學號：B05702095　系級：　會計四　姓名：黃禹翔

(1%) 請說明這次使用的 model 架構，包含各層維度及連接方式。

我使用的模型架構是：

Input:

Channel = 1, height = 48, width = 48 的圖片

模型的長相及各層的輸出大小如下：

| Layer (type) | Output Shape | Param # |
|---|---|---|
| Conv2d-1 | [-1, 64, 48, 48] | 1,664 |
| LeakyReLU-2 | [-1, 64, 48, 48] | 0 |
| BatchNorm2d-3 | [-1, 64, 48, 48] | 128 |
| MaxPool2d-4 | [-1, 64, 24, 24] | 0 |
| Conv2d-5 | [-1, 128, 24, 24] | 73,856 |
| LeakyReLU-6 | [-1, 128, 24, 24] | 0 |
| BatchNorm2d-7 | [-1, 128, 24, 24] | 256 |
| MaxPool2d-8 | [-1, 128, 12, 12] | 0 |
| Conv2d-9 | [-1, 256, 12, 12] | 295,168 |
| LeakyReLU-10 | [-1, 256, 12, 12] | 0 |
| BatchNorm2d-11 | [-1, 256, 12, 12] | 512 |
| MaxPool2d-12 | [-1, 256, 6, 6] | 0 |
| Conv2d-13 | [-1, 256, 6, 6] | 590,080 |
| LeakyReLU-14 | [-1, 256, 6, 6] | 0 |
| BatchNorm2d-15 | [-1, 256, 6, 6] | 512 |
| MaxPool2d-16 | [-1, 256, 3, 3] | 0 |
| Dropout-17 | [-1, 2304] | 0 |
| Linear-18 | [-1, 1024] | 2,360,320 |
| ReLU-19 | [-1, 1024] | 0 |
| BatchNorm1d-20 | [-1, 1024] | 2,048 |
| Linear-21 | [-1, 256] | 262,400 |
| ReLU-22 | [-1, 256] | 0 |
| Linear-23 | [-1, 7] | 1,799 |

Total params: 3,588,743

模型的第一層為卷積層，參數如下：

(conv1): Sequential(

(0): Conv2d(1, 64, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))

(1): LeakyReLU(negative_slope=0.05)

(2): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)

(3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)

激活函數方面我選擇使用 LeakyReLU，效果比 sigmoid 來得好。為了不遺失邊界資訊，我加了 padding。此外，我加入了 batch normalization 層來幫助訓練，防止 gradient vanish。最後，再加上 Maxpooling 層來幫助減少 overfitting 的問題。

其他層卷積的情況也類似，大體上就是不斷把圖變小，channel 變深。參數如下：

```
  (conv2): Sequential(
    (0): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): LeakyReLU(negative_slope=0.05)
    (2): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
  )
  (conv3): Sequential(
    (0): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): LeakyReLU(negative_slope=0.05)
    (2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
  )
  (conv4): Sequential(
    (0): Conv2d(256, 256, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (1): LeakyReLU(negative_slope=0.05)
    (2): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (3): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1,
ceil_mode=False)
  )
```

