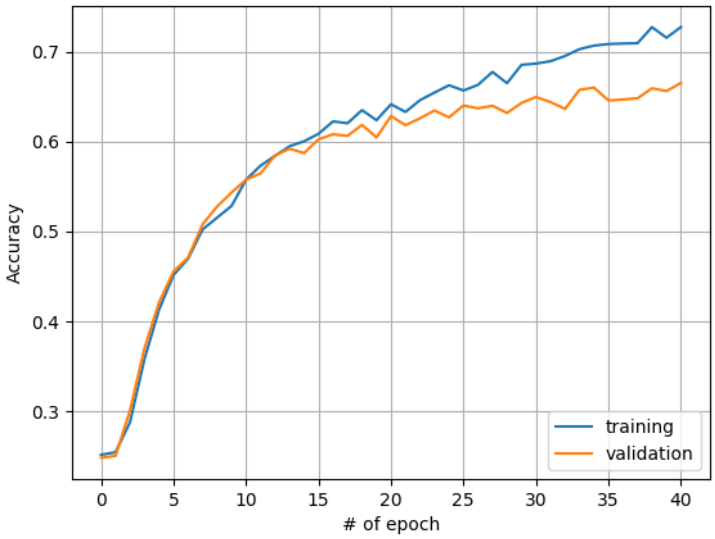
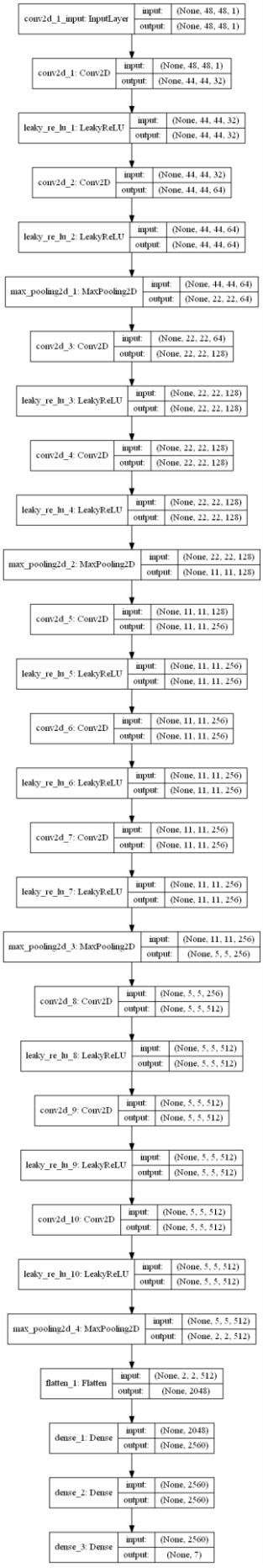
學號：B03901165 系級： 電機四 姓名：謝世暐

1. (1%) 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？

   (Collaborators: 楊耀程)

   答：右圖為我使用的 CNN model 架構，Conv+Dens 共有 10+3 層，Conv layer 的 activation 皆為 Leaky Relu (alpha = 0.004)，每一層用的 neuron 數目為(32,64,128,128,256,256,256,512,512,512,2560,2560,7)。訓練流程中使用 data augmentation，並且由於 data augmentation 增加資料的多樣性比較不容易 overfit，就沒有再加上 Dropout。訓練中切出最後 10%訓練資料作為 validation，以 validation 為參考做 early stopping (patience=5)。訓練後為求更高正確率再做 fine tuning，用小一點的 learning rate，固定前幾層只訓練後幾層，用之前訓練的 model 做初始化，重複幾遍。最後得到在 public 最高的正確率是 0.67874。(epoch-accuracy 圖如下)



