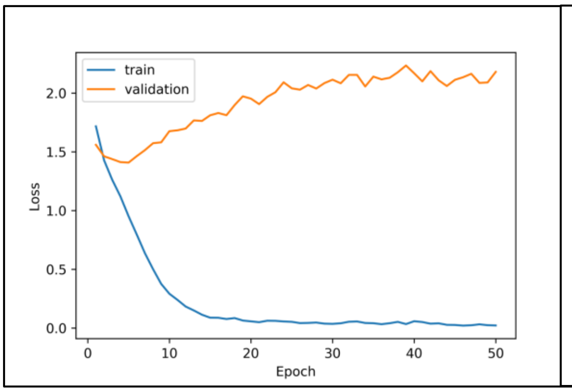
學號：B06507002 系級： 材料三 姓名：林柏勳

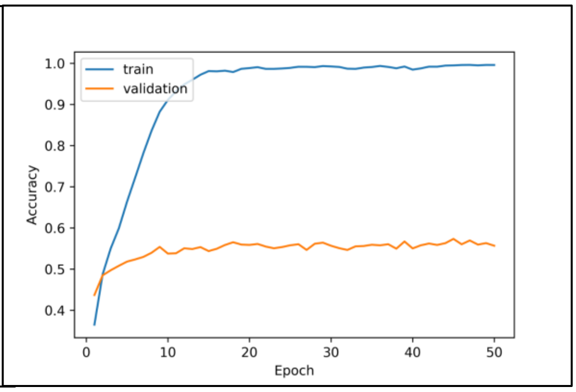
1. (1%) 請說明這次使用的 model 架構，包含各層維度及連接方式。
我在訓練時第一層的 input 為(batch_size,channel,h,w)=(-1,1,48,48)

Layer	Output
Conv2d(1, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))	-1, 32, 48, 48
LeakyReLU(negative_slope=0.05)	-1, 32, 48, 48
BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)	-1, 32, 48, 48
Conv2d(32, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))	-1, 32, 48, 48
LeakyReLU(negative_slope=0.05)	-1, 32, 48, 48
BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)	-1, 32, 48, 48
Conv2d(32, 32, kernel_size=(5, 5), stride=(1, 1), padding=(2, 2))	-1, 32, 48, 48
LeakyReLU(negative_slope=0.05)	-1, 32, 48, 48
BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)	-1, 32, 48, 48
MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)	-1, 32, 24, 24
Dropout(p=0.1, inplace=False)	-1, 32, 24, 24
Conv2d(32, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))	-1, 128, 24, 24
LeakyReLU(negative_slope=0.05)	-1, 128, 24, 24
BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)	-1, 128, 24, 24
Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))	-1, 128, 24, 24
LeakyReLU(negative_slope=0.05)	-1, 128, 24, 24
BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)	-1, 128, 24, 24
Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))	-1, 128, 24, 24
LeakyReLU(negative_slope=0.05)	-1, 128, 24, 24
BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)	-1, 128, 24, 24
MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)	-1, 128, 12, 12
Dropout(p=0.1, inplace=False)	-1, 128, 12, 12
Conv2d(128, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))	-1, 512, 12, 12
LeakyReLU(negative_slope=0.05)	-1, 512, 12, 12
BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)	-1, 512, 12, 12
Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))	-1, 512, 12, 12
LeakyReLU(negative_slope=0.05)	-1, 512, 12, 12
BatchNorm2d(512, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)	-1, 512, 12, 12
Conv2d(512, 512, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))	-1, 512, 12, 12
LeakyReLU(negative_slope=0.05)	-1, 512, 12, 12

