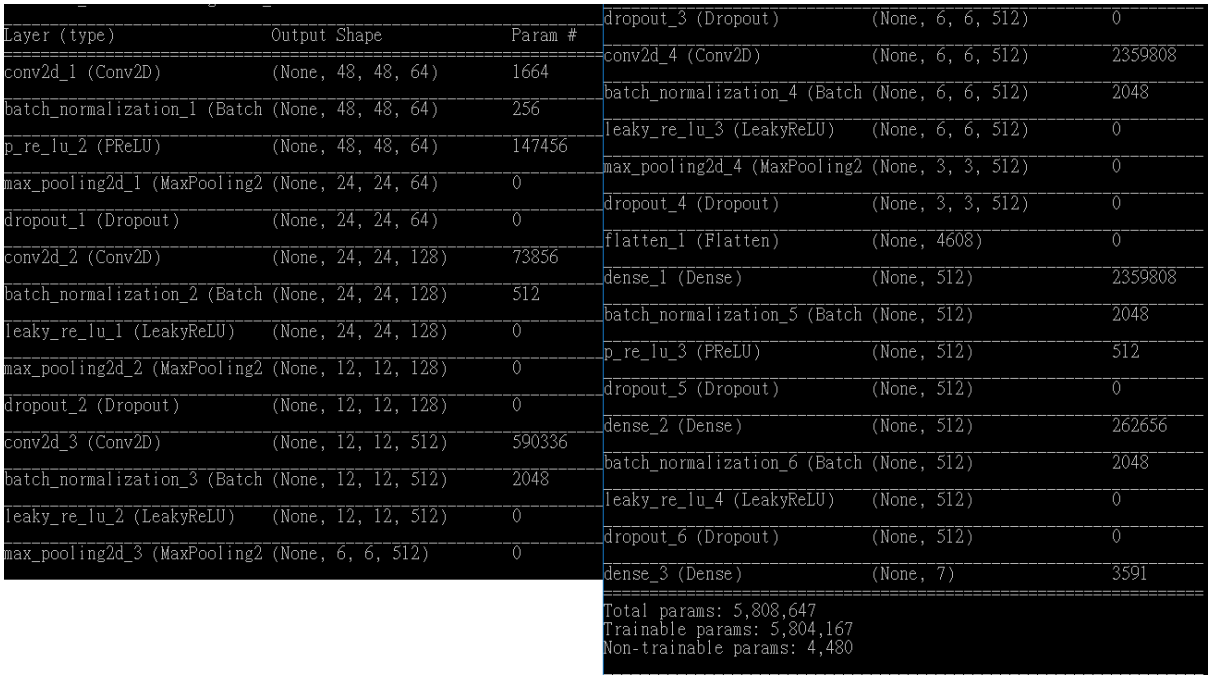
學號：r06921058 系級：電機碩一 姓名：方浩宇

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：



Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 48, 48, 64)	1664
batch_normalization_1 (Batch	(None, 48, 48, 64)	256
p_re_lu_2 (PReLU)	(None, 48, 48, 64)	147456
max_pooling2d_1 (MaxPooling2	(None, 24, 24, 64)	0
dropout_1 (Dropout)	(None, 24, 24, 64)	0
conv2d_2 (Conv2D)	(None, 24, 24, 128)	73856
batch_normalization_2 (Batch	(None, 24, 24, 128)	512
leaky_re_lu_1 (LeakyReLU)	(None, 24, 24, 128)	0
max_pooling2d_2 (MaxPooling2	(None, 12, 12, 128)	0
dropout_2 (Dropout)	(None, 12, 12, 128)	0
conv2d_3 (Conv2D)	(None, 12, 12, 512)	590336
batch_normalization_3 (Batch	(None, 12, 12, 512)	2048
leaky_re_lu_2 (LeakyReLU)	(None, 12, 12, 512)	0
max_pooling2d_3 (MaxPooling2	(None, 6, 6, 512)	0
dropout_3 (Dropout)	(None, 6, 6, 512)	0
conv2d_4 (Conv2D)	(None, 6, 6, 512)	2359808
batch_normalization_4 (Batch	(None, 6, 6, 512)	2048
leaky_re_lu_3 (LeakyReLU)	(None, 6, 6, 512)	0
max_pooling2d_4 (MaxPooling2	(None, 3, 3, 512)	0
dropout_4 (Dropout)	(None, 3, 3, 512)	0
flatten_1 (Flatten)	(None, 4608)	0
dense_1 (Dense)	(None, 512)	2359808
batch_normalization_5 (Batch	(None, 512)	2048
p_re_lu_3 (PReLU)	(None, 512)	512
dropout_5 (Dropout)	(None, 512)	0
dense_2 (Dense)	(None, 512)	262656
batch_normalization_6 (Batch	(None, 512)	2048
leaky_re_lu_4 (LeakyReLU)	(None, 512)	0
dropout_6 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 7)	3591