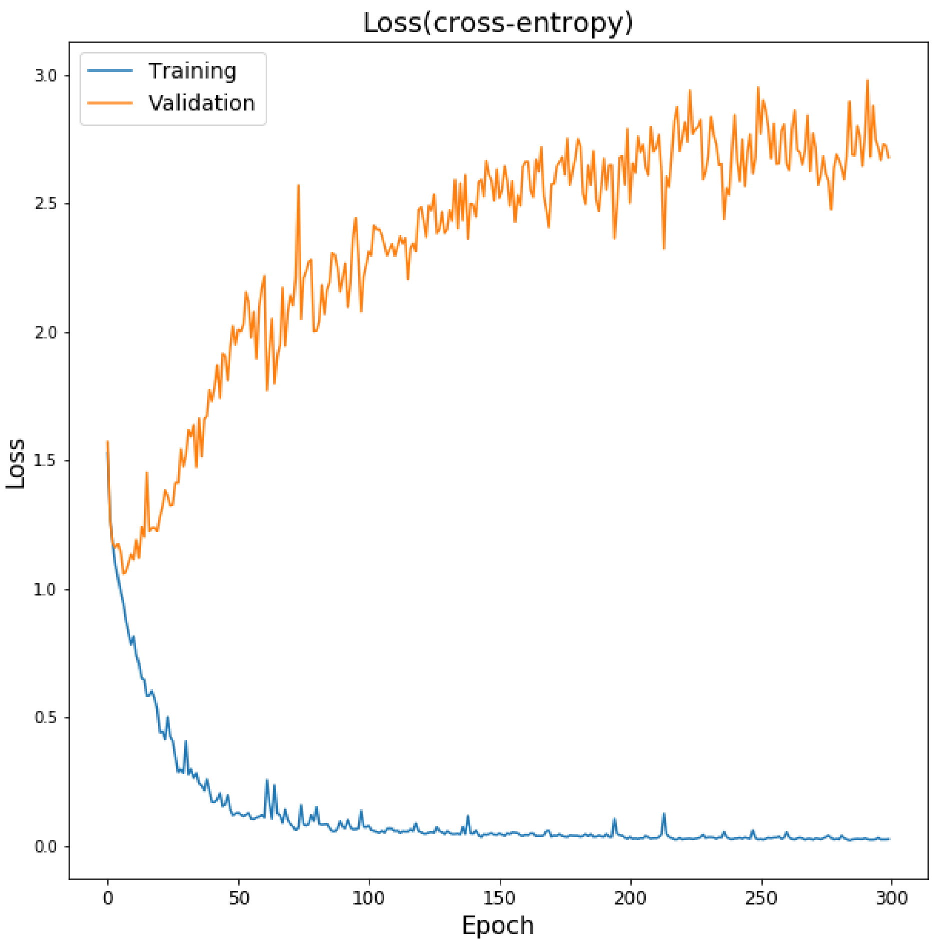
我再卷積層結束後加了一層 average-pooling，降低參數量，並緊接著一個全連接 Flatten 層，用來處理透過卷積層收到的資訊，預測分類結果，參數如下：

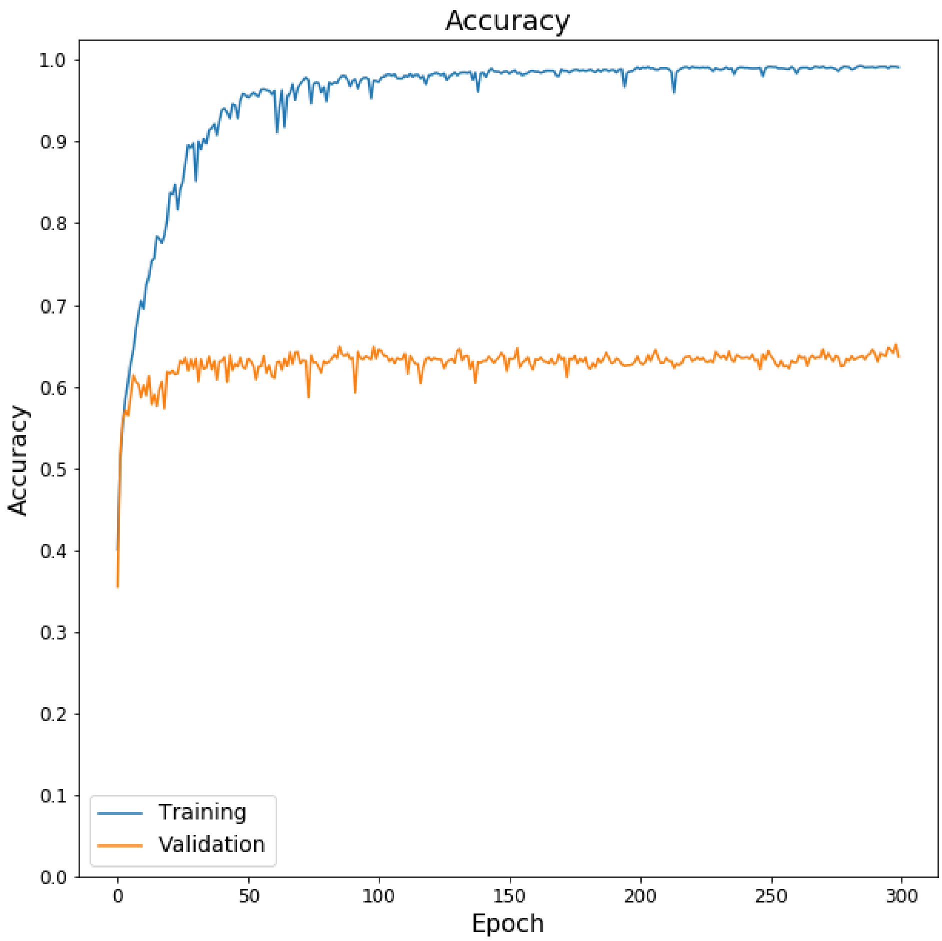
```
  (adapool): AdaptiveAvgPool2d(output_size=(4, 4))
  (fc): Sequential(
    (0): Dropout(p=0.5, inplace=False)
    (1): Linear(in_features=2304, out_features=1024, bias=True)
    (2): ReLU()
    (3): BatchNorm1d(1024, eps=1e-05, momentum=0.1, affine=True,
```

```
track_running_stats=True)
        (4): Linear(in_features=1024, out_features=256, bias=True)
        (5): ReLU()
        (6): Linear(in_features=256, out_features=7, bias=True)
    )
)
```

