

學號：B03701221 系級： 工管四 姓名：王逸庭

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: )

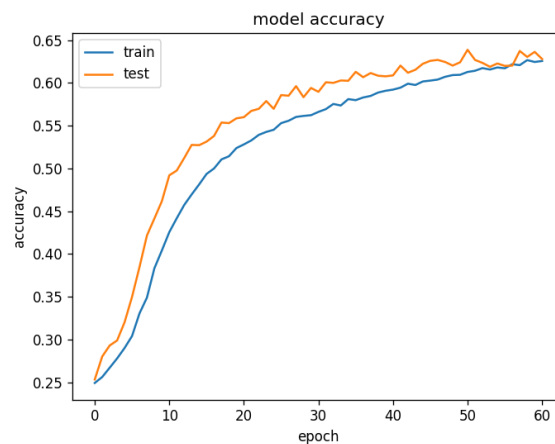
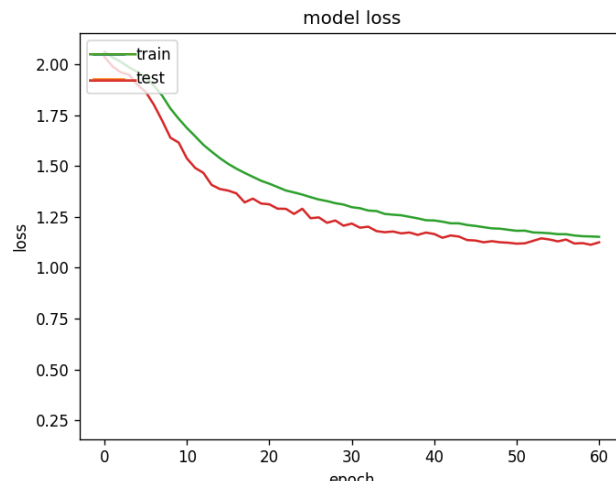
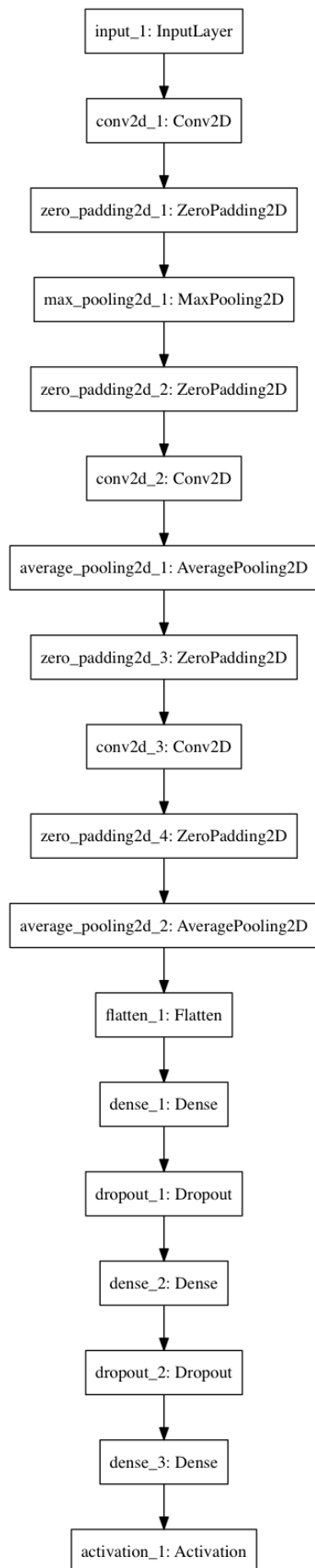
答：

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 48, 48, 1)	0
conv2d_1 (Conv2D)	(None, 44, 44, 32)	832
zero_padding2d_1 (ZeroPaddin	(None, 48, 48, 32)	0
max_pooling2d_1 (MaxPooling2	(None, 46, 46, 32)	0
zero_padding2d_2 (ZeroPaddin	(None, 48, 48, 32)	0
conv2d_2 (Conv2D)	(None, 46, 46, 64)	18496
average_pooling2d_1 (Average	(None, 22, 22, 64)	0
zero_padding2d_3 (ZeroPaddin	(None, 24, 24, 64)	0
conv2d_3 (Conv2D)	(None, 22, 22, 128)	73856
zero_padding2d_4 (ZeroPaddin	(None, 24, 24, 128)	0
average_pooling2d_2 (Average	(None, 11, 11, 128)	0
flatten_1 (Flatten)	(None, 15488)	0
dense_1 (Dense)	(None, 1024)	15860736
dropout_1 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 512)	524800
dropout_2 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 7)	3591
activation_1 (Activation)	(None, 7)	0
Total params: 16,482,311		
Trainable params: 16,482,311		
Non-trainable params: 0		