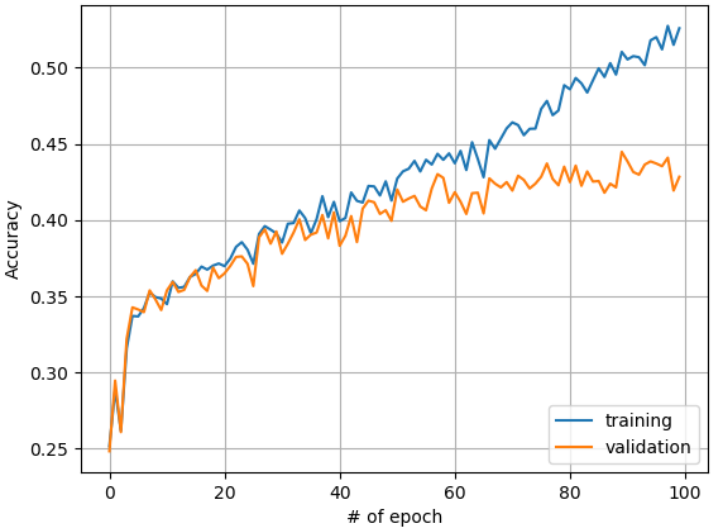


2. (1%) 承上題，請用與上述 CNN 接近的參數量，實做簡單的 DNN model。其模型架構、訓練過程和準確率為何？試與上題結果做比較，並說明你觀察到了什麼？

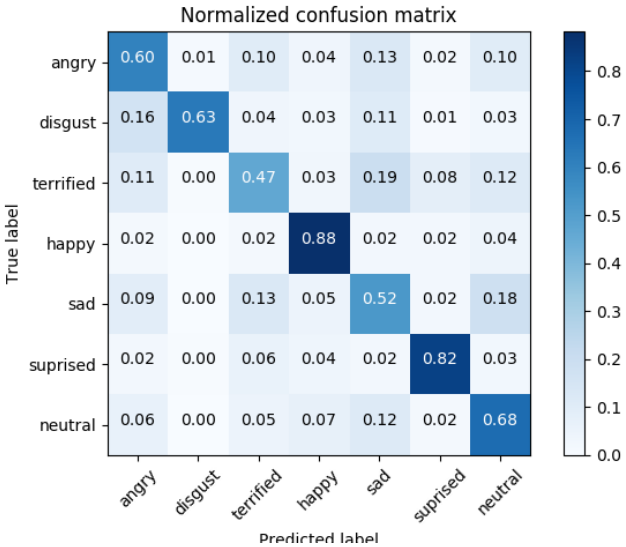
   (Collaborators: 無)

   答：下一頁的右圖為 DNN 的架構，由於 CNN 參數量=19468167，因此 DNN 的設計讓參數接近 CNN(最後為 19356167)，架構如下一頁右圖所示，共有 12 層 Dense，前 11 層每層為 1280 個 neuron，最後一層 7 個，為求公平比較 activation 也用 Leaky Relu (alpha 相同)，其他如 data augmentation 也同 CNN 方式，validation 也是切 training 最後 10%。訓練中 epoch-accuracy 圖如下，而在 public 的正確率為 0.38255。可以發現很明顯 DNN 的準確率比 CNN 低很多，即便用比 CNN 更多的 epoch 準確率也很難

train 上去，表示 CNN 確實有他的功效。

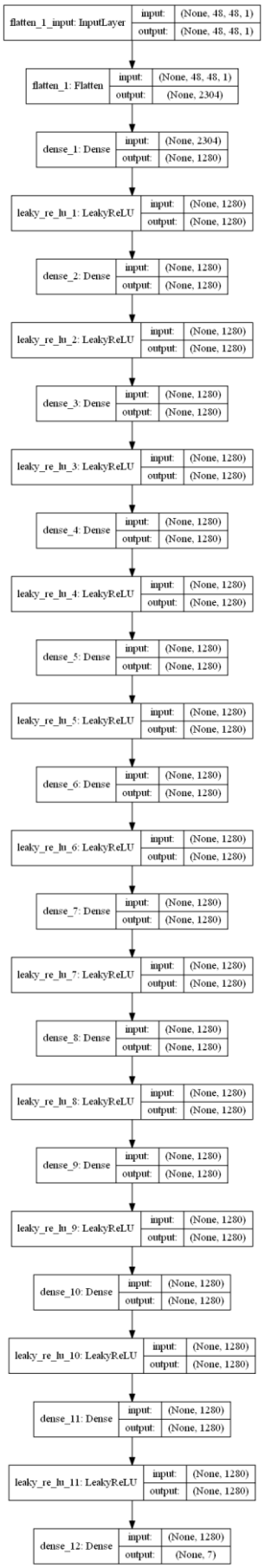


3. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析]
(Collaborators: 楊耀程)
答：Confusion Matrix 在 Validation 的結果如下:

