說明：請各位使用此 template 進行 Report 撰寫，如果想要用其他排版模式也請註明題號以及題目內容（請勿擅自更改題號），最後上傳至 github 前，請務必轉成 PDF 檔，並且命名為 Report.pdf，否則將不予計分。
中英文皆可，但助教強烈建議使用中文。
----------閱讀完以上文字請刪除----------

學號：B04901020 系級： 電機三 姓名：解正平

1. (1%) 請說明你實作的 CNN model，其模型架構、訓練參數和準確率為何？ (Collaborators:)

   答：

   模型架構：CNN x5 DNN x3，BatchNormalization，Dropout 小到大，maxpooling x3

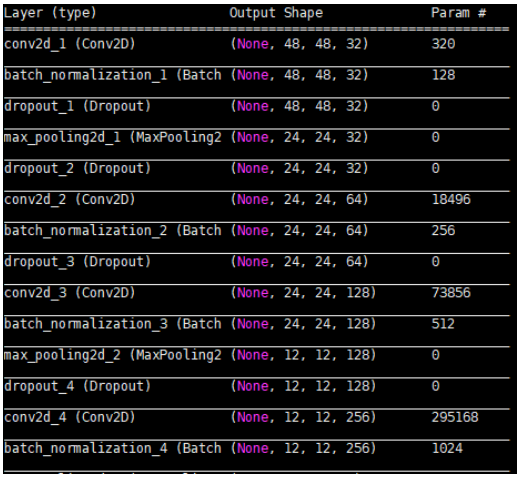
   訓練參數：neuron 小到大(CNN：32->64->128->256->256，DNN：512->1024->7)
   filter = 3x3，activation = selu，padding = same，

   準確率：training accuracy = 0.91，public score = 0.69295，private score =

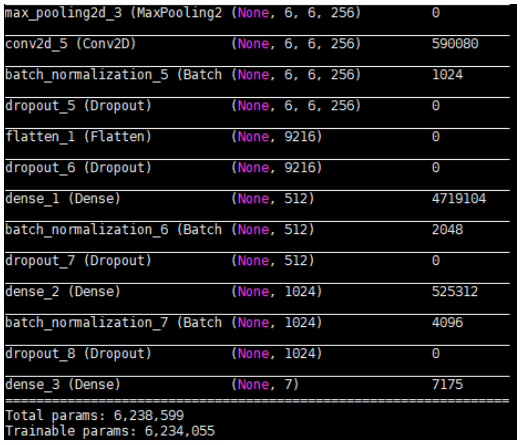
   總共疊了五層 CNN 及三層 DNN，CNN 架構先設定 neuron 數較小因為覺得機器做圖像分類剛開始不需要太多類別，只要能大概區分人臉部位就好，接下來數量變多是希望可以多一點細項分析。

   使用 selu 是因為它是目前較特別較實際的 activation function，然而會因為需要訓練的參數變多，需要 traing 較久也容易造成 overfitting。

   maxpooling 使用三次是因為怕太多次電腦已經無法辨識圖片，每次使用都會使圖片維度減半，從 48x48->24x24->12x12->6x6。



Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 32)	320
batch_normalization_1 (Batch	(None, 48, 48, 32)	128
dropout_1 (Dropout)	(None, 48, 48, 32)	0
max_pooling2d_1 (MaxPooling2	(None, 24, 24, 32)	0
dropout_2 (Dropout)	(None, 24, 24, 32)	0
conv2d_2 (Conv2D)	(None, 24, 24, 64)	18496
batch_normalization_2 (Batch	(None, 24, 24, 64)	256
dropout_3 (Dropout)	(None, 24, 24, 64)	0
conv2d_3 (Conv2D)	(None, 24, 24, 128)	73856
batch_normalization_3 (Batch	(None, 24, 24, 128)	512
max_pooling2d_2 (MaxPooling2	(None, 12, 12, 128)	0
dropout_4 (Dropout)	(None, 12, 12, 128)	0
conv2d_4 (Conv2D)	(None, 12, 12, 256)	295168
batch_normalization_4 (Batch	(None, 12, 12, 256)	1024



max_pooling2d_3 (MaxPooling2	(None, 6, 6, 256)	0
conv2d_5 (Conv2D)	(None, 6, 6, 256)	590080
batch_normalization_5 (Batch	(None, 6, 6, 256)	1024
dropout_5 (Dropout)	(None, 6, 6, 256)	0
flatten_1 (Flatten)	(None, 9216)	0
dropout_6 (Dropout)	(None, 9216)	0
dense_1 (Dense)	(None, 512)	4719104
batch_normalization_6 (Batch	(None, 512)	2048
dropout_7 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 1024)	525312
batch_normalization_7 (Batch	(None, 1024)	4096
dropout_8 (Dropout)	(None, 1024)	0
dense_3 (Dense)	(None, 7)	7175

Total params: 6,238,599
Trainable params: 6,234,055

2. (1%) 請嘗試 data normalization, data augmentation,說明實行方法並且說明對準確率有什麼樣的影響？

   答：

   data normalization 將讀檔的所有圖片的同樣 pixel 位置數值做標準化，使得每個圖片的該 pixel 不會差異太大，但也是會使得每張圖片看起來差不多，減少了圖片的差異，可能會因為 normalize 而造成結果不理想，無法清楚分辨圖片差異。