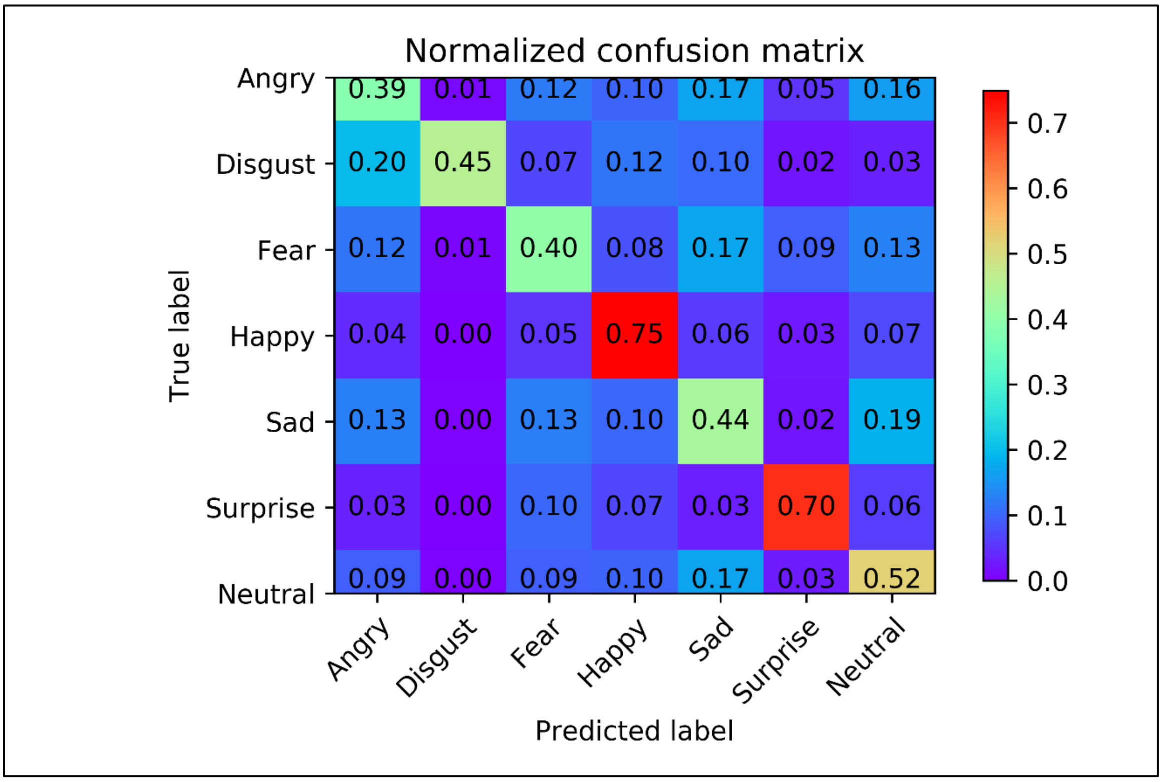
BatchNorm2d(512, eps=1e−05, momentum=0.1, affine=True, track_running_stats=True)	−1, 512, 12, 12
MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1, ceil_mode=False)	−1, 512, 6, 6
Linear(in_features=18432, out_features=512, bias=True)	−1, 512
ReLU()	−1, 512
BatchNorm1d(512, eps=1e−05, momentum=0.1, affine=True, track_running_stats=True)	−1, 512
Linear(in_features=512, out_features=7, bias=True)	−1, 7

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





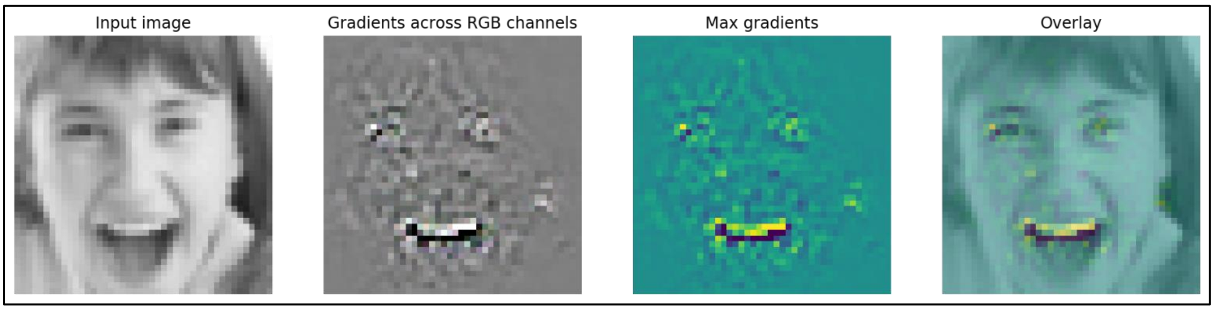
3. (1%) 畫出 confusion matrix 分析哪些類別的圖片容易使 model 搞混，並簡單說明。
(ref: https://en.wikipedia.org/wiki/Confusion_matrix)



