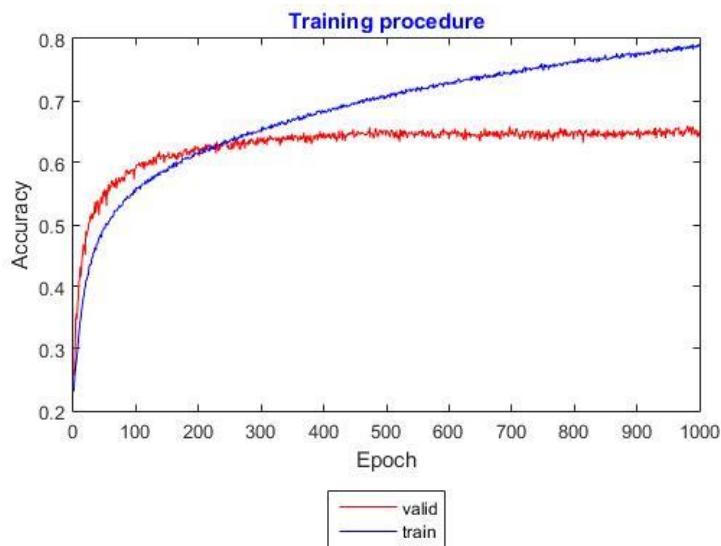


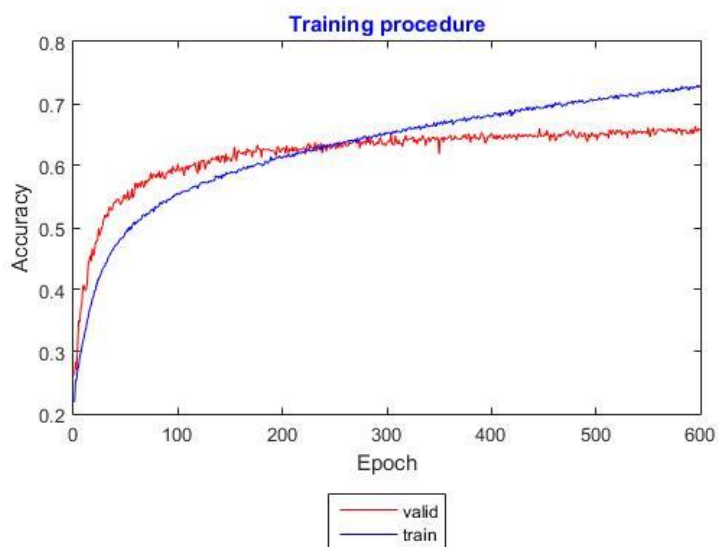
1. 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？

答:

下面圖一是 CNN 的架構，一開始對 feature 先進行 normalization 然後再進入模型，batch size 為 128，我一開始使用兩層的 convolution layer，使用 MaxPooling，跑出來的 accuracy 只有 0.4 多，所以我就再繼續往上加，可是越我上加會跑比較慢，最後我使用五層 convolution layer 和兩層的 fully connect，並把部分改成 PreLU，原本我的 dropout 設定成 0.8，但準確率不高，後來我設定成 0.5，會有比較好點，後來發現在 0.2~0.3 會有比較好的 accuracy，然後 epoch 越多，accuracy 會越好，下圖(二)以及圖(三)分別是 1000 個 epoch 和 600 個 epoch



圖二



圖三

conv2d_1 (Conv2D)	(None, 44, 44, 64)	1664
p_re_lu_1 (PReLU)	(None, 44, 44, 64)	123904
zero_padding2d_1 (ZeroPaddin	(None, 48, 48, 64)	0
max_pooling2d_1 (MaxPooling2	(None, 22, 22, 64)	0
zero_padding2d_2 (ZeroPaddin	(None, 24, 24, 64)	0
conv2d_2 (Conv2D)	(None, 22, 22, 64)	36928
p_re_lu_2 (PReLU)	(None, 22, 22, 64)	30976
zero_padding2d_3 (ZeroPaddin	(None, 24, 24, 64)	0
conv2d_3 (Conv2D)	(None, 22, 22, 64)	36928
p_re_lu_3 (PReLU)	(None, 22, 22, 64)	30976
average_pooling2d_1 (Average	(None, 10, 10, 64)	0
zero_padding2d_4 (ZeroPaddin	(None, 12, 12, 64)	0
conv2d_4 (Conv2D)	(None, 10, 10, 128)	73856
p_re_lu_4 (PReLU)	(None, 10, 10, 128)	12800
zero_padding2d_5 (ZeroPaddin	(None, 12, 12, 128)	0
conv2d_5 (Conv2D)	(None, 10, 10, 128)	147584
p_re_lu_5 (PReLU)	(None, 10, 10, 128)	12800
zero_padding2d_6 (ZeroPaddin	(None, 12, 12, 128)	0
average_pooling2d_2 (Average	(None, 5, 5, 128)	0
flatten_1 (Flatten)	(None, 3200)	0
dense_1 (Dense)	(None, 1024)	3277824
p_re_lu_6 (PReLU)	(None, 1024)	1024
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
p_re_lu_7 (PReLU)	(None, 1024)	1024
# Data augmentation		

