

### 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？

本次作業我主要使用了兩種model,並將結果做ensemble

**Model1:** Conv2D(32,3,3) -> MaxPooling(2,2) -> Conv2D(64,3,3) -> MaxPooling(2,2)  
->Conv2D(128,3,3) -> MaxPooling(2,2) -> flatten() -> dense(512) -> dense(512) ->dense(7)

**Model2:** Conv2D(32,3,3) -> BatchNorm() -> Conv2D(32,3,3) -> BatchNorm()  
->MaxPooling(2,2) ->Conv2D(64,3,3) -> BatchNorm() -> Conv2D(64,3,3) -> BatchNorm()  
->MaxPooling(2,2) ->Conv2D(128,3,3) -> BatchNorm() -> Conv2D(128,3,3) -> BatchNorm()  
->MaxPooling(2,2) -> flatten() -> dense(1024) -> dense(512) ->dense(7)

Activation function使用selu, 最後一層用softmax, Input用dataGenerator產生4倍的資料, 在每層Conv layer間加入dropout 由0.2漸增到0.5, loss function = cross entropy, optimizer = Adam

**Result:** Training accuracy: 0.8344, Public accuracy: 0.71273, Private accuracy: 0.70437

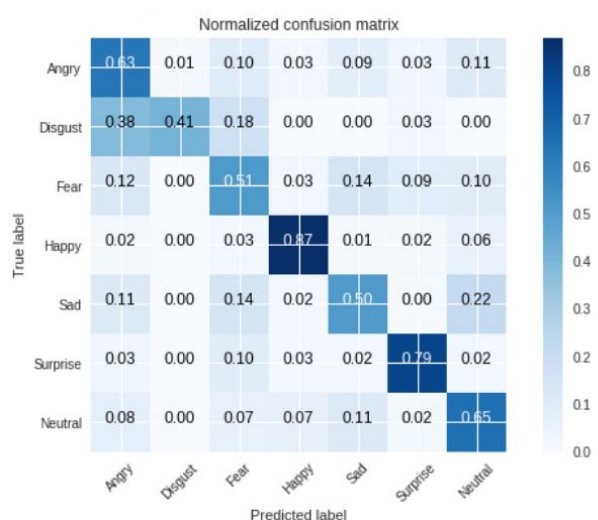
### 2. (1%) 請嘗試 data normalization, data augmentation,說明實行方法並且說明對準確率有什麼樣的影響？

a.) data normalization 計算出所有圖片中所有pixel的平均和標準差,再套用到所有的圖片上, 可以更忠實的呈現該圖片的資訊, 實驗結果在下表

b.) data augmentation直接使用Keras內建的ImageDataGenerator, 並加入旋轉10度,上下左右偏移, 和左右對稱, 使用了data augmentation使資料增加, 減輕overfitting的情況,下表是實驗的結果:

	Training accuracy	Public accuracy	Private accuracy
both augmentation & standardization	0.8344	0.71273	0.70437
without data standardization	0.7855	0.67982	0.66759
without data augmentation	0.9398	0.60388	0.59403

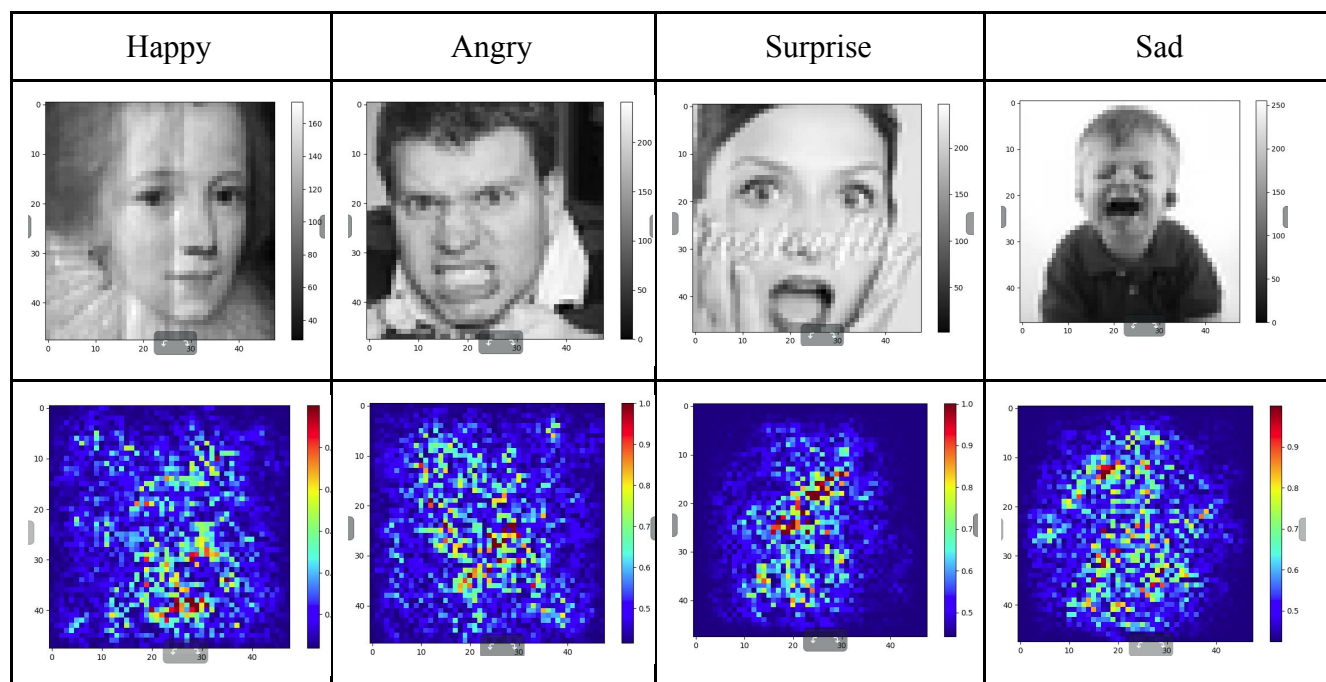
### 3. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析]