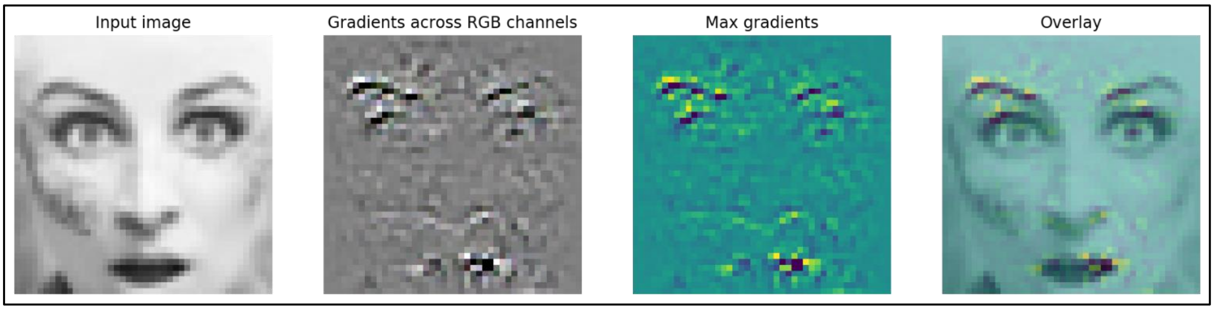
從 Confusion Matrix 的對角線值可以看出，除了 Happy 跟 Surprise 這兩個類別之外，其餘類別均容易使 Model 搞混，它們的分類成功率低於六成，原因可能在於其他類別的資料，圖片裡面的表情不夠顯著，導致分類器很容易誤判成其他類別。

[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。

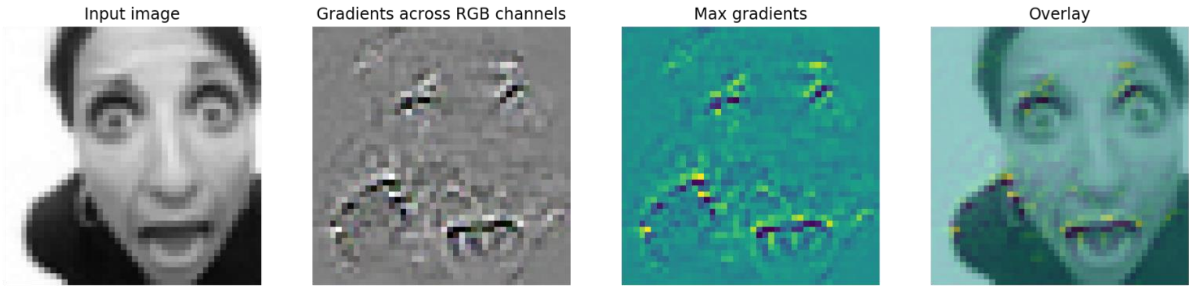
4. (1%) 畫出 CNN model 的 saliency map，並簡單討論其現象。
(ref: https://reurl.cc/Qpjg8b)