此模型有使用 early stopping：停在 61epoch，而 val\_acc :0.638675958105

Public / Private accuracy:大致分佈在 0.63

<b>hw3_kfold_7.1.csv</b> 21 hours ago by <a href="#">kelly wang</a> <a href="#">add submission details</a>	0.63165	0.63025	
<b>hw3_kfold_7.1.1.csv</b> a day ago by <a href="#">kelly wang</a> <a href="#">add submission details</a>	0.63248	0.63415	



1.一開始我先透過 keras ImageDataGenerator 做 data augmentation，透過旋轉平移增加 training 的資料數

2.我操作了 1 個 Conv2D 1 個 Maxpooling，而後也是 1 個 Conv2D 配一個 Avgpooling，連續做兩次

3. Flatten 攤平後，接續兩個 Dense 層數分別是 1024、512，並在中間都有進行 dropout，分別是 0.4、0.5，使用 SGD 作為優化學習的方法

從上圖可以發現 training 的 acc 或是 loss 相對 val 都表現較差，這和一般狀況不一樣，我猜測因為我的參數疊不夠多，同時也有進行 early stopping，所

以沒有產生 training data 表現特別好 val data 表現普通的狀況。

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: )

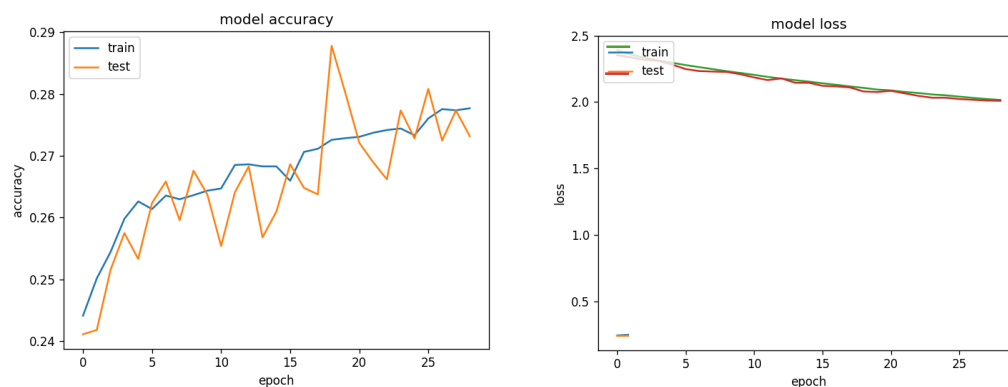
答：

上題的總參數約 16482311，於是我用了以下的模型，總參數大約是 15000000，兩模型參數量接近。

Layer (type)	Output Shape	Param #
input_19 (InputLayer)	(None, 48, 48, 1)	0
flatten_12 (Flatten)	(None, 2304)	0
dense_67 (Dense)	(None, 1024)	2360320
dropout_49 (Dropout)	(None, 1024)	0
dense_68 (Dense)	(None, 4096)	4198400
dropout_50 (Dropout)	(None, 4096)	0
dense_69 (Dense)	(None, 2048)	8390656
dropout_51 (Dropout)	(None, 2048)	0
dense_70 (Dense)	(None, 7)	14343
activation_18 (Activation)	(None, 7)	0
Total params: 14,963,719		
Trainable params: 14,963,719		
Non-trainable params: 0		

關於此模型我也有使用 Stop by early stopping，停在 Epoch 29，val 準確率 是 0.2878048780280123

2. 如上題模型我也有先做 data augmentation，增加 training data 數
3. 先將資料 Flatten 攤平後，疊上了三層 Dense 分別是 1024 4096 2048 nodes，中間也有做 dropout，分別是 0.3 0.5 0.5，使用 SGD 作為優化學習的方法