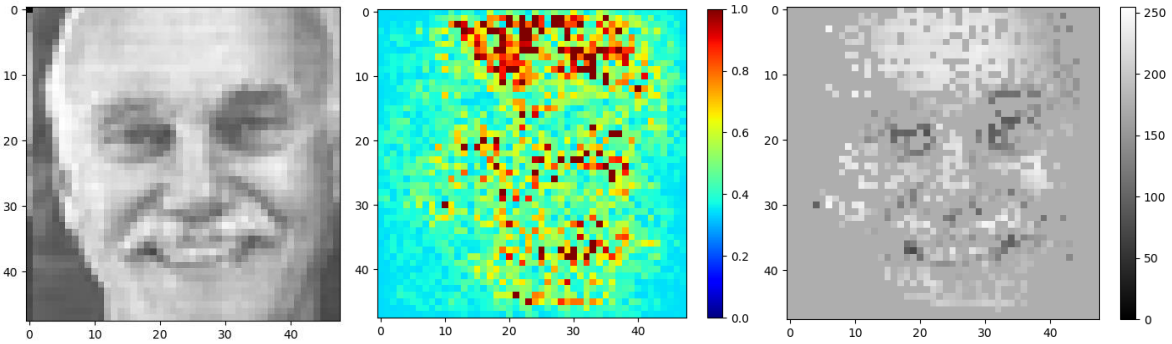


從上圖來看，若以 class 之間「彼此搞混」的機率相加平均來評比的話，以「害怕」和「傷心」為最容易搞混(平均機率為 0.16)，第二容易搞混的是「傷心」和「中立」(平均機率為 0.15)。

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: 楊耀程)



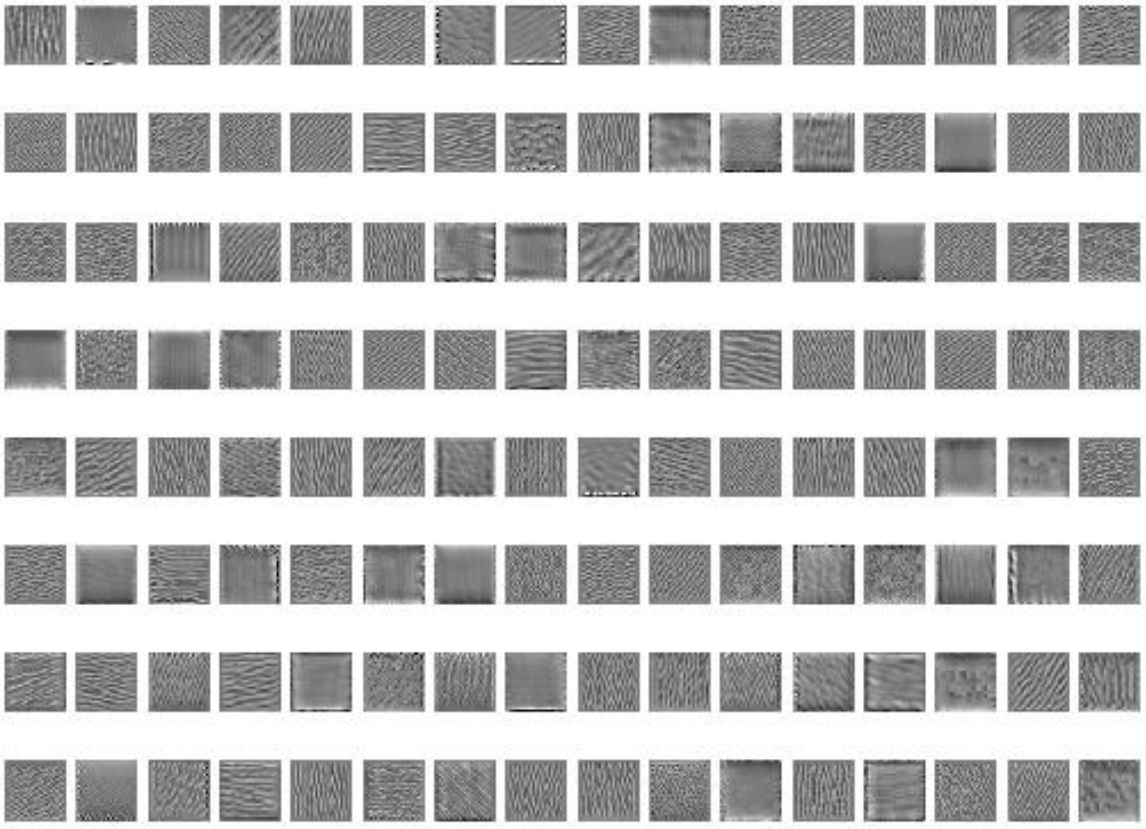
答：以下使用的圖片來自 validation set，分別為原圖，gradient map 和 mask 後圖