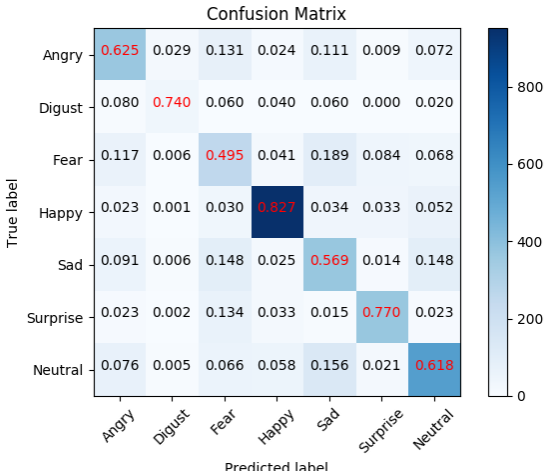
data augmentation 使用 ImageDataGenerator 這個 function 來生成更多 data，包括將 image 高度平移、寬度平移或是整個圖選轉一個小角度，另外再剛開始讀 data 的時候我還將全部 data 的矩陣做映射，就像是每張臉的左右互換照鏡子，這樣就可以直接有雙份的 data 量，較容易成功 train 好我的 model。

	Public score	Private score
Original model	0.63415	0.63499
Data normalization	0.13429	0.13262
Data augmentation	0.67901	0.66870

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]

答：

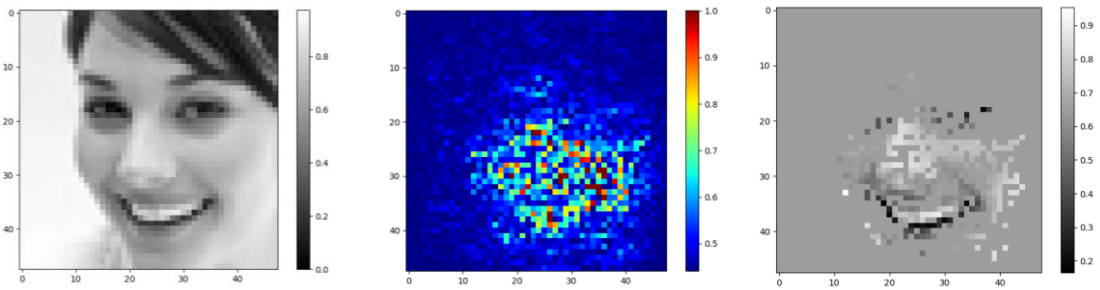
從圖中發現，Fear and Sad 較容易預測錯誤。Fear 正確率較低的原因是常常與 sad 搞混，大概每 10 張 Fear 的 data 就會有兩張錯誤判斷成 sad。再來就是 sad 的錯誤多來自判斷成 Fear 或是 Neutral，大概每 10 張 Sad 的 data 就會有三張被判斷成這兩者之一。

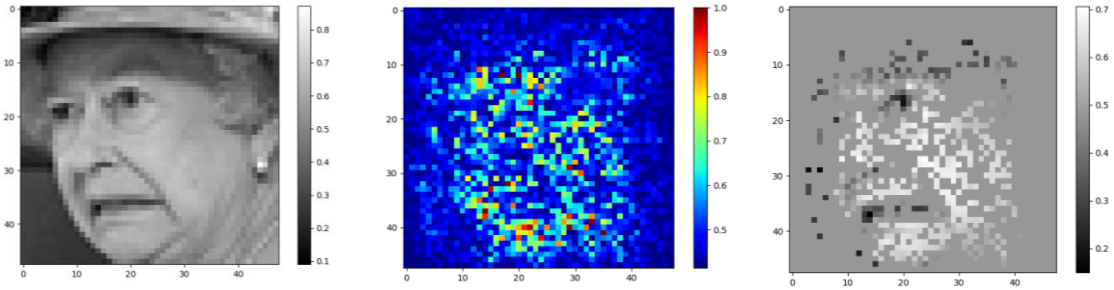


4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？

答：

下列分別為 Happy 以及 fear 的 data，很明顯發現 Happy 著重在嘴巴部分就跟我辨識一樣，開懷大笑就覺得是開心；Fear 則是著重在眼睛和嘴巴，畢竟恐懼會使人眼睛睜大嘴巴下沉。

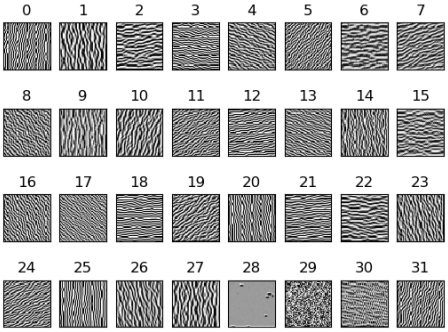




5. (1%) 承(1)(2)，利用上課所提到的 gradient ascent 方法，觀察特定層的 filter 最容易被哪種圖片 activate。

答：

右圖為第一層 layer 透過 gradient ascent 畫出最大化 filter output 圖片。可以發現許多 filter 都是相同線條只是旋轉幾度而已，因為每份 data 角度拍攝角度都不一樣，五官位置也不同，所以為了成功辨識，filter 要從各種角度抓取特徵。



觀察如下左上圖，能 activate filter 的圖片多為較粗的紋理，可能是因為他是第一層 CNN Block，負責抓線條的特徵，像臉部輪廓等，因此擁有粗紋理的圖片較能 activate 這一層的 filter。另外觀察右上圖，它是下一層 CNN 具有 64 個 filter (只呈現 32 個)，則可以發現多著重在面積上面，比如說眼睛跟嘴巴。再來就是為之後兩層 CNN 如下兩圖 (只呈現 32 個)，開始有點變形，只與整個辨識相關的部位會有 pixel，但因為經過 Maxpoolin，已無法思考機器如何判斷。