這張圖片的類別為 Happy，從 Saliency map 中可以看出，對於分類影響最大的部分為女孩的牙齒，其次為眼睛。

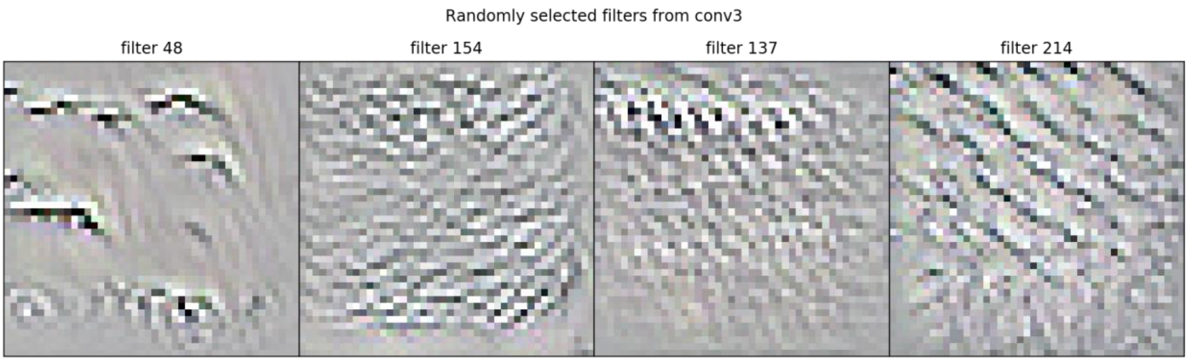


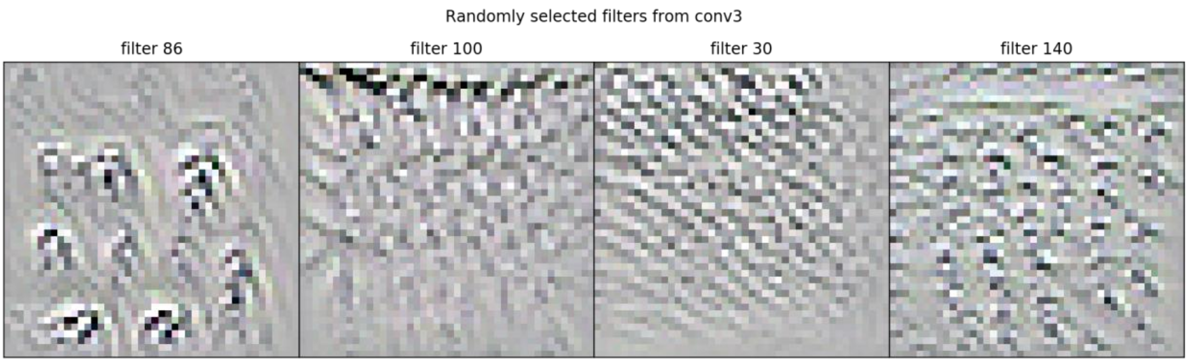
這張圖的類別為 Surprise，從 Saliency map 中可以看出，對於分類影響最大的部分為張的開開的嘴巴跟豎起的眉毛。



這張圖的類別為 Angry，從 Saliency map 中可以看出，對分類影響最大的部分為，眼睛， 開開的嘴巴，跟聳立的肩膀。

5. (1%) 畫出最後一層的 filters 最容易被哪些 feature activate。
(ref: https://reurl.cc/ZnrgYg) (ref: https://reurl.cc/Qpjg8b)





6. (3%)Refer to math problem
https://hackmd.io/JIZ_0Q3dStSw0t0O0w6Ndw

HW3 Hand written Assignment

1. I : $(B, W, H,$ input_channels$)$

Conv2D : (input_channels, output_channels,
kernel_size $= (k_1, k_2)$, stride $= (s_1, s_2)$,
padding $= (P_1, P_2)$)

For each image in a batch, after padding
$(W, H) \rightarrow (W + 2P_1, H + 2P_2)$

Assume after convolution, we get an image size of (n, m)
$(n, m) \in \mathbb{Z}^2$
width height

① $W + 2P_1 \geq k_1 + (n-1)S_1$

$$\frac{W + 2P_1 - k_1}{S_1} \geq n - 1$$

② $H + 2P_2 \geq k_2 + (m-1)S_2$

$$\frac{H + 2P_2 - k_2}{S_2} \geq m - 1$$

The solution of n, m are the largest integers satisfying the inequality of ①, ②.
As a result,

$$n = \left\lfloor \frac{W + 2P_1 - k_1}{S_1} \right\rfloor + 1$$

$$m = \left\lfloor \frac{H + 2P_2 - k_2}{S_2} \right\rfloor + 1$$

So, the output is

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

2. Batch Normalization $\quad \mu_B = \frac{1}{m}\sum_{i=1}^{m} x_i$

$y_i = y_i(\hat{x}_i, \gamma, \beta) = \gamma \hat{x}_i + \beta \qquad l = l(y_i)$

We have $\frac{\partial l}{\partial y_i}$ in gradient descent already.