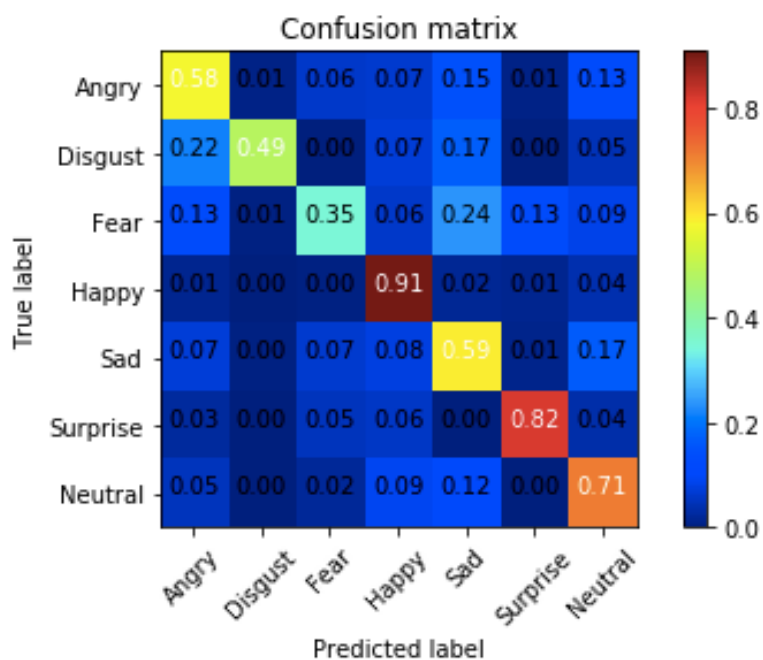


在同樣的總參數下，DNN 的準確率相對於 CNN 的確表現較差，但一次訓練時間則是 DNN 較短。

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：



sad fear disgust angry 這四個是準確率較低

angry 容易和 sad 和 neutral 搞混，而 disgust 則和 angry sad 搞混

fear 則是和 sad 和 surprise，而 sad 則是和 neutral

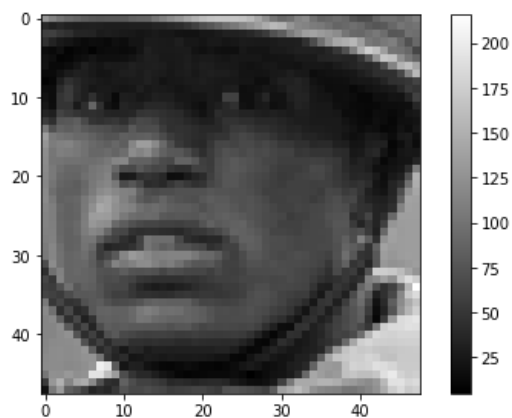
而其實看過訓練資料會發現 disgust 的資料明顯較少，這可能導致 disgust 的準確率較差的原因

```
.
  0   1   2   3   4   5   6
3995 436 4097 7215 4830 3171 4965
```

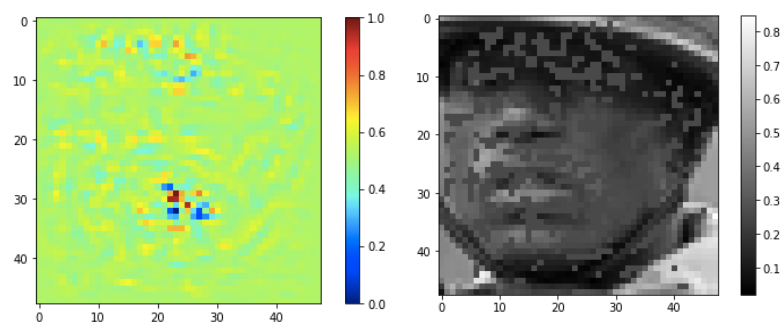
4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？  
(Collaborators: )

答：

原圖：



saliency maps    mask 一小部分的地方



從圖中可以發現 **focus** 在嘴巴的部分，眼睛部分也有一些。其他地方都非常中性（0.4~0.6）

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: )

答：