2. 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？

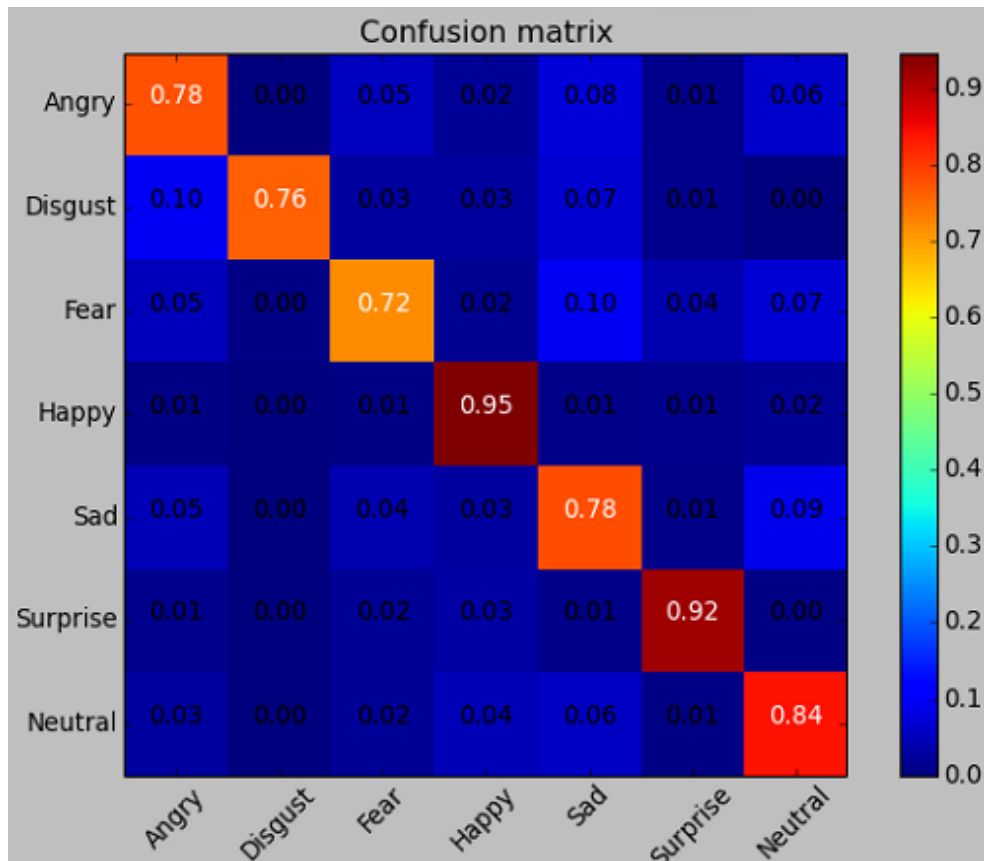
答：

DNN 的架構是由 CNN 最後兩層擴展出來的，把 convolution layer 層拿掉，其他也是一樣使用 PReLU，dropout 設為 0.1，learning rate 設成 0.1，最後訓練發現，不管如何調 Dropout，或是多加幾層都沒有變好，可能是 DNN 的架構因為沒有 convolution filter 進行分類，所以比現得不太理想，CNN 在圖像和語音辨識方面有更優的結果

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 512)	1180160
p_re_lu_1 (PReLU)	(None, 512)	512
dropout_1 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 512)	262656
p_re_lu_2 (PReLU)	(None, 512)	512
dropout_2 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 512)	262656
p_re_lu_3 (PReLU)	(None, 512)	512
dropout_3 (Dropout)	(None, 512)	0
dense_4 (Dense)	(None, 7)	3591
activation_1 (Activation)	(None, 7)	0

