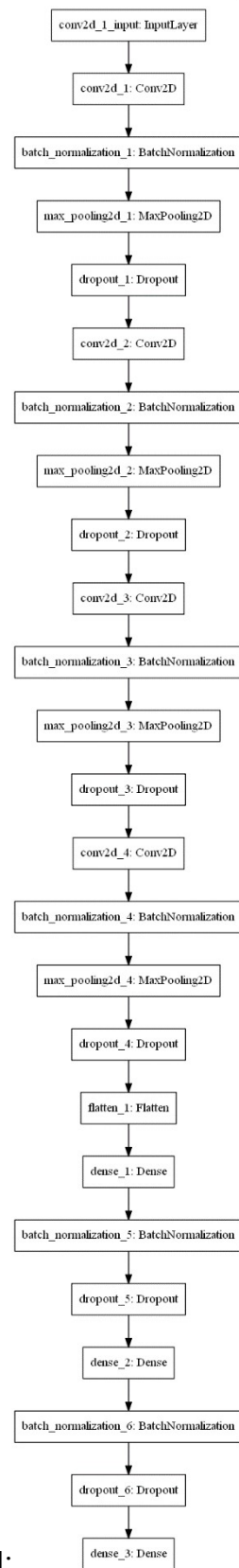
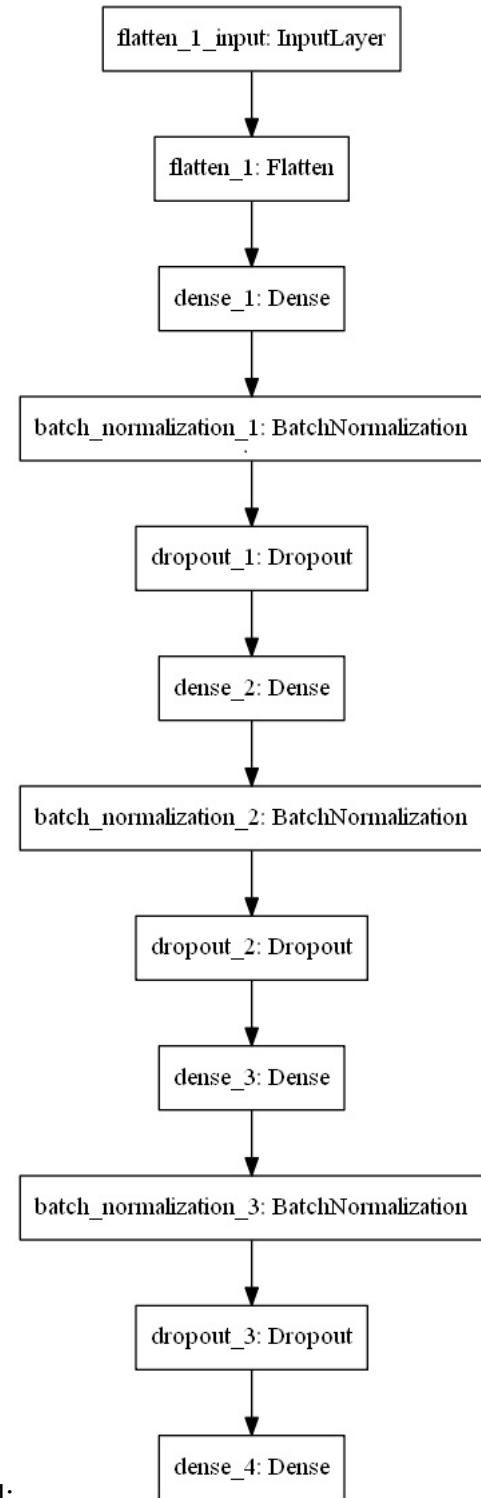


