

學號：R05546014 系級：工工碩二 姓名：洪紹綺

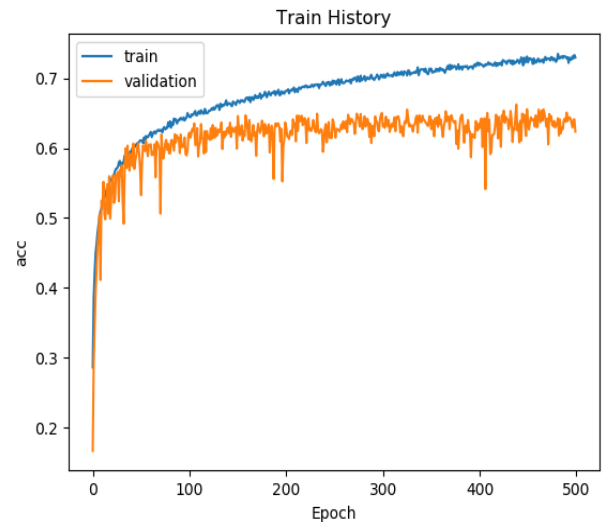
註:四到五題都取#777 圖片

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 #
conv2d_1 (Conv2D)	(None, 48, 48, 64)	1664
batch_normalization_1 (Batch Normalization)	(None, 48, 48, 64)	256
leaky_re_lu_1 (LeakyReLU)	(None, 48, 48, 64)	0
zero_padding2d_1 (ZeroPadding2D)	(None, 52, 52, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 24, 24, 64)	0
zero_padding2d_2 (ZeroPadding2D)	(None, 26, 26, 64)	0
conv2d_2 (Conv2D)	(None, 24, 24, 128)	73856
batch_normalization_2 (Batch Normalization)	(None, 24, 24, 128)	512
leaky_re_lu_2 (LeakyReLU)	(None, 24, 24, 128)	0
zero_padding2d_3 (ZeroPadding2D)	(None, 26, 26, 128)	0
dropout_1 (Dropout)	(None, 26, 26, 128)	0
conv2d_3 (Conv2D)	(None, 24, 24, 64)	73792
batch_normalization_3 (Batch Normalization)	(None, 24, 24, 64)	256
leaky_re_lu_3 (LeakyReLU)	(None, 24, 24, 64)	0
average_pooling2d_1 (AveragePooling2D)	(None, 11, 11, 64)	0
zero_padding2d_4 (ZeroPadding2D)	(None, 13, 13, 64)	0
conv2d_4 (Conv2D)	(None, 11, 11, 128)	73856
batch_normalization_4 (Batch Normalization)	(None, 11, 11, 128)	512
leaky_re_lu_4 (LeakyReLU)	(None, 11, 11, 128)	0
max_pooling2d_2 (MaxPooling2D)	(None, 5, 5, 128)	0
zero_padding2d_5 (ZeroPadding2D)	(None, 7, 7, 128)	0
dropout_2 (Dropout)	(None, 7, 7, 128)	0
conv2d_5 (Conv2D)	(None, 5, 5, 64)	73792
batch_normalization_5 (Batch Normalization)	(None, 5, 5, 64)	256
leaky_re_lu_5 (LeakyReLU)	(None, 5, 5, 64)	0
zero_padding2d_6 (ZeroPadding2D)	(None, 7, 7, 64)	0
average_pooling2d_2 (AveragePooling2D)	(None, 3, 3, 64)	0
flatten_1 (Flatten)	(None, 576)	0
dense_1 (Dense)	(None, 1024)	590848
dense_2 (Dense)	(None, 512)	524800
dropout_3 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 1024)	525312
dropout_4 (Dropout)	(None, 1024)	0
dense_4 (Dense)	(None, 7)	7175
activation_1 (Activation)	(None, 7)	0
Total params: 1,946,887		
Trainable params: 1,945,991		
Non-trainable params: 896		



Train 的準確率大概在 0.7 左右，持續上升，但 validation 在 0.6 左右擺盪了 400 個迭代左右。

Kaggle 準確率: $(0.65784 + 0.65032) / 2 = 0.65408$

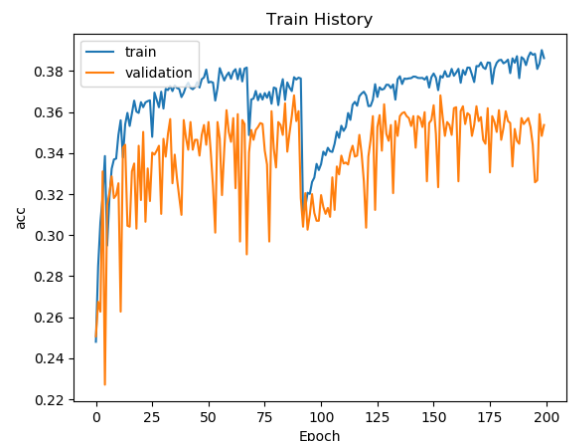
2. (1%) 承上題，請用與上述 CNN 接近的參數量，實做簡單的 DNN model。其模型架構、訓練過程和準確率為何？試與上題結果做比較，並說明你觀察到了什麼？

