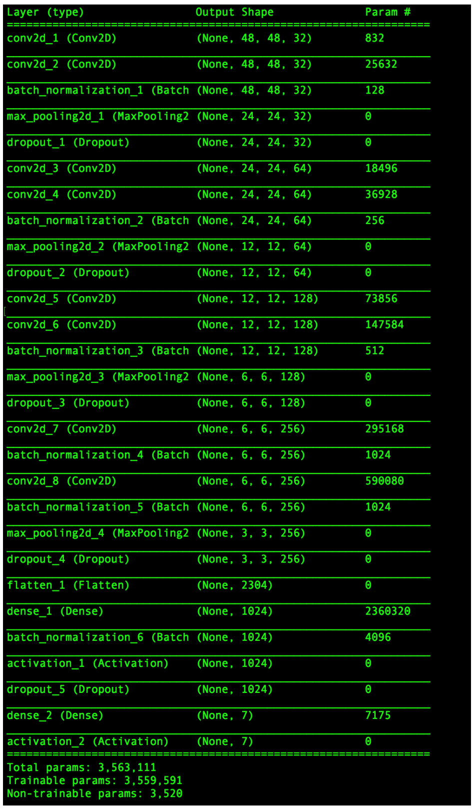
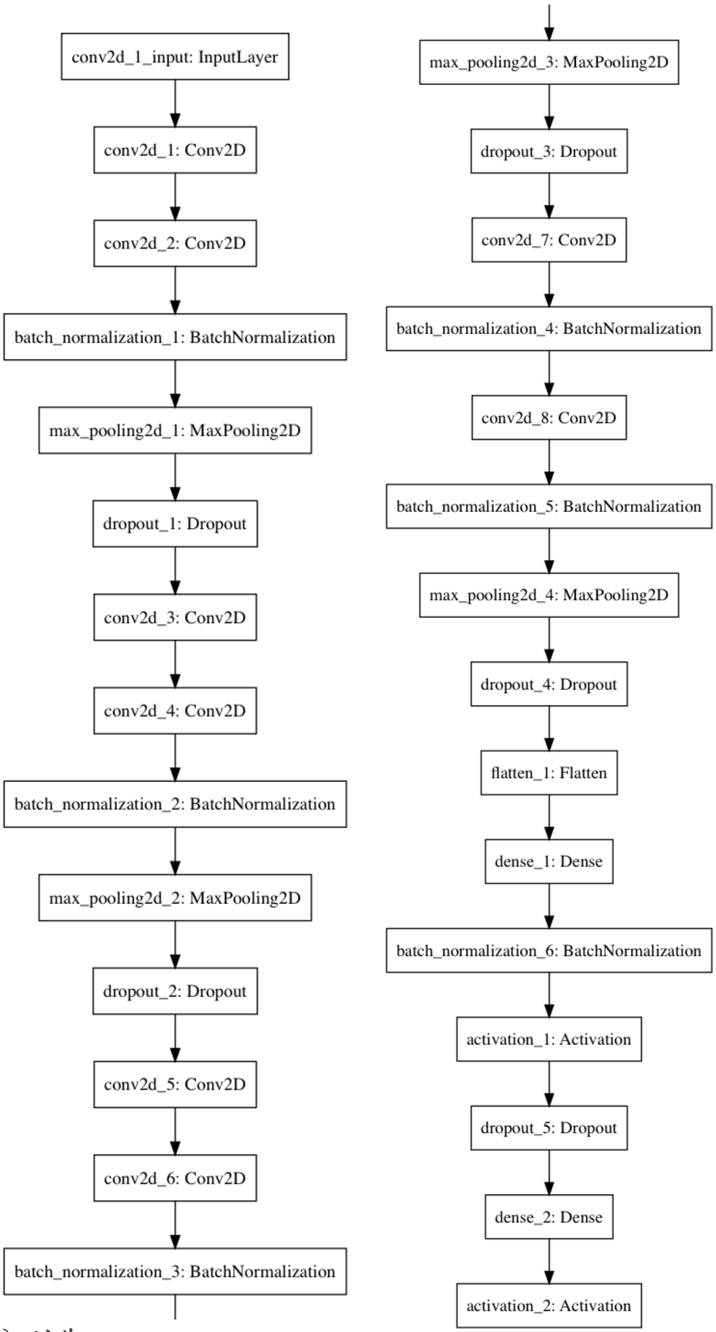
學號：R05525096 系級： 工海碩二 姓名：郭捷

1. (1%) 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？
(Collaborators: 謝朋諺)
答：