學號：B04902021 系級：資工三 姓名：陳弘梵



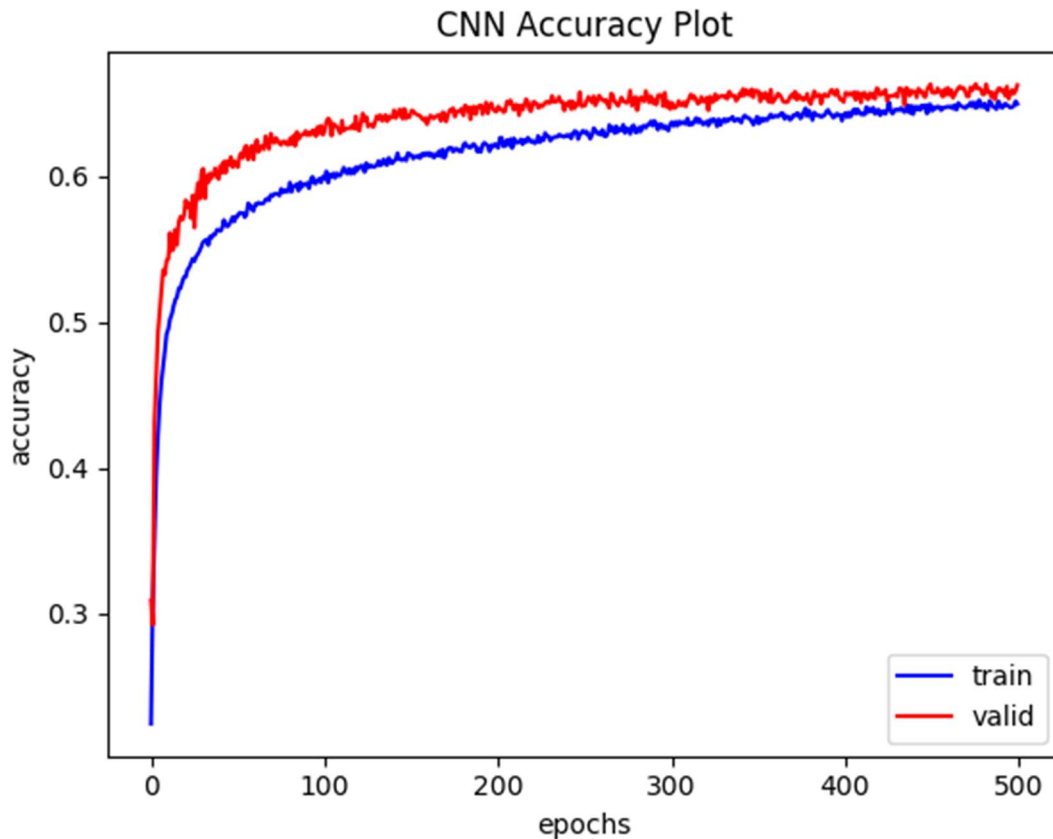
Model Structure:(1)CNN:



(2)DNN:

