

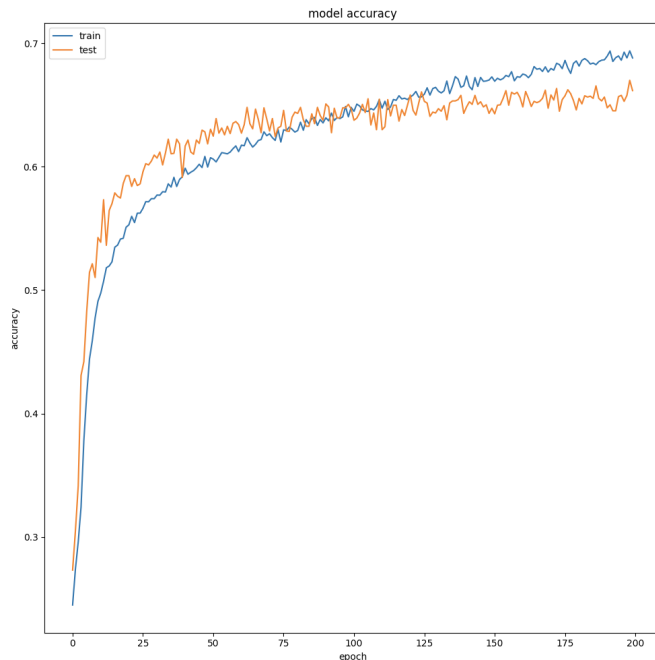
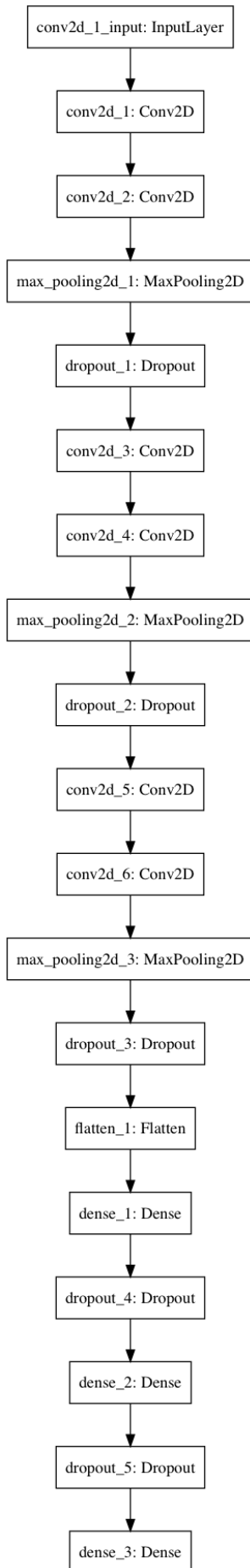
## HW3 Report

學號：R06942074 系級：電信所碩一 姓名：李宇哲

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: Only me)

從右圖可以看到我的 CNN model 的架構非常的長，光是 Convolutional layer 就有六層，然後裡面夾雜 3 個 max\_pooling 2D 和 drop out layer，最後用 3 層 FC 出去。我的 Optimizer 是用 adam，因為相較於 sgd 會比較快收斂。依照 kaggle 的 public score 可以到達 0.6949 的準確率，我的 0.1 validation score 也有 0.699。

下圖是我的 model 前 200 epoch 的 training procedure:

