HW3 Report

學號：R05945012 系級：生醫碩二 姓名：張凱崴

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？

(1) 模型架構：

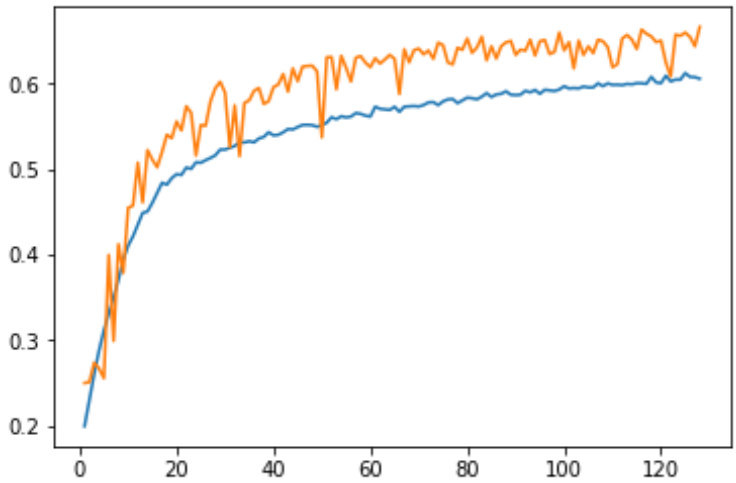
Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 64)	1664
batch_normalization_1 (Batch	(None, 48, 48, 64)	256
max_pooling2d_1 (MaxPooling2	(None, 24, 24, 64)	0
dropout_1 (Dropout)	(None, 24, 24, 64)	0
conv2d_2 (Conv2D)	(None, 24, 24, 128)	73856
batch_normalization_2 (Batch	(None, 24, 24, 128)	512
max_pooling2d_2 (MaxPooling2	(None, 12, 12, 128)	0
dropout_2 (Dropout)	(None, 12, 12, 128)	0
conv2d_3 (Conv2D)	(None, 12, 12, 256)	295168
batch_normalization_3 (Batch	(None, 12, 12, 256)	1024
max_pooling2d_3 (MaxPooling2	(None, 6, 6, 256)	0
dropout_3 (Dropout)	(None, 6, 6, 256)	0
conv2d_4 (Conv2D)	(None, 6, 6, 512)	1180160
batch_normalization_4 (Batch	(None, 6, 6, 512)	2048
max_pooling2d_4 (MaxPooling2	(None, 3, 3, 512)	0
dropout_4 (Dropout)	(None, 3, 3, 512)	0
flatten_1 (Flatten)	(None, 4608)	0
dense_1 (Dense)	(None, 1024)	4719616
batch_normalization_5 (Batch	(None, 1024)	4096
dropout_5 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
batch_normalization_6 (Batch	(None, 1024)	4096
dropout_6 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 7)	7175

Total params: 7,339,271
Trainable params: 7,333,255
Non-trainable params: 6,016

共有四層 convolution layer 和三層 fully-connected layer。每層 conv layer 的 activation 都是 relu，並經過 batch normalization、max pooling 和 dropout，filter 數目由 64 漸增至 512，dropout rate 則由 0.2 漸增至 0.5。Fully-connected layer 則由兩層 1024 units 的 relu 以及最後一層 softmax 組成，並經過 batch normalization 和 dropout (rate=0.5)。

(2) 訓練過程：(橘：valid、藍：train、橫軸：epoch、縱軸：accuracy)

最佳 model 的 valid accuracy 為 0.6659，kaggle 的 private score 為 0.65589。



(因為在 kaggle 截止後才 train 出這個 model，所以此分數比 kaggle 顯示的最高分還高，僅為了撰寫報告才選擇此較佳的 model，程式方面還是會重現正確的 kaggle 成績。)

