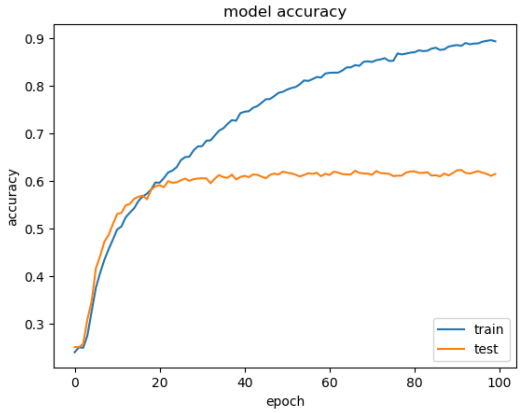
學號：R04546031 系級： 工工碩二 姓名：洪唯凱

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：本模型已經出現 overfitting，可能可以做資料擴增或是再架深一點，在做資料的觀察時，發現第二類明顯很少，和最後一類很難辨別，可以在做 cost 權重的調整

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	(None, 48, 48, 1)	0
conv2d_1 (Conv2D)	(None, 44, 44, 64)	1664
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
zero_padding2d_3 (ZeroPaddin	(None, 24, 24, 64)	0
conv2d_3 (Conv2D)	(None, 22, 22, 64)	36928
average_pooling2d_1 (Average	(None, 10, 10, 64)	0
zero_padding2d_4 (ZeroPaddin	(None, 12, 12, 64)	0
dropout_1 (Dropout)	(None, 12, 12, 64)	0
conv2d_4 (Conv2D)	(None, 10, 10, 128)	73856
zero_padding2d_5 (ZeroPaddin	(None, 12, 12, 128)	0
dropout_2 (Dropout)	(None, 12, 12, 128)	0
conv2d_5 (Conv2D)	(None, 10, 10, 128)	147584
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
dropout_3 (Dropout)	(None, 1024)	0
dense_2 (Dense)	(None, 1024)	1049600
dropout_4 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 7)	7175
activation_1 (Activation)	(None, 7)	0

Total params: 4,631,559
Trainable params: 4,631,559
Non-trainable params: 0

來自 <http://localhost:8888/notebooks/Desktop/hw3/Git/Q3.ipynb>



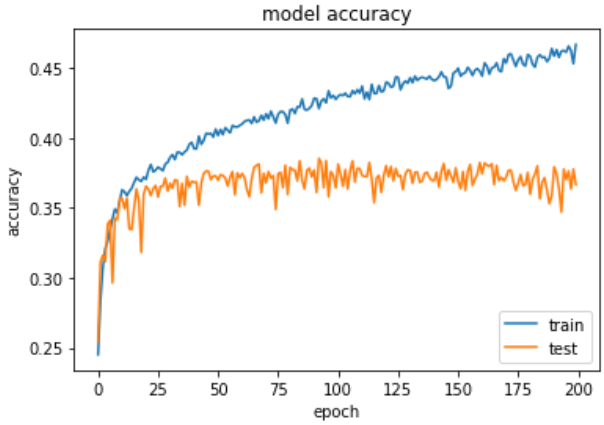
loss: 2.4348 - acc: 0.8943 - val_loss: 13.5134 - val_acc: 0.6155

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：本模型在 training 和 testing 上都表現得很差

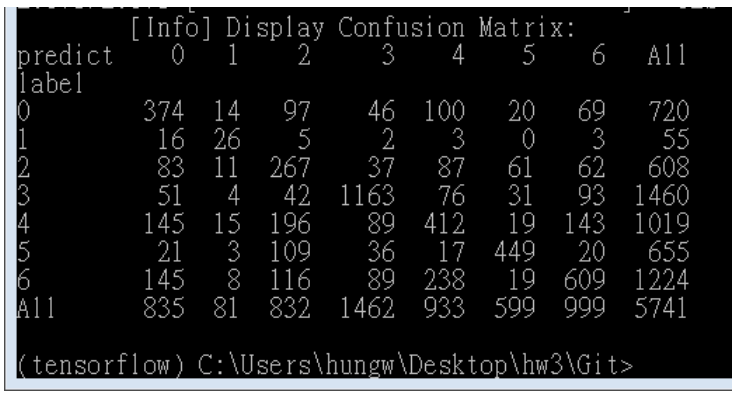
```
loss: 1.4275 - acc: 0.4670 - val_loss: 1.6539 - val_acc: 0.3668
```

Layer (type)	Output Shape	Param #
dense_64 (Dense)	(None, 256)	590080
dense_65 (Dense)	(None, 1024)	263168
dense_66 (Dense)	(None, 1024)	1049600
dropout_20 (Dropout)	(None, 1024)	0
dense_67 (Dense)	(None, 1024)	1049600
dropout_21 (Dropout)	(None, 1024)	0
dense_68 (Dense)	(None, 1024)	1049600
dropout_22 (Dropout)	(None, 1024)	0
dense_69 (Dense)	(None, 512)	524800
dropout_23 (Dropout)	(None, 512)	0
dense_70 (Dense)	(None, 7)	3591
activation_10 (Activation)	(None, 7)	0

Total params: 4,530,439
Trainable params: 4,530,439
Non-trainable params: 0



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：很容易分到第六類

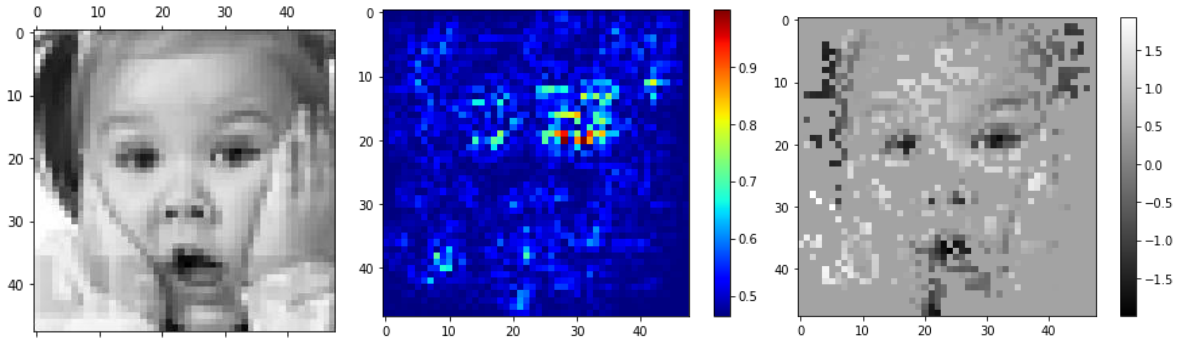


[Info] Display Confusion Matrix:

predict	0	1	2	3	4	5	6	All
label								
0	374	14	97	46	100	20	69	720
1	16	26	5	2	3	0	3	55
2	83	11	267	37	87	61	62	608
3	51	4	42	1163	76	31	93	1460
4	145	15	196	89	412	19	143	1019
5	21	3	109	36	17	449	20	655
6	145	8	116	89	238	19	609	1224
All	835	81	832	1462	933	599	999	5741

(tensorflow) C:\Users\hungw\Desktop\hw3\Git>

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: 林冠廷、洪紹綺、張少豪)
答：嘴巴和眼睛



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: 林冠廷、洪紹綺、張少豪)
答：人臉的五官比較會被 activate