學號：B03705013 系級： 資管三 姓名：徐逸然

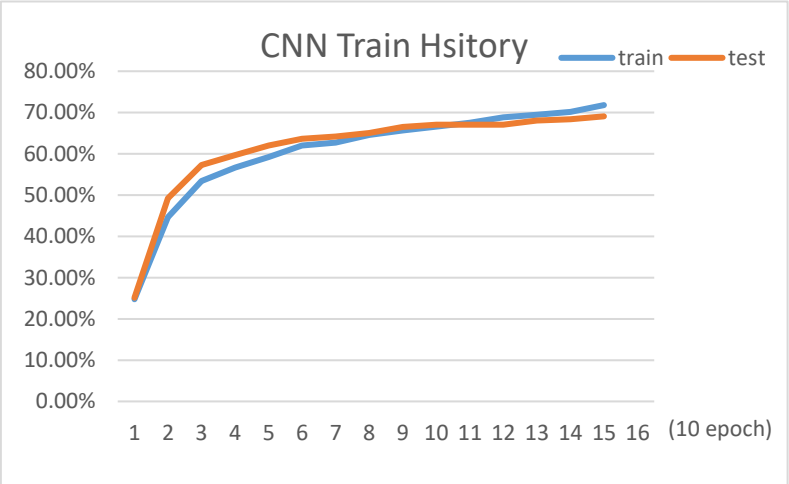
1. 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？

架構為以下 ➔

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 48, 48)	320
dropout_1 (Dropout)	(None, 32, 48, 48)	0
conv2d_2 (Conv2D)	(None, 64, 48, 48)	18496
dropout_2 (Dropout)	(None, 64, 48, 48)	0
max_pooling2d_1 (MaxPooling2	(None, 64, 24, 24)	0
conv2d_3 (Conv2D)	(None, 64, 24, 24)	36928
dropout_3 (Dropout)	(None, 64, 24, 24)	0
conv2d_4 (Conv2D)	(None, 128, 24, 24)	73856
dropout_4 (Dropout)	(None, 128, 24, 24)	0
max_pooling2d_2 (MaxPooling2	(None, 128, 12, 12)	0
conv2d_5 (Conv2D)	(None, 128, 12, 12)	147584
dropout_5 (Dropout)	(None, 128, 12, 12)	0
conv2d_6 (Conv2D)	(None, 256, 12, 12)	295168
dropout_6 (Dropout)	(None, 256, 12, 12)	0
max_pooling2d_3 (MaxPooling2	(None, 256, 6, 6)	0
flatten_1 (Flatten)	(None, 9216)	0
dropout_7 (Dropout)	(None, 9216)	0
dense_1 (Dense)	(None, 1024)	9438208
dropout_8 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 7)	7175

Total params: 10,017,735
Trainable params: 10,017,735
Non-trainable params: 0

訓練過程 ➔



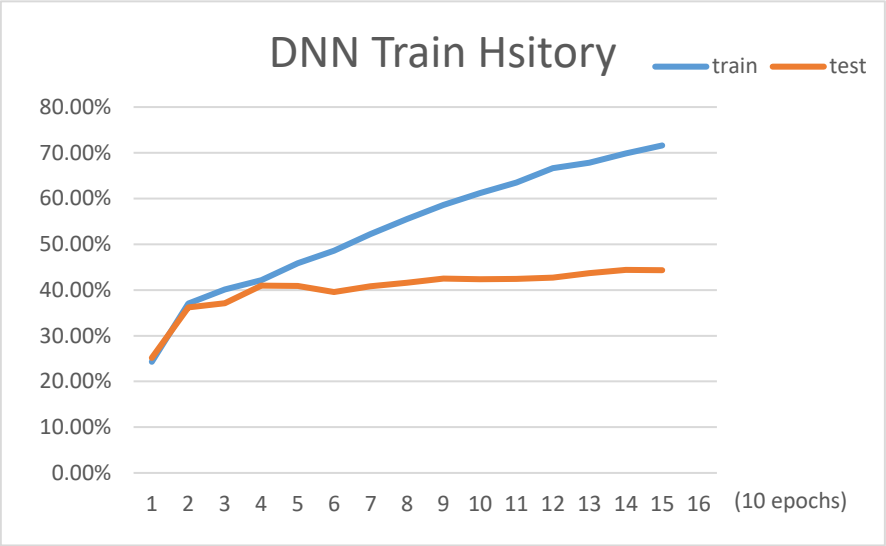
2. 承上題，請用與上述 CNN 接近的參數量，實做簡單的 DNN model。其模型架構、訓練過程和準確率為何？試與上題結果做比較，並說明你觀察到了什麼？

架構為以下 ➔

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

訓練過程 ➔



DNN model 有 700 多萬個參數，然而 test error 大約 40% 上下就很難再提升，即使已經頻繁使用 Dropout 也一樣。CNN 由於能夠偵測重點 feature 如眼睛、嘴巴等 ，故能達到更高的準確率。

3. 觀察答錯的圖片中，哪些 class 彼此間容易用混？

Confusion Matrix ➔

			Prediction					Total
	245	1	32	9	50	11	52	400
	11	20	0	2	6	3	2	44
	44	1	203	16	63	46	37	410
Actual	12	0	3	656	9	12	30	722
	50	0	34	14	281	4	100	483
	12	0	20	19	6	252	9	318
	18	0	17	36	76	5	345	497

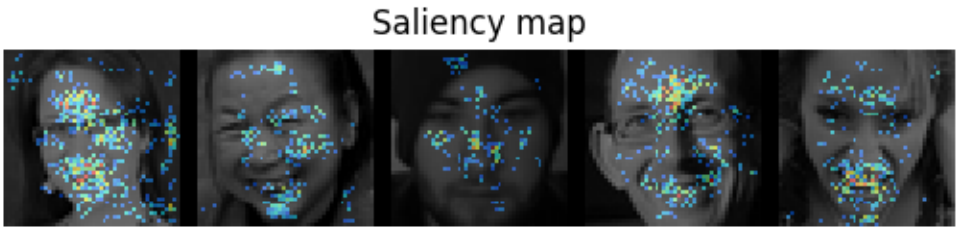
由 2000 多個 validation data 做出來的結果可以得出：

Class 0 最容易被分類錯至 Class 6
Class 1 最容易被分類錯至 Class 0
Class 2 最容易被分類錯至 Class 4
Class 3 最容易被分類錯至 Class 6
Class 4 最容易被分類錯至 Class 6
Class 5 最容易被分類錯至 Class 2
Class 6 最容易被分類錯至 Class 4

➔ Class 6(中立) 與 Class 4(難過) 最容易被搞混

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？

   使用內建 visualize_saliency 時大部分的圖會出現詭異了藍色 bug，因此我只挑出幾張正常的來做分析。從下面五張人臉能發現我的 model 會針對雙眼、嘴、額頭以及臉頰做辨識。是一個直觀且合理的結果，畢竟大部分的表情都能藉由這些器官觀察出來。



5. 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:

   我繪出第四層 Convolution2D 的結果，從 128 個 filter 裡取 64 個 filter 出來觀察，顯然機器眼中的世界跟我們不太一樣。我推測由於表情屬於較細緻的判斷，因此此層的每個 filter 看不太出甚麼大方向，反而比較類似各種紋理。