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？

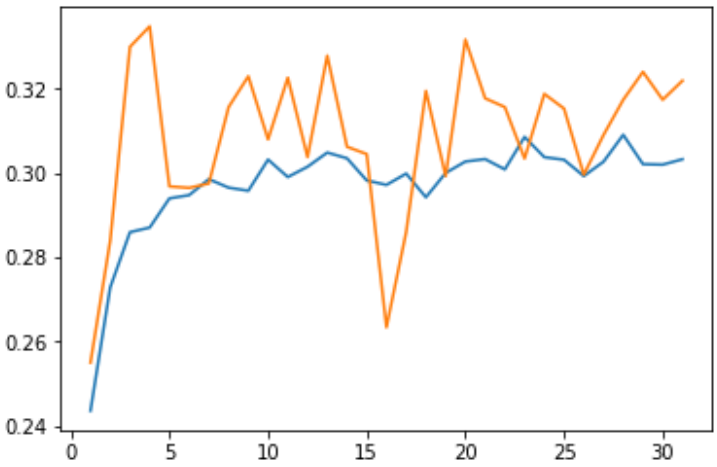
(1) 模型架構：

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 1024)	2360320
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
dropout_2 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 1024)	1049600
dropout_3 (Dropout)	(None, 1024)	0
dense_4 (Dense)	(None, 1024)	1049600
dropout_4 (Dropout)	(None, 1024)	0
dense_5 (Dense)	(None, 1024)	1049600
dropout_5 (Dropout)	(None, 1024)	0
dense_6 (Dense)	(None, 1024)	1049600
dropout_6 (Dropout)	(None, 1024)	0
dense_7 (Dense)	(None, 7)	7175

Total params: 7,615,495
Trainable params: 7,615,495
Non-trainable params: 0

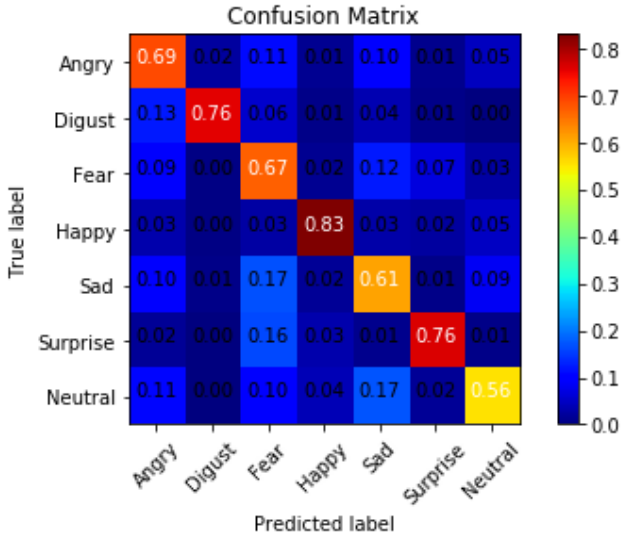
共七層 fully-connected layer，包含六層 1024units 的 relu 和一層 softmax。與 Q1 相同，共有七百萬左右個 parameters。

(2) 訓練過程：(橘：valid、藍：train、橫軸：epoch、縱軸：accuracy)



Train 與 valid accuracy 皆為 0.3 左右，且震盪幅度很大，即便調整 dropout 也無法讓 accuracy(無論是 train 還是 valid)上升。這是因為僅用 DNN 做訓練的話，只考慮每個單一像素的權重，會無法將相鄰像素之間的關係考慮在內，因此無法將影像具有的特徵訓練出來，而且無法考慮到影像平移縮放的特性，因此對準確率的估計震盪很大(因為隨機選擇到的 sample 對結果有很大的影響)。

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]



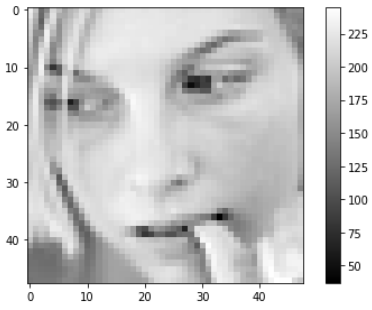
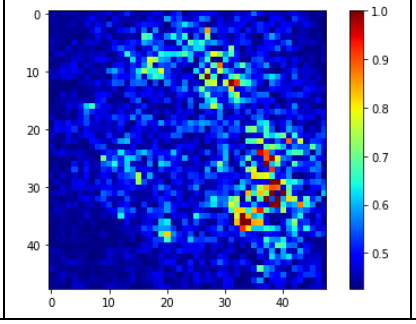
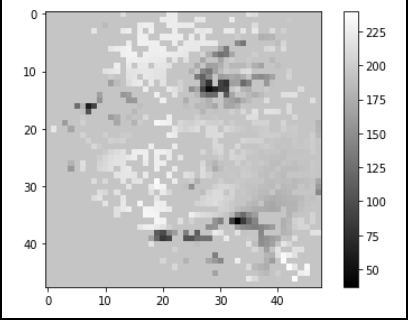
由上圖可觀察到，angry、fear、sad 三種 class 容易互相搞混。可能是因為這三種情緒在五官上表現的特徵相近，實際觀察圖片後發現有些圖片用人眼也無法很好地分辨出來。