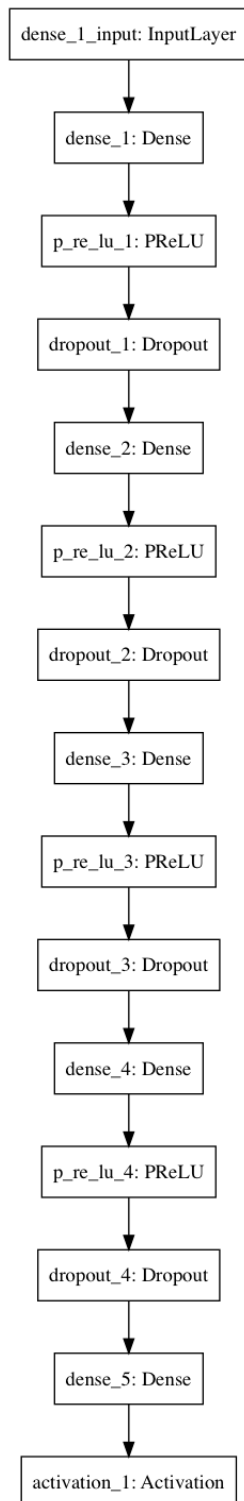


可以看出我的 validating score 在前 200 epoch 就有高達 0.66 的準確率，即使 training acc 還在 0.7 附近。

Total params 接近 5,000,000

```
activation_1 (Activation) (None, 7) 0
=====
Total params: 4,845,063
Trainable params: 4,845,063
Non-trainable params: 0
```

2. 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: Only me)

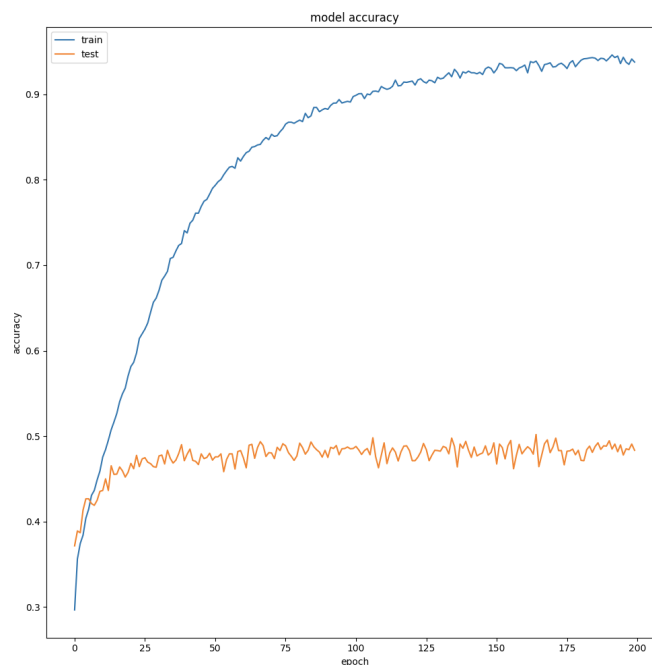


這裡我就稍微接了 5 層 DNN 的架構，總共的參數量也是接近 5,000,000，但是還稍微比 CNN 少！

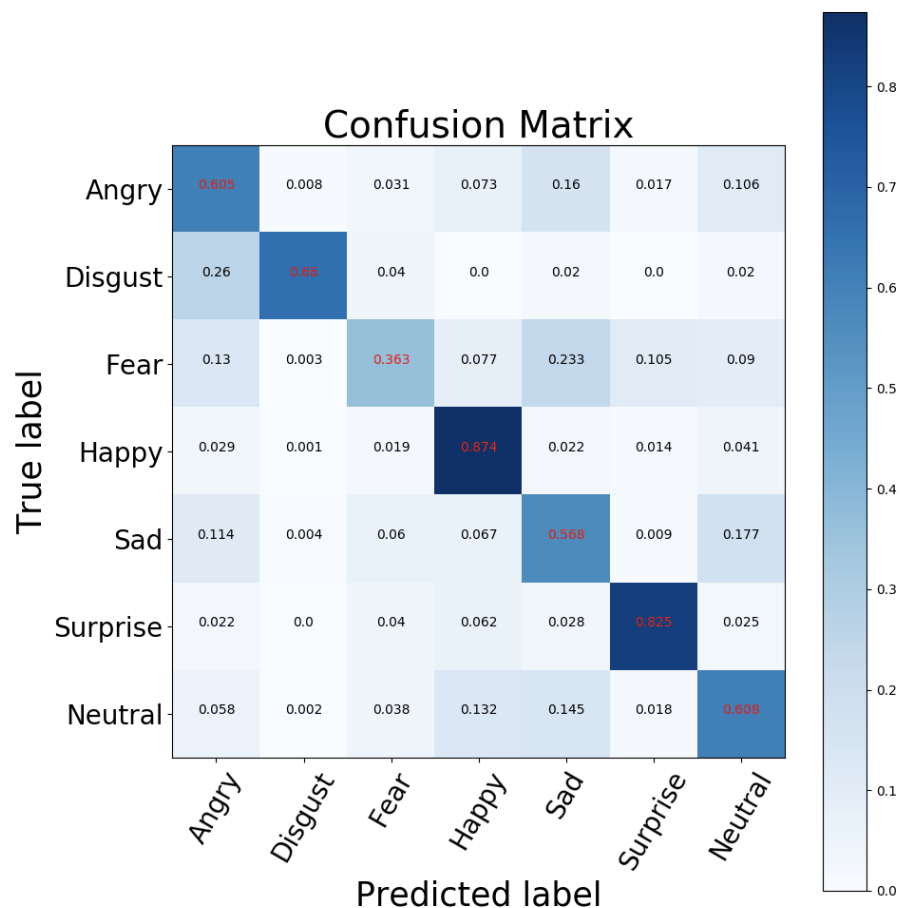
```

activation_1 (Activation)      (None, 7)      0
=====
Total params: 4,991,495
Trainable params: 4,991,495
Non-trainable params: 0
  