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

如下圖，可以看到，Training set 滿穩定收斂的，但 Validation Set 的表現還沒有辦法收斂到令人滿意的程度。可能之後要考慮多加一些 Regularization 的方式。



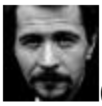



(1%) 畫出 confusion matrix 分析哪些類別的圖片容易使 model 搞混，並簡單說明。

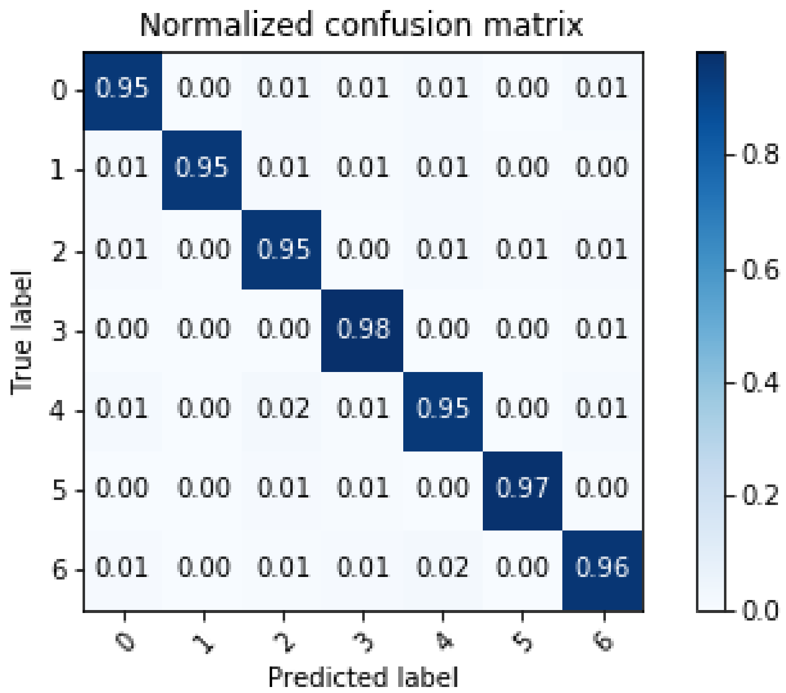
(ref: https://en.wikipedia.org/wiki/Confusion_matrix)

從下圖看起來，模型很容易把難過預測成高興，把中立預測成難過。前者我覺得主要是因為，有些難過的圖片表情比較浮誇，可能跟一些大笑的圖片一樣，都會露出很多嘴巴的部分，眼睛也都會瞇起來，因此模型容易搞混。如：

 (高興) v.s. (難過)

；後者我覺得可能是因為有些難過的表情較不浮誇，和中立一樣，傾向閉著嘴巴，導致模型比較難以分辨。如：(中立) v.s. (難過)。

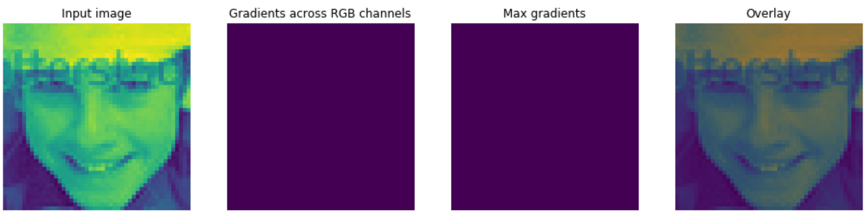
或也有可能是本身存在的標記錯誤。




[0：生氣， 1：厭惡， 2：恐懼， 3：高興， 4：難過， 5：驚訝， 6：中立]

[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。

(1%) 畫出 CNN model 的 saliency map，並簡單討論其現象。
(ref: https://reurl.cc/Qpjg8b)




由於圖片是灰階，因此 RGB channel 沒抓到什麼東西。Max gradient 也不明顯。
看起來模型沒有特別明顯的學到什麼特徵，有滿多可以改進的空間。

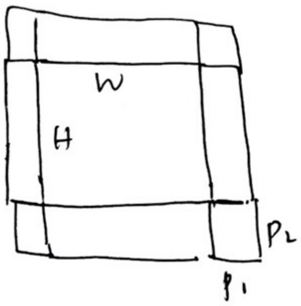
(1%) 畫出最後一層的 filters 最容易被哪些 feature activate。
(ref: https://reurl.cc/ZnrgYg)