(Collaborators:)

答：DNN 參數雖較 CNN 多，但準確率極差。

DNN 模型比 CNN 略多 1% 參數，Train 跟 Validation 準確率在 100 個迭代間在 0.35 上下擺盪停滯。

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 256)	590080
dense_2 (Dense)	(None, 256)	65792
dense_3 (Dense)	(None, 1024)	263168
dense_4 (Dense)	(None, 512)	524800
dropout_1 (Dropout)	(None, 512)	0
dense_5 (Dense)	(None, 1024)	525312
dropout_2 (Dropout)	(None, 1024)	0
dense_6 (Dense)	(None, 7)	7175
activation_1 (Activation)	(None, 7)	0
Total params: 1,976,327		
Trainable params: 1,976,327		
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：從矩陣可以看出生氣、難過與恐懼有些會混淆，符合現實人類表情狀況；驚訝與恐懼之間有時也會混淆，難過與無表情也會混淆。

Angry	Hate	Fear	Joy	Sad	Surprise	Neutral
-------	------	------	-----	-----	----------	---------

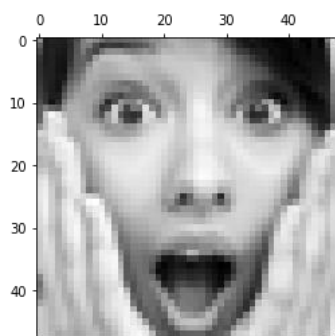
[Info] Display Confusion Matrix:								
predict label	0	1	2	3	4	5	6	All
0	162	4	<u>35</u>	14	<u>48</u>	16	<u>36</u>	315
1	4	15	3	0	1	1	0	24
2	<u>33</u>	1	133	9	<u>73</u>	<u>32</u>	<u>37</u>	318
3	16	1	13	424	20	11	26	511
4	<u>41</u>	1	30	12	190	2	<u>51</u>	327
5	8	0	17	14	3	187	7	236
6	23	0	20	27	<u>49</u>	8	220	347
All	287	22	251	500	384	257	377	2078

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？

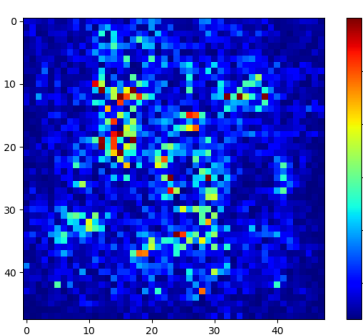
(Collaborators: 林冠廷、洪唯凱、張少豪)

答：從圖中可以看出，模型分類的時候主要是抓眼睛跟嘴巴，反應較明顯。

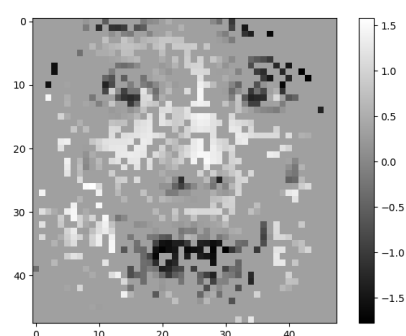
原圖



Saliency Map

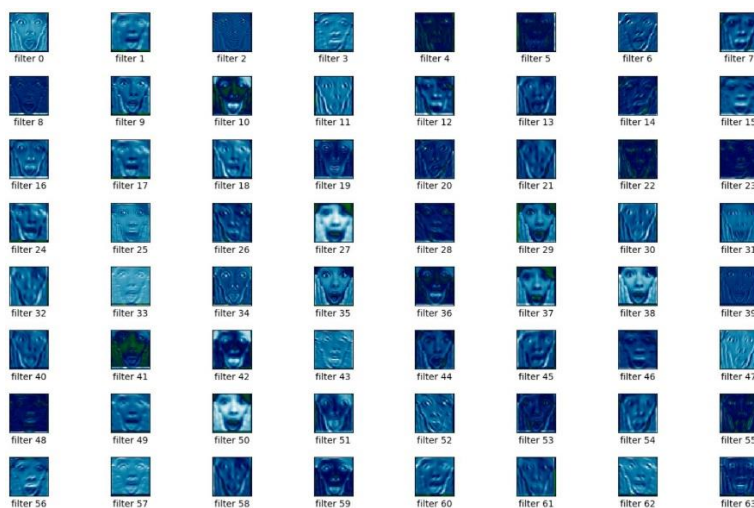


Mask 掉 heat 小的部份



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、洪唯凱、張少豪)



答：第一層 Conv2d_1



最後一層 Batch_normalization_5

眼睛嘴巴較明顯的圖片較易 activate。