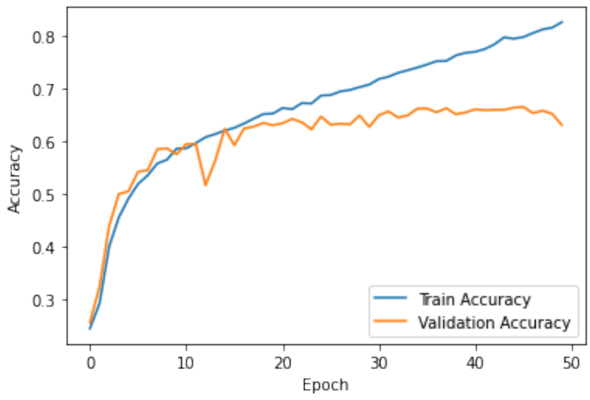
Report Hw3

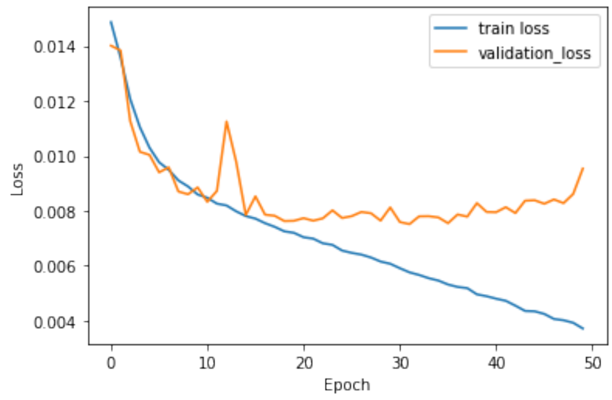
學號：B06705024 系級： 資管四 姓名：郭宇軒

1. (1%) 請說明這次使用的model架構，包含各層維度及連接方式。

nn.Conv2d(3, 64, 3, 1, 1) # [64, 128, 128]
nn.BatchNorm2d(64),
nn.LeakyReLU(),
nn.MaxPool2d(2, 2, 0),
nn.Dropout(0.25),
nn.Conv2d(64, 128, 3, 1, 1), # [128, 64, 64]
nn.BatchNorm2d(128),
nn.LeakyReLU(),
nn.MaxPool2d(2, 2, 0),
nn.Dropout(0.25),
nn.Conv2d(128, 256, 3, 1, 1), #[256, 32, 32]
nn.BatchNorm2d(256),
nn.LeakyReLU(),
nn.MaxPool2d(2, 2, 0),
nn.Dropout(0.25),
nn.Conv2d(256, 512, 3, 1, 1), # [512, 16, 16]
nn.BatchNorm2d(128),
nn.LeakyReLU(),
nn.MaxPool2d(2, 2, 0),
nn.Dropout(0.25),
nn.Conv2d(512, 512, 3, 1, 1), # [512, 8, 8]
nn.BatchNorm2d(128),
nn.LeakyReLU(),
nn.MaxPool2d(2, 2, 0),
nn.Dropout(0.25),
nn.Linear(512*4*4, 1024),
nn.LeakyReLU(),
nn.Linear(1024, 512),
nn.LeakyReLU(),
nn.Linear(512, 7)

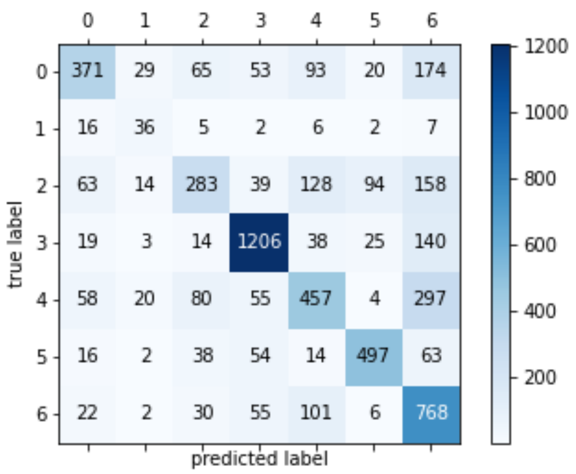
2.(1%) 請附上model的training/validation history (loss and accuracy)。