Total params: 5,808,647
Trainable params: 5,804,167
Non-trainable params: 4,480

這是我的模型架構。其中先使用一層(64,(5,5))的 conv2D，然後是(128,(3,3)),(512,(3,3))的 conv2D，其中每一個 conv2D 中間都有加入一個 LeakyRelu(alpha 為 0.05),除了第一層的為 PRelu，還有一個 2*2 的 Maxpooling2D，以及一個 BatchNormalization 與 Dropout。最後將整個 Model Flatten 之後，全連接層是兩個 Dense(512)，最後輸出層是 Dense(7)，activation 依序為 PRelu,LeakyRelu,softplus

參數設定的部分，第一層為 Conv2D(64,(5,5),border_mode='same',kernel_initializer='glorot_normal',input_shape=(48,48,1))

之後的 Conv2D 除了 Filter 數量和大小之外都與第一層相同， filter 的數量與大小在前一段已經描述。LeakyRelu 的 alpha 都是 0.05，PRelu 初始都是 0，Maxpooling 皆為(2,2)，loss 為'categorical_crossentropy'，optimizer 為’adam’

總共訓練 700 個 Epochs，有使用 Shuffle，samples_per_epoch 為原始訓練資料的一半數量。

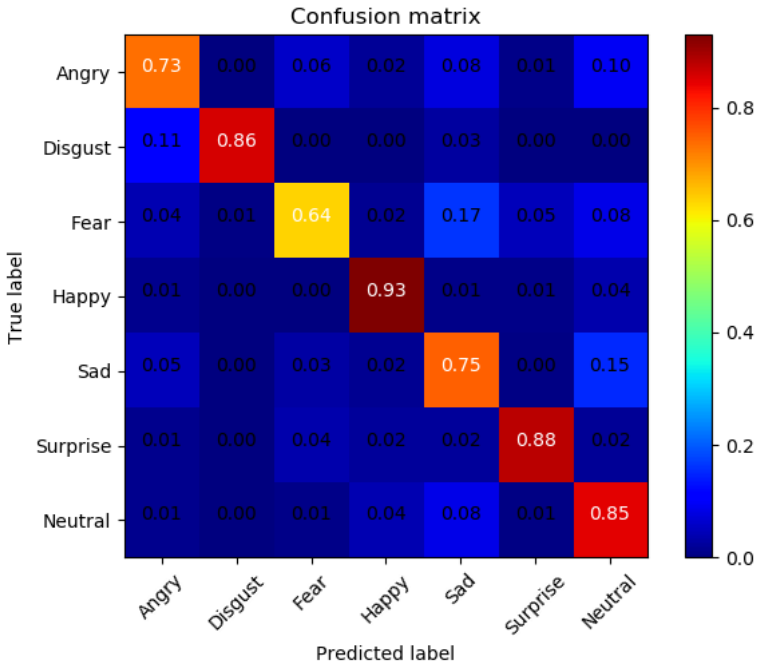
predict_7_Prelu_New3_700.csv 2 days ago by r06921058_>.O add submission details	0.68709	0.69295

