

學號：R06725041 系級：資管碩一 姓名：彭証鴻

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**: 參考網路上的 github code 來實作，並參考助教的 sample code 來調整參數。)

答：

模型架構：

Model1：六層 CNN 加上兩層 Dense layer，output layer 的 activation function 設為 softmax。

Model2：八層 CNN 加上兩層 Dense layer，output layer 的 activation function 設為 softmax，與 Model1 相比有做 **early stopping**。

Model3：八層 CNN 加上兩層 Dense layer，output layer 的 activation function 設為 softmax，與前兩個 Model 相比僅改變第一層 CNN 的 activation function。

訓練參數：

Model1：

CNN：參數 filters 分別為(32,32,64,64,128,128)，kernel_size 則為(3,5,3,5,3,3)，CNN 的 activation function 皆為 relu。而 Dense layer 的 activation function 為 PReLU。此外，因為一開始的 model 都會有 overfitting 的傾向，所以在每一層 CNN 後會加上 Dropout 和 BatchNormalization。

Model2：

與 Model1 相比，多家的兩層 CNN filters 分別為(256,256)，kernel_size 則為(3,5)，其餘不變。

Model3：與前兩個 Model 相比僅將第一層 CNN 的 activation function 改為 selu。

準確率：

Model1：Public score: 0.65756 Private score: 0.66090

Model2：Public score: 0.66536 Private score: 0.66118

Model3：Public score: 0.67233 Private score: 0.65171

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？

(Collaborators: normalization 有參考助教的 sample code)

答：

寫出實作 **data normalization** 過程、與實作前、實作後準確率。

首先，將讀進來的 pixel array 先除以 255，使 pixel value 介於 0~1 之間，隨後扣掉其平均值並除以其標準差，得到 normalization 後的 pixel array。

實作前：public score: 0.65561 private score: 0.65254

實作後：public score: 0.66536 private score: 0.66118

寫出實作 **data augmentation** 過程、與實作前、實作後準確率。

透過 ImageDataGenerator 將 pixel array 做旋轉、平移縮放，產生更多的 training data 使 training 變得更好。

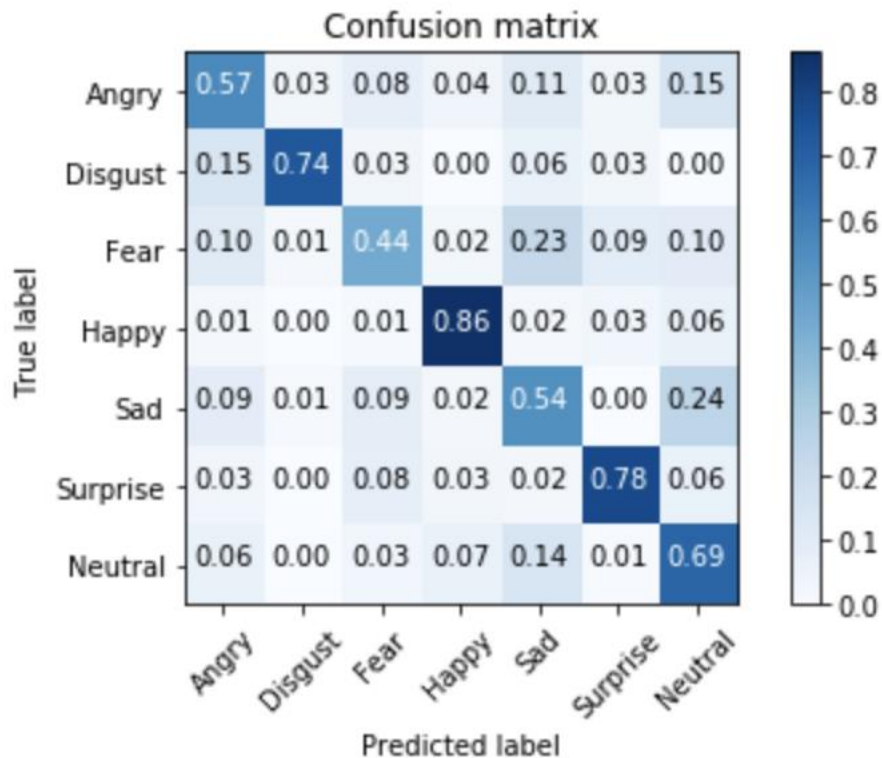
實作前：public score: 0.65912 private score: 0.64892

實作後：public score: 0.66035 private score: 0.65171

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: 參考 scikit-learn 關於 confusion matrix 的相關 code)

答：發現 **fear** 跟 **sad** 和 **sad** 跟 **Neutral** 答錯比率高達兩成。



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：合理說明 **test** 的圖片和觀察到的東西 -> 0.5 分
貼出 **saliency** 圖片 -> 0.5 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**。
(Collaborators:)

答：合理說明 **test** 的層數和觀察到的東西 -> 0.5 分
貼出 **filter input and output** 的圖片 -> 0.5 分