True label 為開心，Saliency map 計算方式由 output label 對 input image 作微分取 gradient，經過取絕對值後 normalize 讓標準差為 0.2，平均在 0.5。Masking 是將原圖 mask 掉 gradient<0.5 的部分，也就是低於平均值的部分，並用 128 取代。從 masking 過後的圖可以看出左右邊和五官無關的地方多數都被 mask 掉了，而五官部分則保留了下來，代表五官對於判斷表情才是最重要的部分，我們也確實可以單從被 mask 出來的部分判斷出這張的 label 應為開心。

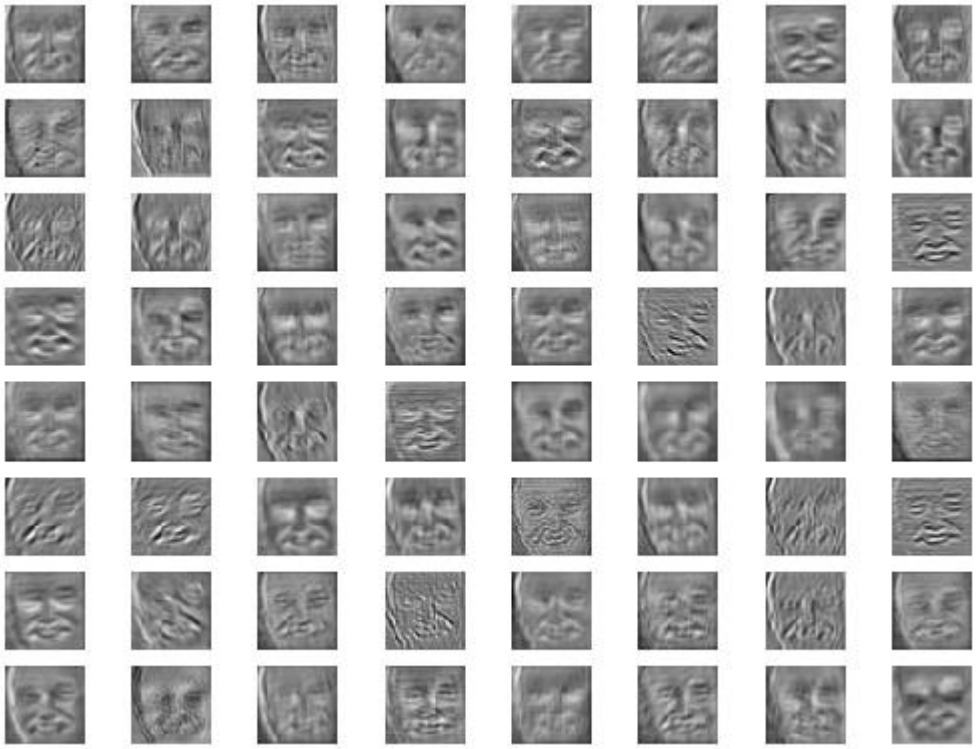
5. (1%) 承(1)(2)，利用上課所提到的 gradient ascent 方法，觀察特定層的 filter 最容易被哪種圖片 activate。
   (Collaborators: 楊耀程)

答：下圖為 conv2d_3 的所有 filter 將 noise 做 gradient ascent 的結果(共 128 個):



上圖中，filter output 都還是斜線條、斑點這種比較簡單的圖形，如果試用最後幾層如 conv2d_8，則就會顯現出比較複雜的紋路，但礙於那一層有 512 個 filter，沒空間擺所有 filter，就沒放那層了。我的 model 一開始 activation 實際上是使用 relu，而非 Leaky relu，但做到這題將 filter output 卻發現有很多 filter 的 gradient 都是 0，沒辦法做 gradient ascent，我猜測是因為 relu 會有 Dead relu 的問題，因此才將他換成 leaky 版本，也確實在換之後就沒有出現 gradient 是 0 的問題了。

下圖為用跟(4)一樣的圖經過 conv2d_2 的結果:



由於這邊是經過 conv2d_2，所以只有 64 張圖。可以觀察到大部分的圖有抓到五官的特徵，有些圖跟影像處理中做 gradient 或是 Laplacian 長的很類似，像是在做 edge detection。而影像處理中剛好 edge detection、corner detection、Laplacian 這些也是傳統做特徵擷取常用的技巧，代表 CNN 有將比較這些底層的處理確實的學起來。