學號：B03902130 系級： 資工四 姓名：楊書文

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: b03505040 林後維、b03505052 李吉昌)**

   我的模型架構如連結: https://paste.ofcode.org/KStWBUnx7rHNwy9ifQXABW，是個非常大的 model。參考 VGG 的架構，每單位三層 conv，疊了四單位共 12 層 conv，最後再接三層 dense。其中 activation 都使用 relu，每個 conv 後都接 batch normalization，且每單位 conv 與 dense 之後都接 dropout，並根據 dropout 前的 output tensor 大小決定 dropout 的比例。Train 的過程先使用 Adadelta 可以較快速的收斂到準確率約 0.6 初頭，再靠 SGD 慢慢將 accuracy 推上去。將原始的 data 經過 argumentation 產生 13 倍與 20 倍 data 後，用同一個模型架構和訓練參數分別 train 出 model1 與 model2，accuracy 的極限都在 0.68 左右。最後再將兩個 model 合併起來做 ensemble 就可以達到 accuracy 0.706。

MODEL	PUBLIC	PRIVATE
model1	0.68180	0.67149
model2	0.68124	0.67511
ensemble	0.70632	0.70047

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: b03505040 林後維、b03505052 李吉昌)**

   Data normalization: 在我的 model 中有無 data normalization 並沒有非常明顯的影響，我想這是因為架構中有 batch normalization，也可以有 normalize 的效果。若把 batch normalization 拿掉，則從表上可看到明顯退步。我 data normalization 的方式是: `( imgae - image.mean() ) / image.std()` 。

MODEL	PUBLIC	PRIVATE
w data normalization	0.64781	0.64196
w/o data normalization	0.64530	0.63666
w/o batch normalization	0.56338	0.55920

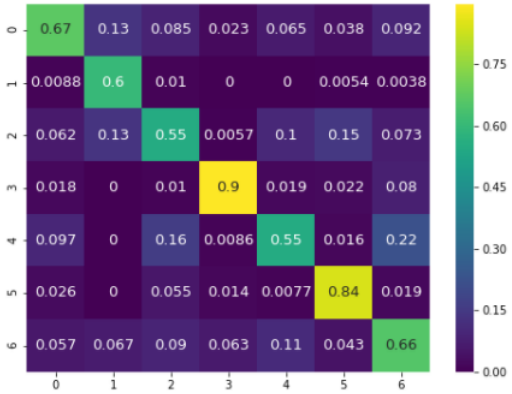
   Data augmentation: 有沒有做 augmentation 會有明顯影響，我想可能是因為我的 model 相當龐大，如果沒有夠大量的 data 則容易 overfitting。我 augmentation 的作法是將原圖上下左右移動、水平翻轉、小角度旋轉，共產生了 13 倍與 20 倍的 data，表格中紀錄了 20 倍 data 的數據。

MODEL	PUBLIC	PRIVATE
w augmentation	0.68124	0.67511
w/o augmentation	0.64781	0.64196

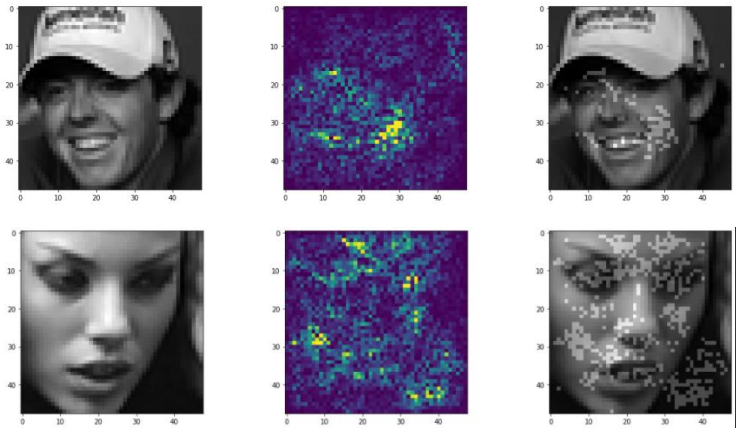
3. **(1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析] (Collaborators: b03505040 林後維、b03505052 李吉昌)**