```

不過準確率和 training procedure 如下，一樣都是經過前 200epoch，看得出來 DNN 的 training acc 已經可以馬上超過 0.9，但是 validation 的 acc 基本上已經停留在 0.5 以下上不來，所以 DNN 的 performance 基本上幾乎沒有 CNN 好。



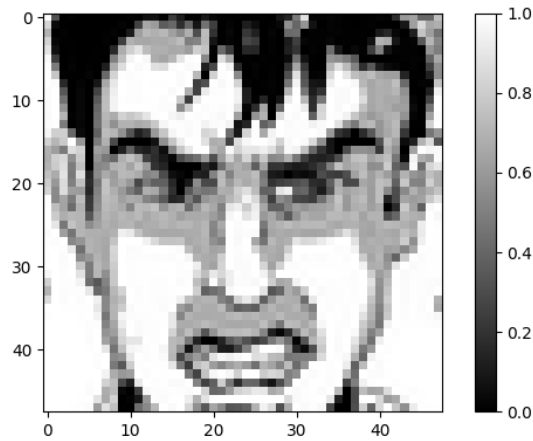
3. 觀察答錯的圖片中，哪些 class 彼此間容易用混？ (Collaborators: Only me)



把我們最後 model 的 confusion matrix 印出來可以發現，Happy 和 Surprise 這兩個 class 最後分對，而且其他 class 被誤判成這兩個 class 的機率都不高，然後 Fear 的狀況最慘，基本一半被分到 Sad 和 Angry 的地方，或許 Fear 的表情真的跟 Sad 和 Angry 很像。其實從這裡可以看出有些表情難免會有誤判和相似之處。

4. 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: Only me)

這裡我是用 training dataset 裡面的 38 號圖當作範例：

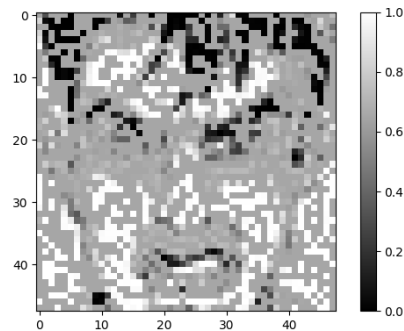
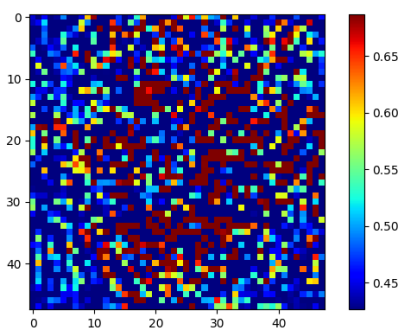


(左方這張是原圖)