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？

實作 CNN 並且跑了 500 epoch 的模型如第一頁所示，其訓練過程與準確率如下：

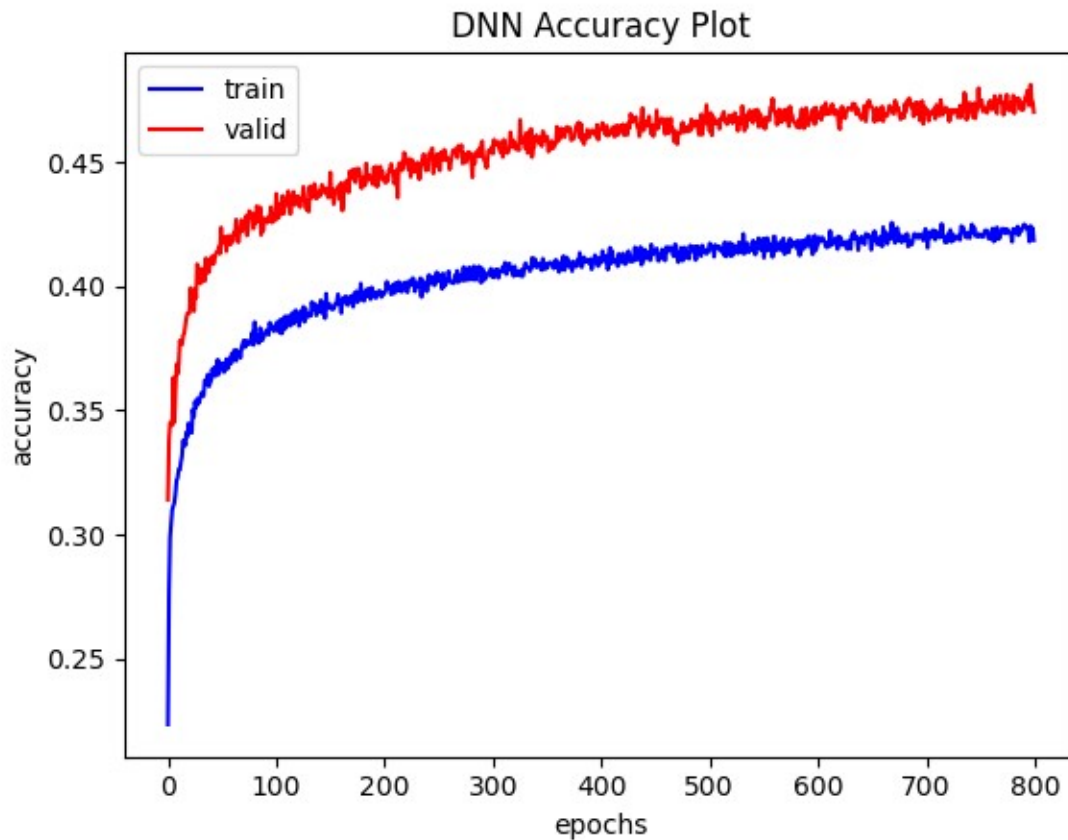


Trainable params: 1,362,055 · Validation 切 10% · 最終 val\_acc=0.66300。

- (1) Conv2D 與 Max\_pooling 採用了四層，dense 使用兩層，activation 皆使用 selu。
- (2) 由於模型中採用了大量的 dropout layer，可以發現我的 val\_acc 因為不像 train\_acc 有些 neuron 在訓練時沒有經過，導致準確率一直都保持著高於 train。
- (3) 而整個圖表中，明顯的變化是顯現在 train\_acc 的進步上，最後慢慢逼近 valid\_acc。

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？

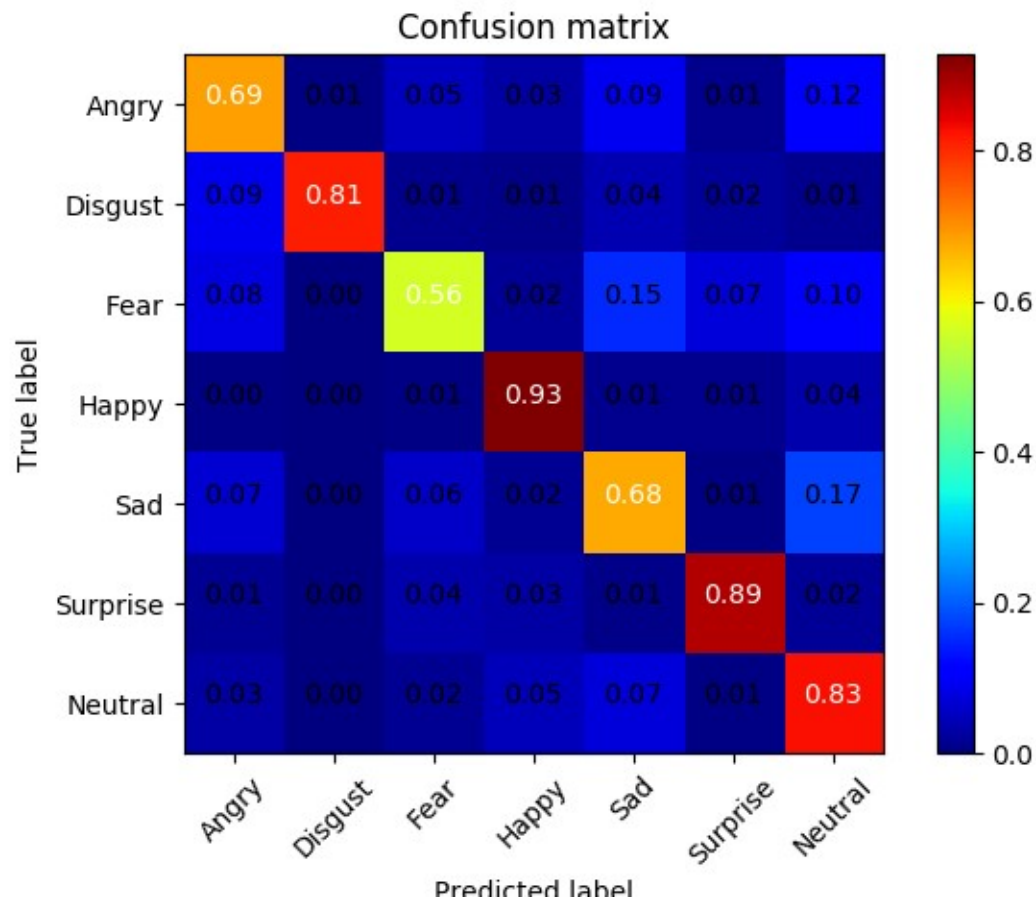
實作 DNN 並且跑了 800 epoch 的模型如第一頁所示，其訓練過程與準確率如下：