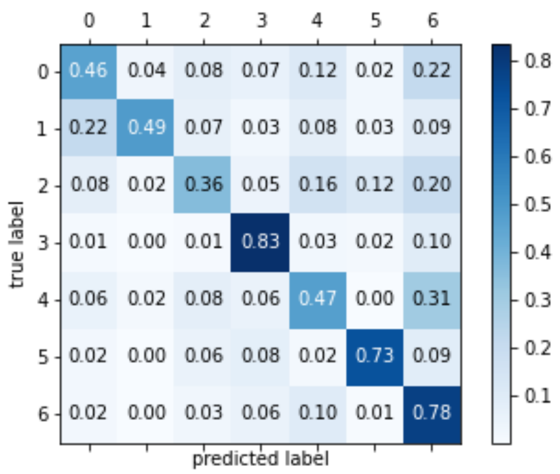




我的Optimizer 使用 Adam (learning rate = 0.001)，training epoch = 50。可能因為 epoch 次數沒有非常多的關係，所以我們還沒辦法非常好的去觀察到整個training 的趨勢。但可以觀察到training accuracy 會呈現穩定上升的趨勢，但Validation Accuracy 則大概在 epoch 20 開始固定在 60%準確率附近很緩慢的上升。而Loss 的部分，training loss 也都呈現穩定下降的趨勢，但 validation loss 則大概在 epoch 20 開始下降的非常不明顯。

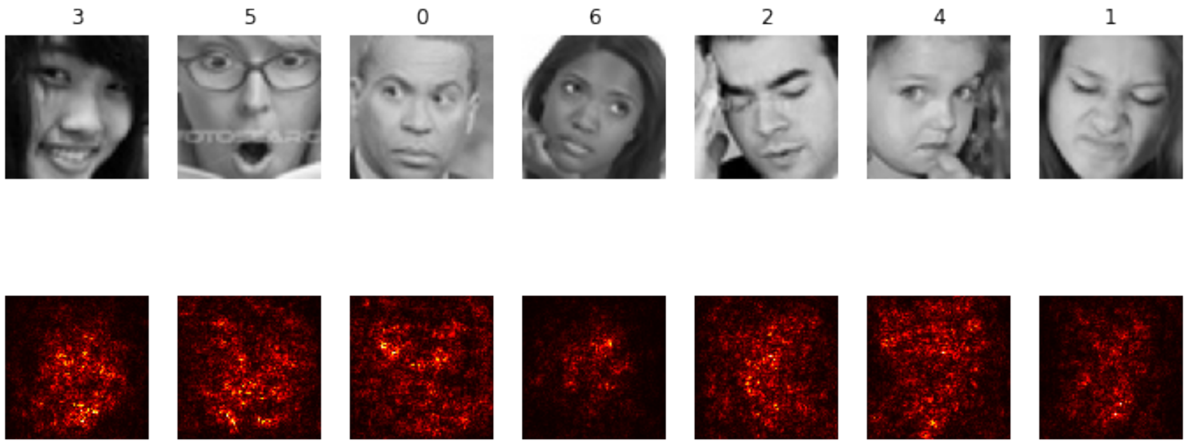
2. (1%) 畫出confusion matrix分析哪些類別的圖片容易使model搞混，並簡單說明。





從 confusion matrix 中我們會發現高興(3)和的表情被判斷的較正確，有8成3的準確率。最不準的則是恐懼表情(2)，只有約 3 成的準確率，而仔細觀察會發現大部分會被錯誤分類到難過(4)和中立(6)表情中、至於難過(4)表情則容易被誤判為中立(6)表情，厭惡(1)表情容易被誤判為生氣(0)，生氣(0)容易被誤判為中立(6)。

3.(1%) 畫出CNN model的saliency map，並簡單討論其現象。



我從每一種表情各挑出一張來實作 saliency map。可以發現大部分被model 挑出來的特徵都集中於五官，像是第一張和第二張高興和驚訝的表情，會著重在嘴巴、生氣的表情則會著重在眼睛、最後一張厭惡的表情則會著重在臉頰肌肉的皺摺。

3. (1%) 畫出最後一層的filters最容易被哪些feature activate。

Layer :5

Layer :15

Layer :30

我藉著兩種不同的方法去 visualize filter。第一種方式，則只挑選最後一層 convolution layer 的特定 filter，來進行 visualize，可以看到眼睛和嘴巴的線條特徵最明顯，但可能因為 CNN 層數沒有了多的關係，所以圖片都還大致看得出原圖是什麼東西。

Reference:https://github.com/utkuozbulak/pytorch-cnn-visualizations

第二種方式為訓練完 5-layer CNN model 後，在最後一個 convolution layer 中取出 64 個 filter來觀察。

可以看到每一個 filter 都還可以大致看到臉部，比較模糊的則可看出眼睛和嘴巴的位置，推測只要藉由找到眼睛和嘴巴之間的位置即可以偵測到臉部，再進一步的來依照不同的表情分類。

Reference: https://debuggercafe.com/visualizing-filters-and-feature-maps-in-convolutional-neural-networks-using-pytorch/

4. (3%)Refer to math problem

https://hackmd.io/@ASZWRvp7SjOEdYLqF3JYdg/HJMbtPOdD

1.