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？

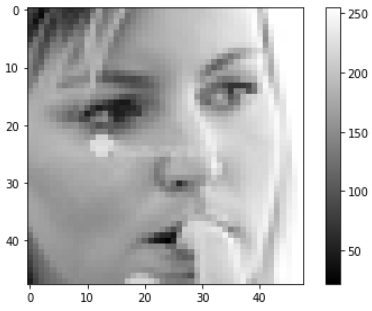
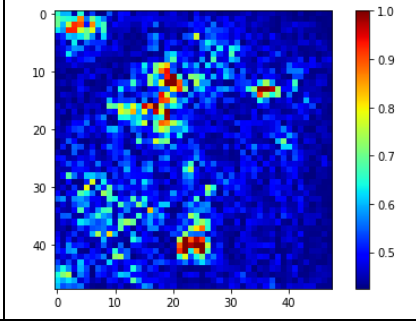
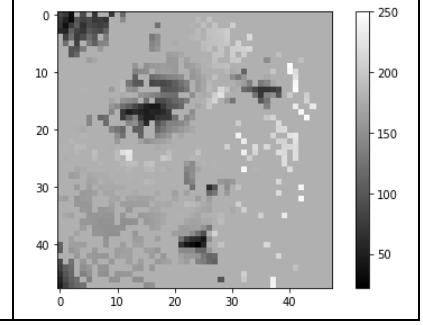
Ans:

我故意選擇原圖相似，一張預測正確另一張卻預測錯誤的兩張圖做比對。兩張圖的 saliency map 主要都 focus 在眼睛、眉毛和嘴巴的輪廓，且頭髮部分都被 mask 掉，代表我的 CNN 模型是以這些特徵當作預測的標準，所以這兩張圖構圖相似的圖被預測的結果是相同的(2.恐懼)。同時這也可以理解第一張圖預測錯誤的原因，因為模型並沒有抓到足以分辨這兩張圖的特徵差異。(即便以人眼，第一張圖也是很難判斷正確的。)

Index: 1234, Truth: 4, Prediction: 2

原圖	Saliency Map	Masked out
		

Index: 2345, Truth: 2, Prediction: 2

原圖	Saliency Map	Masked out
		

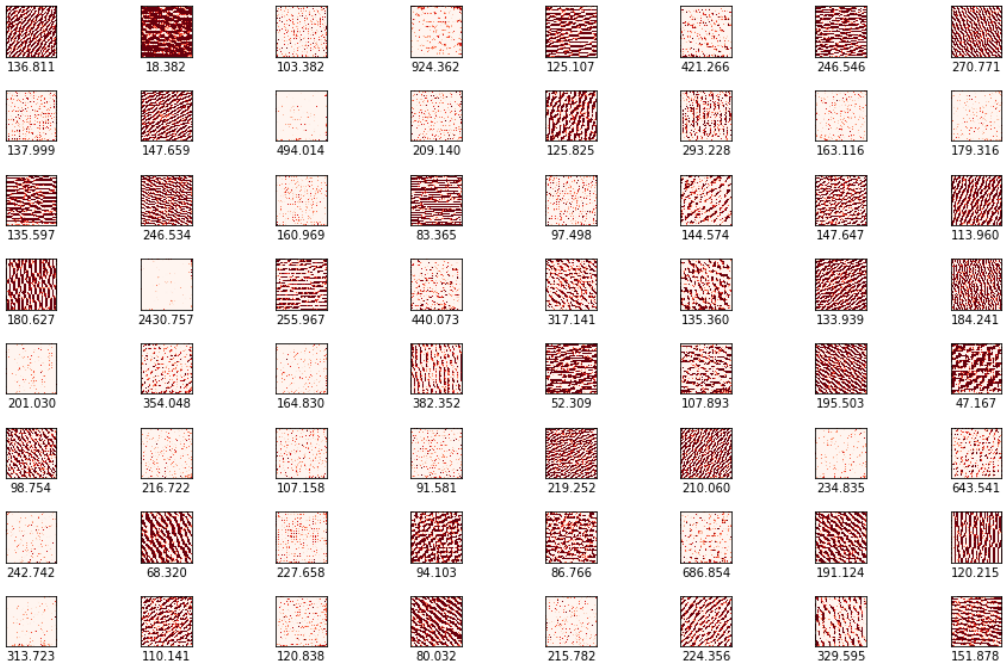
5. (1%) 承(1)(2)，利用上課所提到的 gradient ascent 方法，觀察特定層的 filter 最容易被哪種圖片 activate。

Ans:

結果顯示在下頁。

我觀察的是第一層 convolution layer 的所有 64 個 filter，並將兩張圖片分別輸入並觀察 output。我們可以很明顯地觀察到各別 filter 所具有的特性，例如：filter 6,12 是眼睛與嘴巴等五官、filter 24,31 是臉型等。

Filters of layer maxpooling2d_1 (# Ascent Epoch 128)



Output (Given image 2345)

Output (Given image 2000)