Trainable params: 1,381,127 · Validation 同切 10% · 最終 val\_acc= 0.48140。

- (1) 一樣因為使用 dropout，valid\_acc 高於 train\_acc。
- (2) 而相較於 CNN，觀察到這個模型的 val\_acc 與 train\_acc 在後期一直保持著相近的成長率，在精確率上也一直維持著一定數值的差異。
- (3) 整體而言的表現都比 CNN 同樣參數量的模型遜色許多，尤其是成長幅度非常緩慢，最後也無法突破 50%。

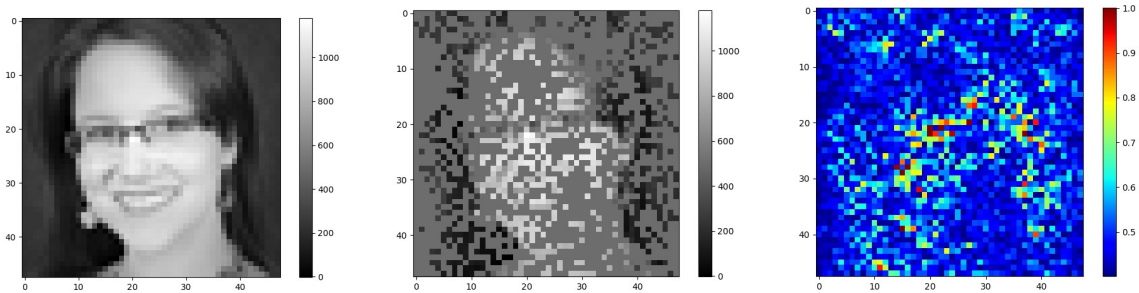
3. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析]