(B, W, H, input_channels) $\longrightarrow$

$$\left(B, \left\lfloor \frac{W+2P_1-k_1}{S_1} +1 \right\rfloor, \left\lfloor \frac{H+2P_2-k_2}{S_2} +1 \right\rfloor, \text{output\_channels} \right)$$

2.

$$\frac{\partial l}{\partial \hat{x_i}}, \frac{\partial l}{\partial \sigma_B^2}, \frac{\partial l}{\partial \mu_B}, \frac{\partial l}{\partial x_i}, \frac{\partial l}{\partial \gamma}, \frac{\partial l}{\partial \beta}. \qquad \hat{y_i} = (\gamma \hat{x_i} + \beta)$$

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

2. $$\frac{\partial l}{\partial \sigma_B^2} = \sum_{i=1}^{m} \frac{\partial l}{\partial \hat{x_i}} \cdot \frac{\partial \hat{x_i}}{\partial \sigma_B^2} = \sum_{i=1}^{m} \frac{\partial l}{\partial \hat{x_i}} (x_i - x_\mu)^{\frac{1}{2}} \cdot \frac{-1}{2} (\sigma_B^2 + \epsilon)^{-\frac{3}{2}}$$

3. $$\frac{\partial l}{\partial \mu_B} = \sum_{i=1}^{m} \frac{\partial l}{\partial \hat{x_i}} \cdot \frac{\partial \hat{x_i}}{\partial \mu_B} + \frac{\partial l}{\partial \sigma_B^2} \frac{\partial \sigma_B^2}{\partial \mu_B}$$

$$= \sum_{i=1}^{m} \frac{\partial l}{\partial \hat{x_i}} \cdot \frac{-1}{\sqrt{\sigma_B^2 + \epsilon}} + \frac{\partial l}{\partial \sigma_B^2} \cdot \frac{\sum_{i=1}^{m} (-2x_i + 2\mu_B)}{m} \quad \#.$$

4. $$\frac{\partial l}{\partial x_i} = \frac{\partial l}{\partial \hat{x_i}} \cdot \frac{\partial \hat{x_i}}{\partial x_i} + \frac{\partial l}{\partial \sigma_B^2} \cdot \frac{\partial \sigma_B^2}{\partial x_i} + \frac{\partial l}{\partial \mu_B} \cdot \frac{\partial \mu_B}{\partial x_i}$$

$$= \sum_{i=1}^{m} \frac{\partial l}{\partial \hat{x_i}} \cdot \frac{+1}{\sqrt{\sigma_B^2 + \epsilon}} + \frac{\partial l}{\partial \sigma_B^2} \cdot \frac{2(x_i - \mu_B)}{m} + \frac{\partial l}{\partial \mu_B} \cdot \frac{1}{m} \quad \#.$$

$$\frac{\partial l}{\partial \gamma} = \sum_{i=1}^{m} \frac{\partial l}{\partial y_i} \frac{\partial y_i}{\partial \gamma} = \sum_{i=1}^{m} \frac{\partial l}{\partial y_i} \cdot \frac{\partial}{\partial \gamma} (\gamma \hat{x_i} + \beta) = \sum_{i=1}^{m} \frac{\partial l}{\partial y_i} \cdot \hat{x_i}$$

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

3.

$$\text{softmax}(z_t) = \frac{e^{z_t}}{\sum_i e^{z_i}}, \quad L_t(y_t, \hat{y}_t) = -y_t \log \hat{y}_t$$

$$\frac{\partial L_t}{\partial z_t} = \frac{\partial L_t}{\partial \hat{y}_t} \frac{\partial \hat{y}_t}{\partial z_t} = \frac{\partial(-y_t \log \hat{y}_t)}{\partial \hat{y}_t} \cdot \frac{\partial\left(\frac{e^{z_t}}{\sum_i e^{z_i}}\right)}{\partial \cdot z_t}$$

$$= \frac{-y_t}{\hat{y}_t} \cdot \frac{e^{z_t}}{\sum e^{z_i}} \cdot \left(1 - \frac{e^{z_t}}{\sum e^{z_i}}\right)$$

$$= \frac{-y_t}{\hat{y}_t} \cdot \hat{y}_t(1-\hat{y}_t) = -y_t + y_t\hat{y}_t = \underline{\underline{\hat{y}_t - y_t}} \;\#.$$