$\frac{1}{\rho} \sum_{i=1}^n \frac{\partial e}{\partial x_i} = \frac{\partial e}{\partial \rho}$$

[illegible]

$$= -y_t(1 - y_t) - \sum_{i=1}^n y_i \frac{1}{y_i} (-y_i)$$

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

$$-y_e + y_c + \sum_{i \neq e} y_i y_e$$

$$y_{t+1}^* - y_t^* = y_t^* (\sum_{i=1}^n \dot{z}_i) - y_t^*$$