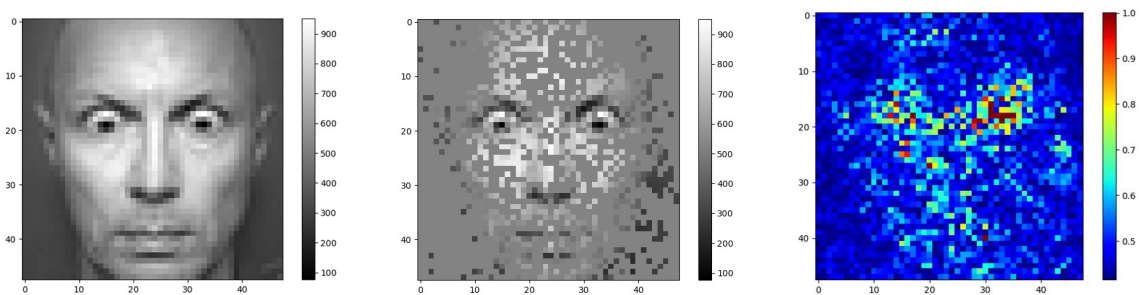
從圖表中(同樣 validation=10%)，發現我的：

- (1) 模型誤差最大的表情是 **Fear**，其次是 **Angry** 與 **Sad**，都是偏向負面且較誇張的表情。因為我的模型是著重**眼睛**部分的判斷，誤差最大的三者之間可能是在眼睛的差異上較小。
- (2) 誤差最小的是 **Happy**，再來是 **Surprise**，可以發現這些是較好分辨、獨特性較高的表情。包含眼睛與嘴巴的特徵都彼此較不相同，下題即選擇這兩個表情分析在模型中的辨識能力。
- (3) 比較意外的是模型對於 **Neutral** 的辨識度居然高居第三，因為我在實際看圖片的時候，反而覺得滿難判斷的，或許這類表情的眼睛特徵，與其他種類意外的較相近。

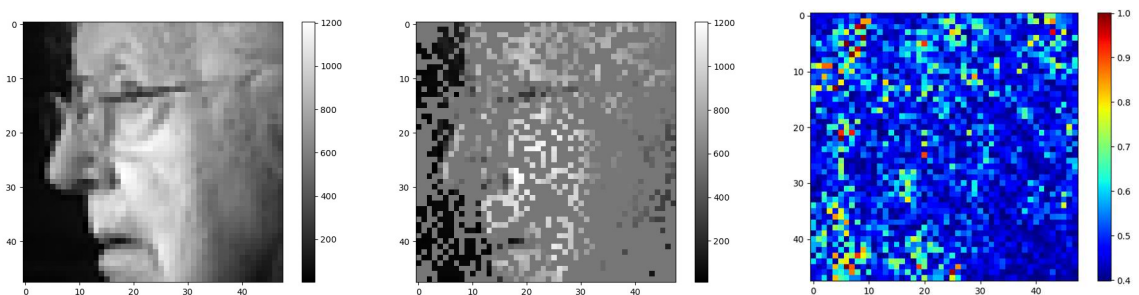
4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？



(Happy): 這是 model 中辨識度最高的表情，可以觀察出最著重的部位大約在眼睛的區域像兩側涵蓋到耳朵區域，而額頭以及臉頰的部分都可以算是幾乎忽略了。



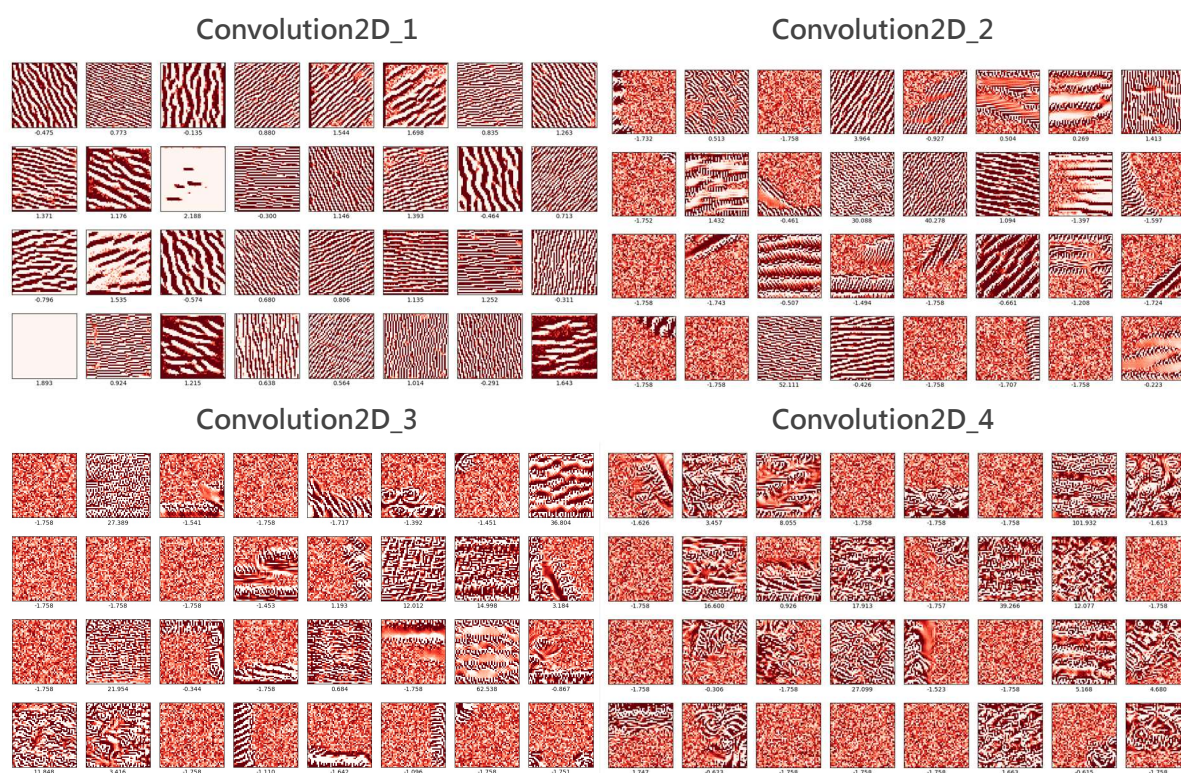
(Surprise): 辨識度第二名的表情一樣是最強調眼睛的區域，強調區域同樣也有包含到耳朵、鼻樑與嘴巴。