①

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

②

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

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

Ans in ①

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

③

$$= \sum_{i=1}^{m}\frac{\partial l}{\partial y_i}\gamma\frac{x_i - \mu_B}{(\sigma_B^2 + \epsilon)^{\frac{3}{2}}}\left(-\frac{1}{2}\right)$$

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

Ans in ①

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

$$= \sum_{i=1}^{m}\frac{\partial l}{\partial y_i}\gamma\frac{(-1)}{\sqrt{\sigma_B^2+\epsilon}}$$

④

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

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

$$= \frac{\partial l}{\partial y_i}\gamma\frac{1}{\sqrt{\sigma_B^2+\epsilon}} + \frac{-\gamma}{m}\frac{1}{\sqrt{\sigma_B^2+\epsilon}}\sum_{i=1}^{m}\frac{\partial l}{\partial y_i}$$

$$+ \frac{\cancel{2}^{(-1)}}{m}(x_i - \mu_B)\left(-\cancel{\frac{1}{2}}\right)\sum_{i=1}^{m}\frac{\partial l}{\partial y_i}\gamma\frac{x_i - \mu_B}{(\sigma_B^2+\epsilon)^{\frac{3}{2}}}$$

$$= \frac{1}{m\sqrt{\sigma_B^2+\varepsilon}}\,\gamma\left[\frac{\partial l}{\partial y_i}m - \sum_{i=1}^{m}\frac{\partial l}{\partial y_i} - \frac{x_i-\mu_B}{\sqrt{\sigma_B^2+\varepsilon}}\sum_{i=1}^{m}\frac{\partial l}{\partial y_i}\frac{x_i-\mu_B}{\sqrt{\sigma_B^2+\varepsilon}}\right]$$

$$= \frac{1}{m\sqrt{\sigma_B^2+\varepsilon}}\,\gamma\left[\frac{\partial l}{\partial y_i}m - \sum_{i=1}^{m}\frac{\partial l}{\partial y_i} - \hat{x}_i\sum_{i=1}^{m}\frac{\partial l}{\partial y_i}\hat{x}_i\right]$$

⑤ $$\frac{\partial l}{\partial \gamma} = \sum_{i=1}^{m}\frac{\partial l}{\partial y_i}\frac{\partial y_i}{\partial \gamma} = \sum_{i=1}^{m}\frac{\partial l}{\partial y_i}\hat{x}_i$$

⑥ $$\frac{\partial l}{\partial \beta} = \sum_{i=1}^{m}\frac{\partial l}{\partial y_i}\frac{\partial y_i}{\partial \beta} = \sum_{i=1}^{m}\frac{\partial l}{\partial y_i}$$

Since we can calculate $\frac{\partial l}{\partial \gamma}$, $\frac{\partial l}{\partial \beta}$, we can update them using gradient descent.

3. $L(y, \hat{y}) = -\sum_j y_j \log \hat{y}_j \qquad L_t = -y_t \log \hat{y}_t$

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

$$\frac{\partial L_t}{\partial \hat{y}_t} = \frac{-y_t}{\hat{y}_t}$$

$$\frac{\partial \hat{y}_t}{\partial z_t} = \hat{y}_t + \frac{e^{z_t}}{(\sum_i e^{z_i})^2}(-1)\,e^{z_t} = \hat{y}_t - \hat{y}_t^2$$

$$\frac{\partial L_t}{\partial z_t} = \frac{\partial L_t}{\partial \hat{y}_t}\frac{\partial \hat{y}_t}{\partial z_t} = -\frac{y_t}{\hat{y}_t}(\hat{y}_t - \hat{y}_t^2) = \hat{y}_t - y_t$$

#