(3%)Refer to math problem

https://hackmd.io/JIZ_0Q3dStSw0t0O0w6Ndw

1、

$(B, W, H, \text{input\_channels})$




$\text{stride} = (s_1, s_2)$

$\rightarrow (W+2p_1) \times (H+2p_2)$

$\text{kernel} = (k_1, k_2)$

* 考慮不整除

→ New width

$$= \left\lfloor \frac{W + 2p_1 - (k_1 - 1) - 1}{S_1} + 1 \right\rfloor$$

New height

$$= \left\lfloor \frac{H + 2p_2 - (k_2 - 1) - 1}{S_2} + 1 \right\rfloor$$

→ New shape :

$$\left( B, \left\lfloor \frac{H + 2p_1 - (k_1 - 1) - 1}{S_1} + 1 \right\rfloor, \left\lfloor \frac{H + 2p_2 - (k_2 - 1) - 1}{S_2} + 1 \right\rfloor \right.$$

$$\left. , \text{output\_channels} \right)$$

ML-hw3-Handwrite.

2.

Batch Normalization

*$\eta$ = learning rate

$$\mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i$$

$$\sigma_B^2 = \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_B)^2$$

$$\hat{x}_i = \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}}$$

$$BN_{\gamma,\beta}(x_i) = y_i = \gamma \hat{x}_i + \beta.$$

$$\gamma^{t+1} \leftarrow \gamma^t - \eta \cdot \frac{\partial l}{\partial \gamma}$$

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

② $$\frac{\partial l}{\partial \beta} = \sum_{i=1}^{m} \frac{\partial l}{\partial y_i} \cdot \frac{\partial y_i}{\partial \beta} = \sum_{i=1}^{m} \frac{\partial l}{\partial y_i}$$

③

$$\frac{\partial l}{\partial \hat{x}_i} = \frac{\partial l}{\partial y_i} \cdot \frac{\partial y_i}{\partial \hat{x}_i}$$

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

$$\frac{\partial l}{\partial \mu_B} = \frac{\partial l}{\partial \hat{x}_i} \cdot \frac{\partial \hat{x}_i}{\partial \mu_B} + \frac{\partial l}{\partial \sigma_B^2} \cdot \frac{\partial \sigma_B^2}{\partial \mu}$$

④

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

$$\frac{\partial \sigma_B^2}{\partial \mu_B} = \frac{1}{m} \sum_{i=1}^{m} 2 \cdot (x_i - \mu_B)(-1)$$

⑤

$$\frac{\partial l}{\partial \sigma_B^2} = \frac{\partial l}{\partial \sigma_B^2} \cdot \frac{\partial \hat{x}}{\partial \sigma_B^2}$$

$$\frac{\partial x}{\partial \sigma_B^2} = \frac{\partial}{\partial \sigma_B^2} \sum_{i=1}^{m} (x_i - \mu_B) \cdot (\sigma_B^2 + \varepsilon)^{-0.5}$$

$$= -0.5 \sum_{i=1}^{m} (x_i - \mu) \cdot (\sigma_B^2 + \varepsilon)^{-1.5}$$

4

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

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

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

$$= \sum_{i=1}^{m} \frac{\partial l}{\partial \hat{x}_i} \cdot \frac{-1}{\sqrt{\sigma_B^2 + \varepsilon}}$$

6

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

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

$$+ \frac{\partial l}{\partial \sigma_B^2} \cdot \frac{2(x_i - \mu)}{m}$$

3. $L_t = -y_t \log \hat{y}_t$

$$\hat{y}_t = \text{softmax}(z_t) = \frac{e^{z_t}}{\sum_i e^{z_i}}$$

$$\frac{\partial L_t}{\partial z_t} = \frac{\partial L_t}{\partial \hat{y}_t} \cdot \frac{\partial \hat{y}_t}{\partial z_t}$$

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

$$= -\frac{y_t}{\hat{y}_t} \cdot \frac{e^{z_t} \cdot \sum_i e^{z_i} - e^{z_t} \cdot e^{z_t}}{(\sum_i e^{z_i})^2}$$

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

$$= -\frac{y_t}{\cancel{\hat{y}_t}} \cdot \cancel{\hat{y}_t}(1 - \hat{y}_t)$$

$$= -y_t + y_t \hat{y}_t \ ; \ y_t = 1 \text{ or } 0$$

$$= \begin{cases} 0, & \text{if } y_t = 0 \\ \hat{y}_t - y_t, & \text{if } y_t = 1 \end{cases}$$