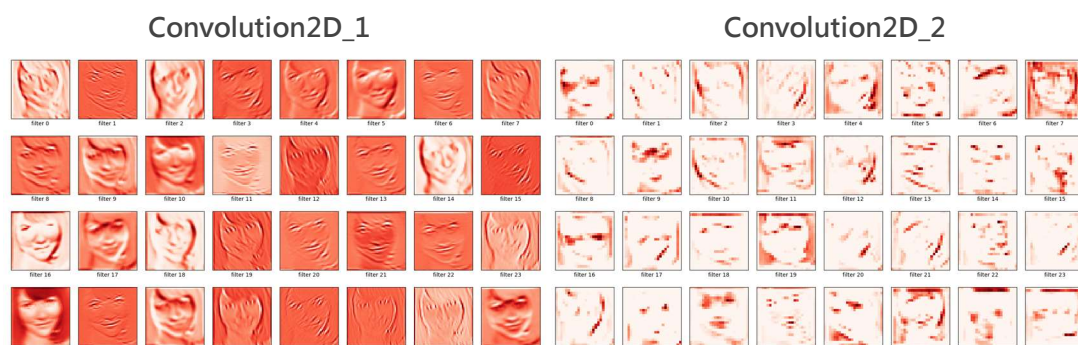
(Neutral): 選了一張側臉來測試，模型很厲害的也捕捉到了眼睛部分，但是最強調的部分跑到臉的外部，可見對側臉的辨識能力不佳。



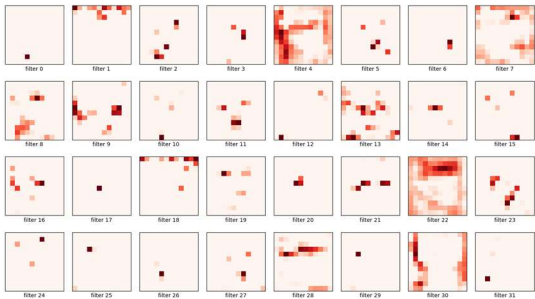
5. (1%) 承(1)(2)，利用上課所提到的 **gradient ascent** 方法，觀察特定層的 **filter** 最容易被哪種圖片 **activate**。



- (1) 可以發現第一層辨識的特徵都偏向粗條、簡單的紋理，有幾個 filter 甚至只有一點角度的差異。因此擁有粗條紋路的圖片較能夠 activate 第一層。
- (2) 到了後面幾層之後，特徵開始偏向越來越細的複雜圖形，包括一些海浪狀、細胞排列狀的圖形。可能是臉部較細的五官特徵較能 activate 後面兩層。Activate 情形如下列所示：



Convolution2D\_3



Convolution2D\_4