   從下頁 confusion matrix 中可以看到，model 對於恐懼和難過的圖片是比較不準的，其中恐懼的圖片較容易被誤判成厭惡、難過或驚訝，而難過的圖片則容

易被誤判成恐懼或中立。

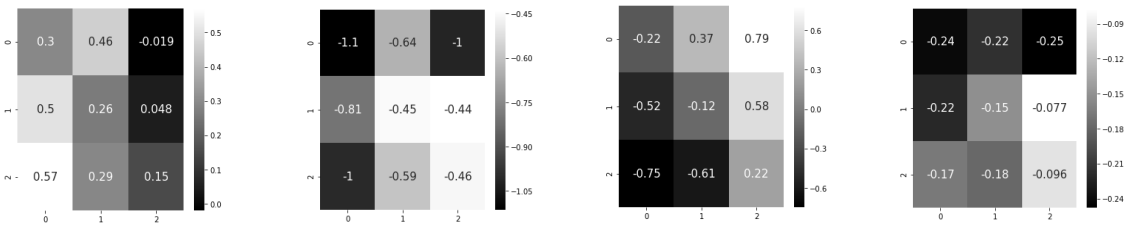


4. **(1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？**
   (Collaborators: b03505040 林後維、b03505052 李吉昌)

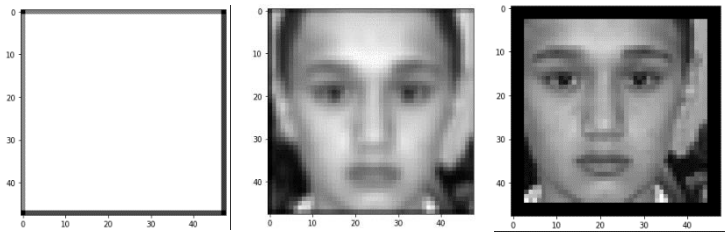


右圖中分別是開心與厭惡兩類別的 saliency map。最左圖為原圖；中間為 saliency map，越亮的代表越能影響最後的機率；最右圖則將影響機率較高的 pixel 在原圖中標示出來。可以看到我的 model 似乎會 focus 在臉部肌肉線條的部分，開心的類別會 focus 在嘴角的笑容；而厭惡的類別則 focus 在臉頰些微的揚起和眼睛周圍的形狀。做這個實驗時我有特別挑 model predict 正確、且很確定的 (該類別 output 機率達 90%以上) 圖片來實驗，因為有時候挑的圖片根本預測錯誤，或 model 自己其實相當沒信心，那就不容易觀察到有用的資訊。

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: b03505040 林後維、b03505052 李吉昌)
   我使用我 model 中的第七層 conv 來實驗，因為比較過幾層的 filter 之後，發現較前面的 filter 有時只能學到一些太簡單的 pattern，而導致整層 filter 將其 gradient ascent 出來都是類似的東西，尤其我第一層 conv 僅為 3*3 的 filter，能表達的 pattern 會太簡單。將其畫出來大概如下圖:



第一層中有很多類似這樣的 filter，將其 gradient ascent 出來的圖會如下左圖的方框狀，而圖片經由這樣的 filter 後會得到如中間那張的 activation map。可以看到和原圖沒有太大差別，僅是邊框部分有被



些微處理過。觀察太前面的層數似乎較沒有明顯的資訊，我猜想是因為我做 data augmentation 時為了讓臉在圖中移位，加入大量將圖片 crop 後於任意位置做 zero padding 的圖片，如最右邊那張。這類圖片都有明顯的邊框，因此可能 model 將其認定為有用的 pattern。最後我選擇較中間的第 7 conv 層，避免

太前面的 pattern 太簡單或太後面的太複雜。input image 使用哀傷的圖片，並把其 filter 的 gradient ascent 和其 activation map 畫出來。以 (row, column) 來標示並把左上角標為 (0, 0) 的話，(0, 0) 感覺在辨認頭髮，而 (3, 3) 一橫線的 pattern 則能抓到嘴型與眼睛眉毛那邊的形狀。而 (3, 0) 則感覺在認皮膚的質感，activation map 上他感應到的的確是額頭、兩邊臉頰與嘴巴。