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

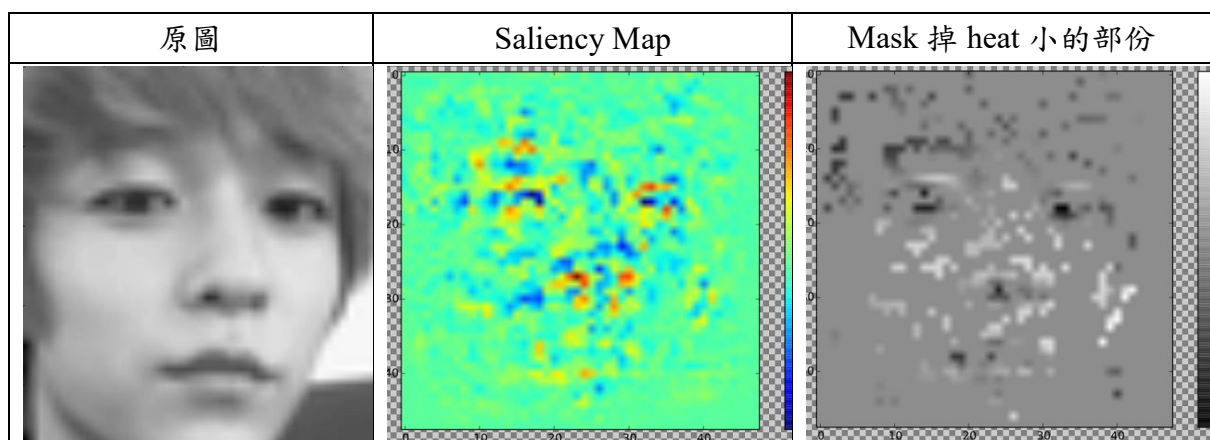
最容易被搞混的應該是 fear，fear 會有部分被認為是 sad，但 sad 卻不太會被認為是 fear，再來是 angry 和 disgust，有些 disgust 會被判斷成為 angry，但卻沒有 angry 被判斷成 disgust，這樣的結果讓我很意外，最後我其實認為 happy 和 surprise，其實是蠻難分辨的，可是不知道為甚麼最後出來的結果，卻比我覺得要好很多。。



4. 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？

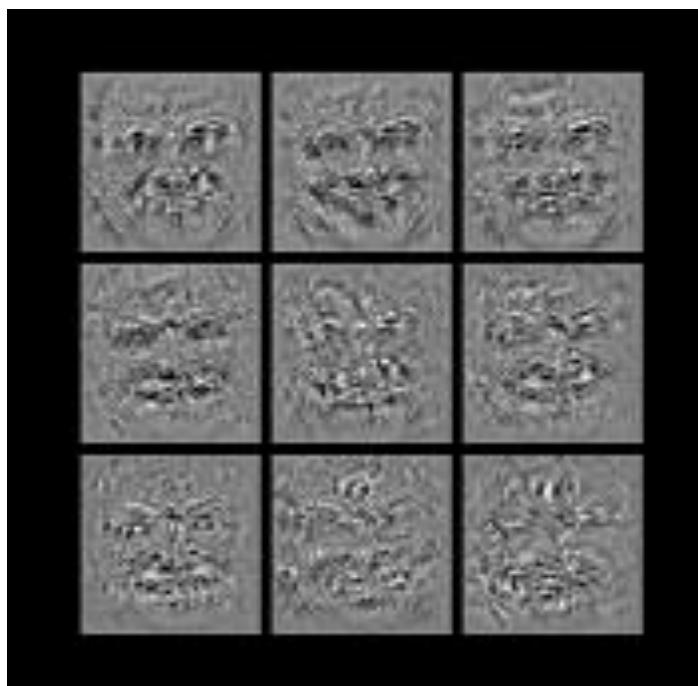
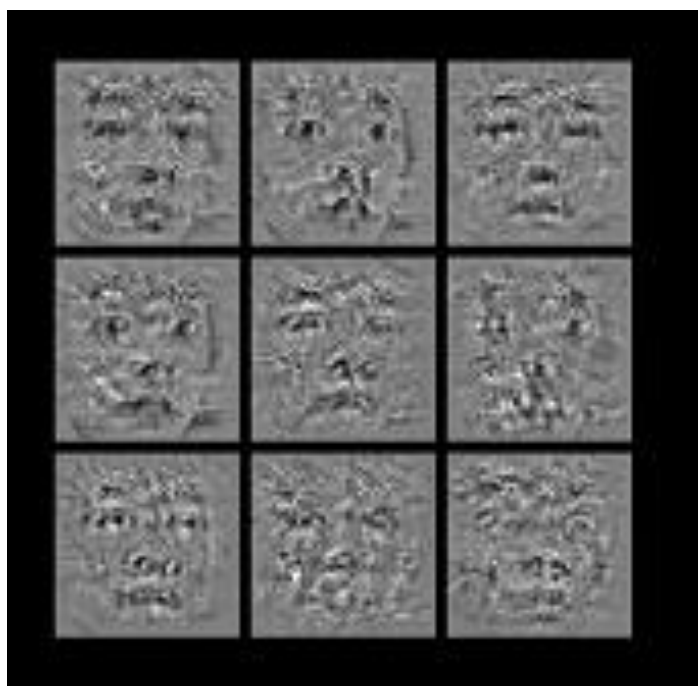
答:

從 Mask 掉 heat 小的部份的這張圖可以看到眼睛、鼻子，還有部分人臉的輪廓以及嘴巴，所以我認為他是 focus 在眼睛、鼻子、臉的輪廓為最重要的部分。



5. 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。

我覺得這些 filter 感覺最容易被類似眼睛和鼻子的圖片來 activate，可以看道出其實還是可以看到一些眼睛和鼻子的圖案，我覺得應該就像我們人類一樣，也幾乎是用眼睛和鼻子來判斷不同的人。



可以看到上面第一排還可以看到一些輪廓，但越往下一排就越不清楚，到了最後一排



幾乎看不出來人臉。