這張是屬於生氣的 class

當我做完 heatmap 後可以得到下面兩張圖，下面兩張圖其實都是這張圖在我的 model 上的 heatmap，只是 normalization 和 clip 的方法不同，不過最後得到的結

果都是下面那張 mask 掉的圖。

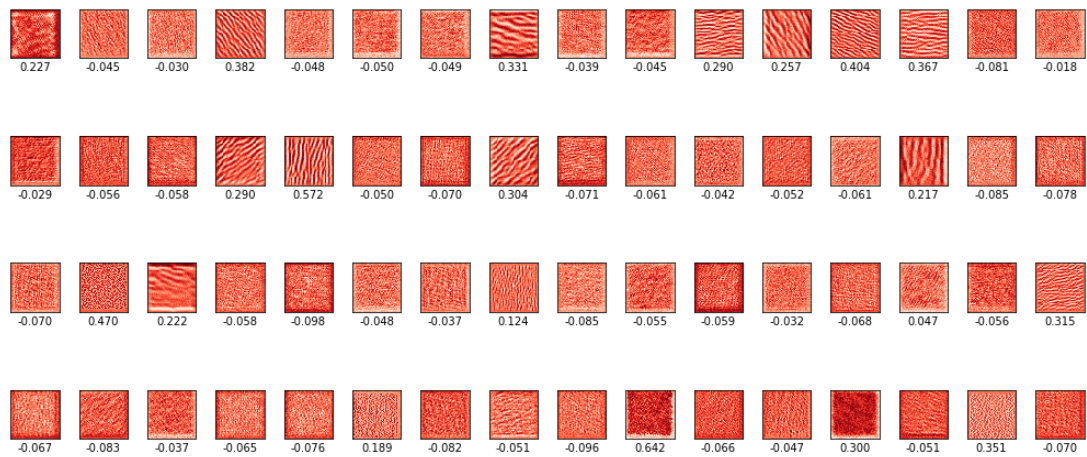


看起來生氣的這張圖，model 主要 focus 在額頭嘴吧跟臉頰的地方，其實算是蠻合理。

5. 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: Only me)

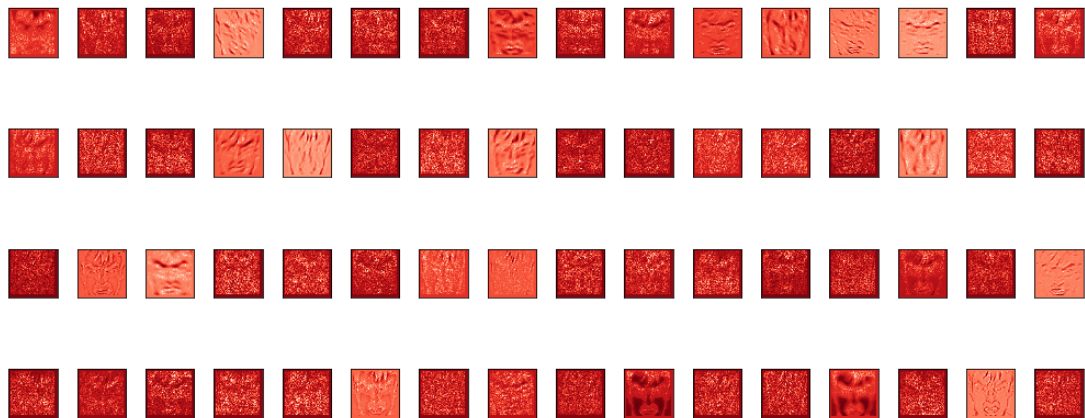
我拿 zero\_padding2d\_1 的第一層出來看，是因為從 layer output 可以觀察到和原本圖片的相似特徵。不過為了要利用 gradient ascent 看到哪些圖片可是是這一層最 activate，用了 random input 去經過大概 180 epoch 的結果如下：

Filters of layer zero\_padding2d\_1 (# Ascent Epoch 180 )



可以看出每一個 zero\_padding2d\_1 都有不同能夠被最 activate 的圖片

Output of layer0 (Given image38)



這個是把我們這張生氣的圖片喂進去產生的 gradient 結果