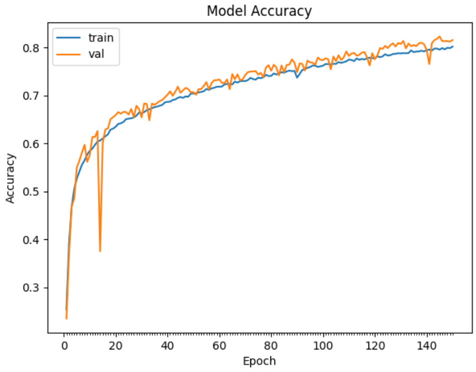


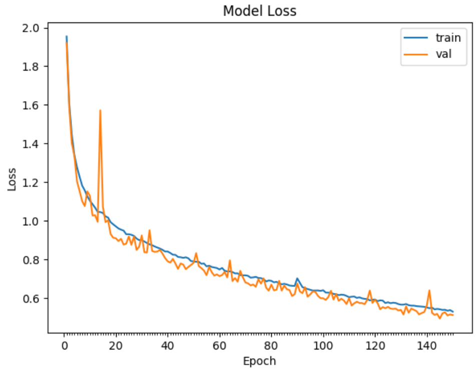
Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 32)	832
conv2d_2 (Conv2D)	(None, 48, 48, 32)	25632
batch_normalization_1 (Batch	(None, 48, 48, 32)	128
max_pooling2d_1 (MaxPooling2	(None, 24, 24, 32)	0
dropout_1 (Dropout)	(None, 24, 24, 32)	0
conv2d_3 (Conv2D)	(None, 24, 24, 64)	18496
conv2d_4 (Conv2D)	(None, 24, 24, 64)	36928
batch_normalization_2 (Batch	(None, 24, 24, 64)	256
max_pooling2d_2 (MaxPooling2	(None, 12, 12, 64)	0
dropout_2 (Dropout)	(None, 12, 12, 64)	0
conv2d_5 (Conv2D)	(None, 12, 12, 128)	73856
conv2d_6 (Conv2D)	(None, 12, 12, 128)	147584
batch_normalization_3 (Batch	(None, 12, 12, 128)	512
max_pooling2d_3 (MaxPooling2	(None, 6, 6, 128)	0
dropout_3 (Dropout)	(None, 6, 6, 128)	0
conv2d_7 (Conv2D)	(None, 6, 6, 256)	295168
batch_normalization_4 (Batch	(None, 6, 6, 256)	1024
conv2d_8 (Conv2D)	(None, 6, 6, 256)	590080
batch_normalization_5 (Batch	(None, 6, 6, 256)	1024
max_pooling2d_4 (MaxPooling2	(None, 3, 3, 256)	0
dropout_4 (Dropout)	(None, 3, 3, 256)	0
flatten_1 (Flatten)	(None, 2304)	0
dense_1 (Dense)	(None, 1024)	2360320
batch_normalization_6 (Batch	(None, 1024)	4096
activation_1 (Activation)	(None, 1024)	0
dropout_5 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 7)	7175
activation_2 (Activation)	(None, 7)	0

Total params: 3,563,111
Trainable params: 3,559,591
Non-trainable params: 3,520



本次我用的 CNN 模型是根據 VGG16 的模型進行修改。總共用了 8 層的 Convolution2D，並且在每兩次的 Convolution2D 之後會做一 BatchNormalization，BatchNormalization 之後會做 MaxPooling，並且在每次 MaxPooling 之後會做 Dropout。在做完卷積選取特徵之後，將其攤平後進入兩個 Dense 層做分類。在 Private 和 Public 的表現分別為 0.68152、0.67344

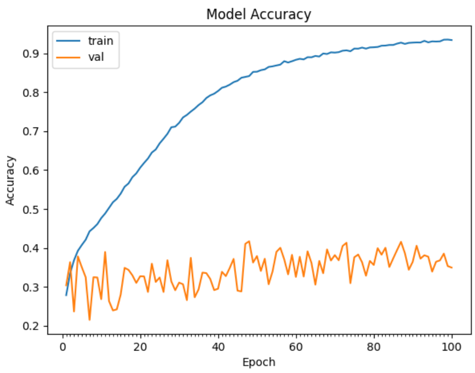


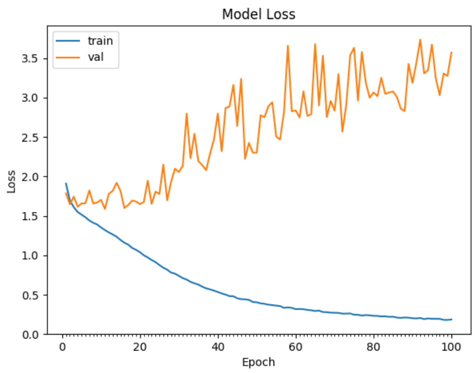


2. (1%) 承上題，請用與上述 CNN 接近的參數量，實做簡單的 DNN model。其模型架構、訓練過程和準確率為何？試與上題結果做比較，並說明你觀察到了什麼？
(Collaborators:)
答：本次我用的 DNN model 一共做了 4 層，units 分別是 2048、512、512、7。但是在總參數上保持與 CNN 的 model 相近，從 train 的速度來說 DNN 速度比 CNN 要快很多，但是準確率相對較低，在 Private 和 Public 的表現分別為 0.36277、0.33797。我認為可能的原因是 DNN 的 model 沒有能找到比較好的 feature 的能力