從confusion matrix中可以明顯看出, isgust 很容易被判斷成Angry, 但Angry卻不容易被判斷成Disgust, 且Disgust為所有情緒中誤判率最高的

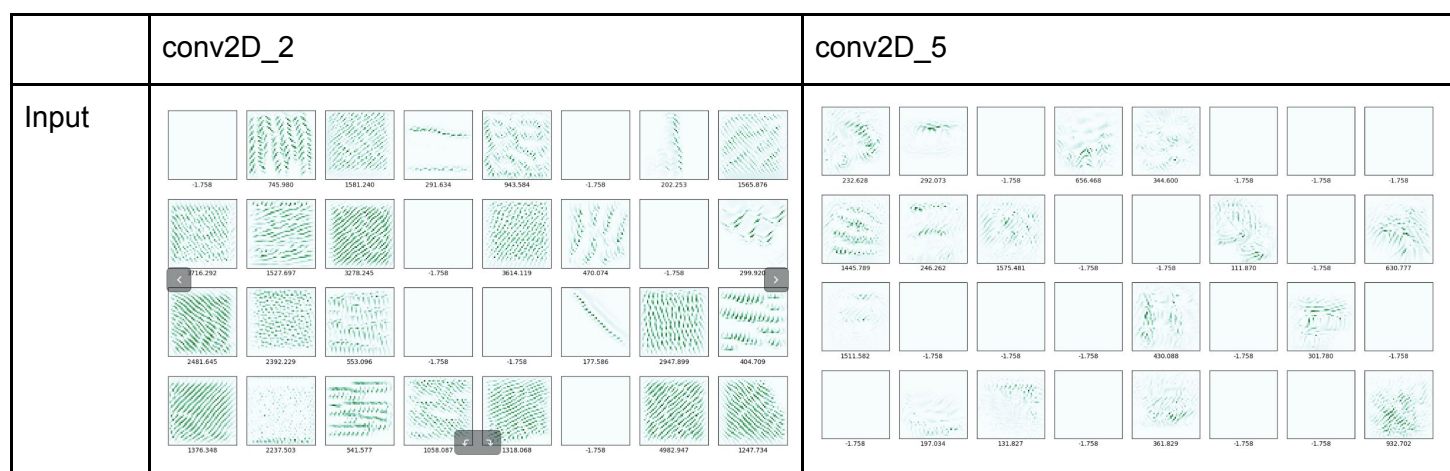
此外Fear和Sad彼此間都有0.14%誤判的機率

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？

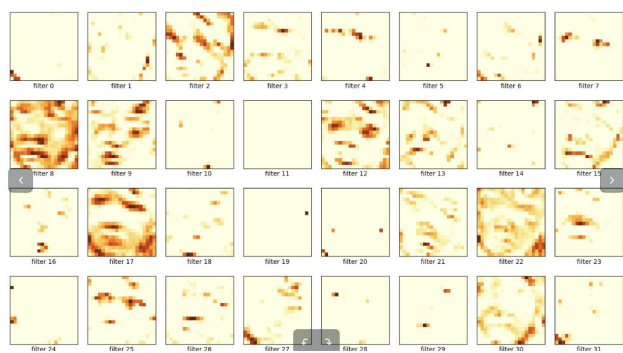


由上面的幾個結果可以看出，CNN在做分類時，會注重人嘴型的大小，由嘴型的樣式可以大概分類出是什麼樣的情緒，此外有些例子眼睛附近heat會很高，這可能是因為此張照片的眼球相較於其他照片異常的大或小，使機器可以判斷出這張照片是什麼樣的情緒

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: )



## Output



在較上層可以看出明顯是比較粗的紋路，雖然其中不乏有相似的filter只是角度有轉而已，而較下層可以看到許多filter長的像鼻子或嘴巴，顯示在一層層的filter後，CNN不只能辨識很粗淺的紋路，也能去尋找人臉上的特徵