準確率如圖，public 為 0.69，private 為 0.68

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？
   Data normalization 的部分，我是將全部的同一位置的 pixel 作平均以及標準差，然後個別減掉平均之後除以標準差，Data augmentation 的部分則是使用 ImageDataGenerator，參數為 rotation_range=40, width_shift_range=0.2, height_shift_range=0.2, shear_range=0.2, zoom_range=0.2, horizontal_flip=True, fill_mode='nearest'。Kaggle 上的分數如圖，其中 1,2 使用相同參數，只差 Augmentation，3,4 使用相同參數，只差 Normalization，每個都是訓練 300 個 Epochs。
   5 在訓練時最後準確率為 0.56，validation 為 0.53~0.6 之間移動。
   4 訓練時最後準確度為 0.55，validation 為 0.58~0.61 之間移動。
   2 訓練時最後準確度為 0.98，validation 為 0.59~0.62 之間移動。
   1 訓練時最後準確度為 0.62，validation 為 0.61~0.64 之間移動。
   理論上 Normalization 和 Augmentation 應該都會對結果有所幫助，其中沒有 Normalization 反而提高準確率的原因，可能是因為沒有標準化變動比較快，而我的 Model 沒有讓他們收斂的最後，所以才造成沒有 Normalization 反而提高準確率的結果。

Submission and Description	Private Score	Public Score
predict_report_5_Norm_300.csv a few seconds ago by r06921058_>.O add submission details	0.52187	0.54026
predict_report_4_NNorm_300 (2).csv just now by r06921058_>.O add submission details	0.62496	0.60741
predict_report_2_dataNoAug_300.csv a few seconds ago by r06921058_>.O add submission details	0.60546	0.61270
predict_report_1_dataAug_300.csv 6 hours ago by r06921058_>.O add submission details	0.63750	0.64697

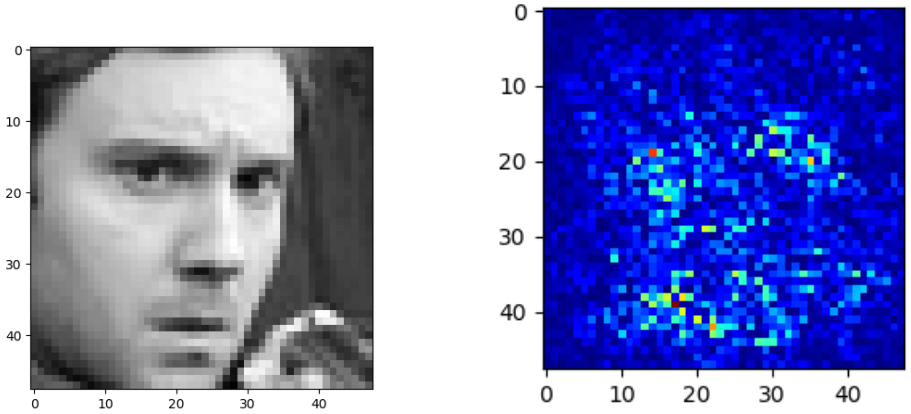
3. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分



析]

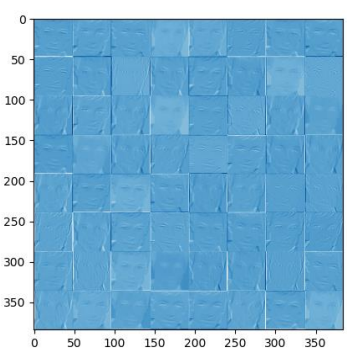
其中 SAD 容易預測成 Neutral，Fear 容易預測成 Sad，推測可能是因為 SAD 和中立的表情嘴部比較接近一點。

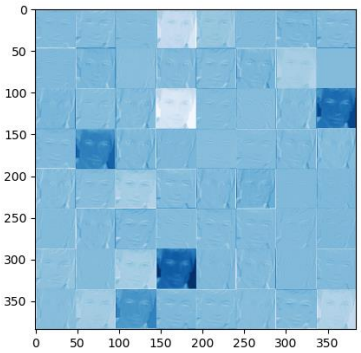
4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？



從這張可以看出，主要是 focus 在眼睛，鼻子及嘴巴的部分。

5. (1%) 承(4) 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 與觀察 filter 的 output。





這是 conv2D_1 觀察內容。可以發現其中不少都著重在眼睛，嘴巴的部分。