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 2048)	4720640
batch_normalization_1 (Batch	(None, 2048)	8192
activation_1 (Activation)	(None, 2048)	0
dropout_1 (Dropout)	(None, 2048)	0
dense_2 (Dense)	(None, 512)	1049088
batch_normalization_2 (Batch	(None, 512)	2048
activation_2 (Activation)	(None, 512)	0
dropout_2 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 512)	262656
batch_normalization_3 (Batch	(None, 512)	2048
activation_3 (Activation)	(None, 512)	0
dropout_3 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 7)	3591
activation_4 (Activation)	(None, 7)	0

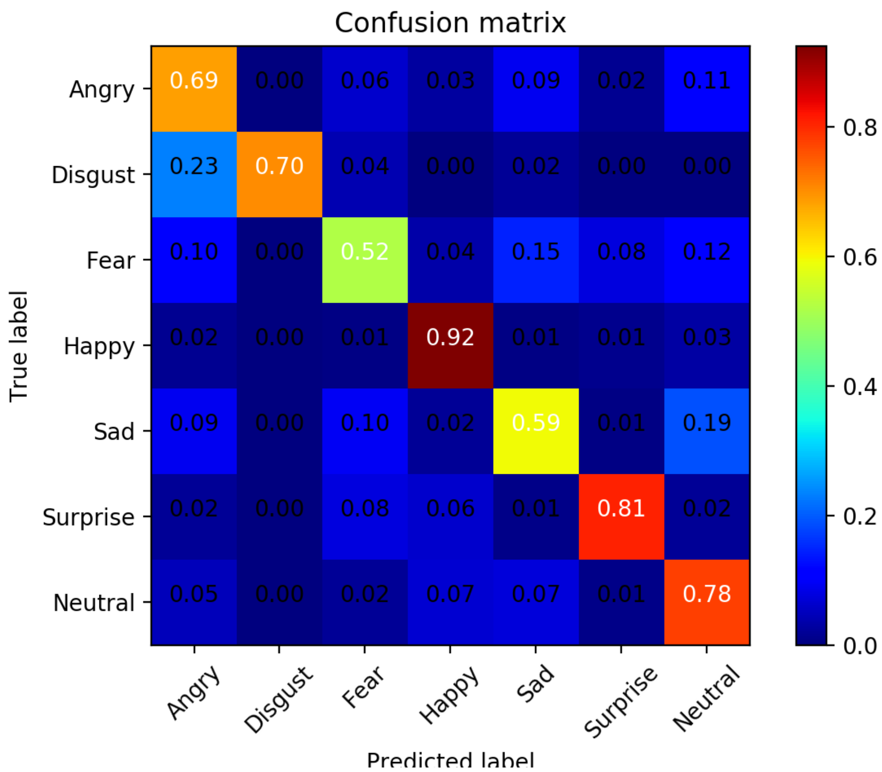
Total params: 6,048,263
Trainable params: 6,042,119
Non-trainable params: 6,144



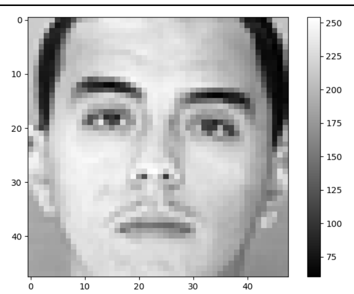
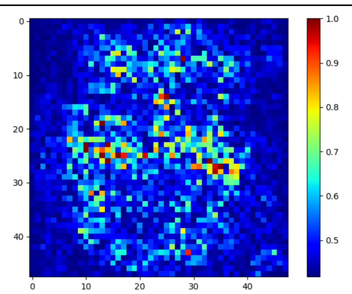
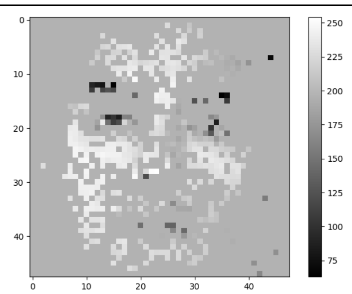


3. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析]

(Collaborators:)
答：從 confusion matrix 可以看出 Fear 和 Sad 分得比較不好比較容易混淆。Angry 和 Disgust 也容易被分錯，我認為的原因是這幾個表情比較接近很難用一個 label 去表達。



4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？
(Collaborators: 謝朋諺)
答：

原圖	Saliency Map	Mask 掉 heat 小的部份
		

從 Saliency Map 中可以看出模型在做 classification 時主要 focus 在眼睛和臉頰的部分，在嘴巴和鼻子的部分也有少量 focus，在嘴巴和鼻子部分比想像中 focus 得要少，可能的原因的是 model 還不夠準確。

5. (1%) 承(1)(2)，利用上課所提到的 gradient ascent 方法，觀察特定層的 filter 最容易被哪種圖片 activate。
(Collaborators: 謝朋諺)
答：通過 gradient ascent 畫出的 filter 圖片中觀察可以看出在第三層 conv2d 中，最容意 activate filter 的圖片為一些較粗的線條，他們應該是負責抓取臉部輪廓的特徵，所以大部分為一些斜條文。而一些橫條紋應該是負責抓取眼睛、嘴巴和眉毛的特徵。可見在第三層的 conv2d 中 model 已經有一定的能力可以抓取臉部特徵了。