學號：B04902053 系級：資工二 姓名：鄭淵仁

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？**

**答：**

* **模型架構：**

最後的總參數量是882,151。

* **訓練過程：**

我訓練了很多組model，每一組model的架構都一樣，只有參數有些微變化。最後我把這些model predict出來的機率加起來，輸出機率最大的表情。

其中，對於每一組model，我的架構都是：先對所有圖片**前處理**，接著使用上面描述過的模型架構**訓練參數**，之後**處理testing data**，再**predict**出結果。

圖片前處理的作法如下：

1. 對所有圖片（training data、testing data）以單一圖片為單位做**normalization**，以減少bias。
2. 使用keras的Image Data Generator，在每個epoch前隨機對圖片做**鏡像**、**旋轉**、**裁切**等操作，以增加資料量。

而在predict testing data之前，我先把testing data也生成出**上下左右移動**以及**旋轉**等多組data，再用這些data一一predict出個別表情的機率。接著把機率加起來，使用最大機率的表情做為結果輸出。

* **準確率：**

單一個model訓練出來的結果在training set上的accuracy是0.8135，在自己切的validation set上的accuracy是0.6714。最後把多個model加總起來之後，在validation set上的accuracy是0.6838。

在只有一個model的訓練過程中，各個epoch的accuracy變化如下圖所示：

1. **(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？**

**答：**

* **模型架構：**

總參數量則是987,911，大約等於第1題CNN的訓練參數量（882,151）。

* **訓練過程：**

為了能和上一題CNN的數據比較，我的訓練過程和DNN的處理方式一模一樣，也就是：一樣生出多個model再把機率加總起來取最大值、一樣有圖片前處理、一樣在predict前處理test data。

* **準確率：**

訓練出來的結果在training set上的accuracy是0.6546，在自己切的validation set上的accuracy是0.4307。

訓練過程中，各個epoch的accuracy變化如下圖：

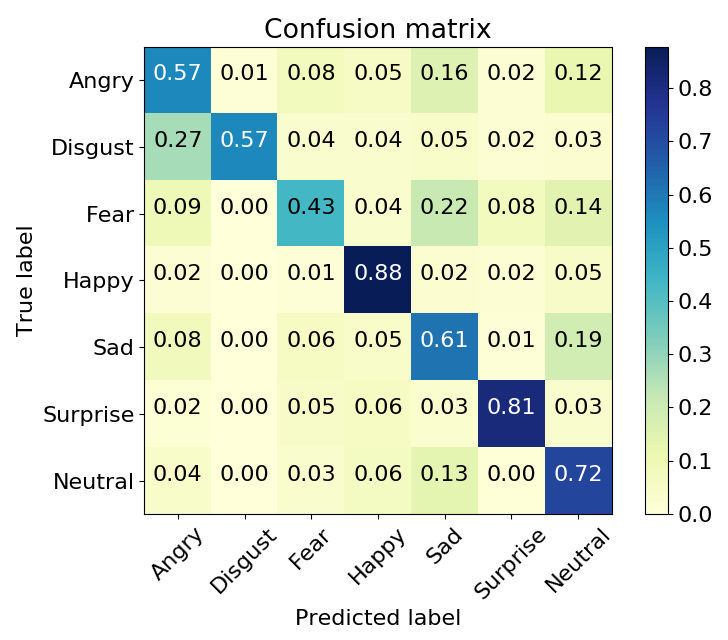
* **觀察到了什麼：**

我觀察到：DNN在training set和validation set上面的正確率明顯都比CNN低，上升速度也慢很多。

1. **(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]**

**答：**

下圖是我在validation set上predict之後，輸出的結果與正確答案的confusion matrix：



從圖中可以發現：許多原本label是disgust的圖片，被辨識成angry，其次則是fear被辨識成sad。

但是原本被label成disgust的圖片只占所有圖片中的1.5%左右，所以disgust的辨識會比較不準，可能不具有代表性。

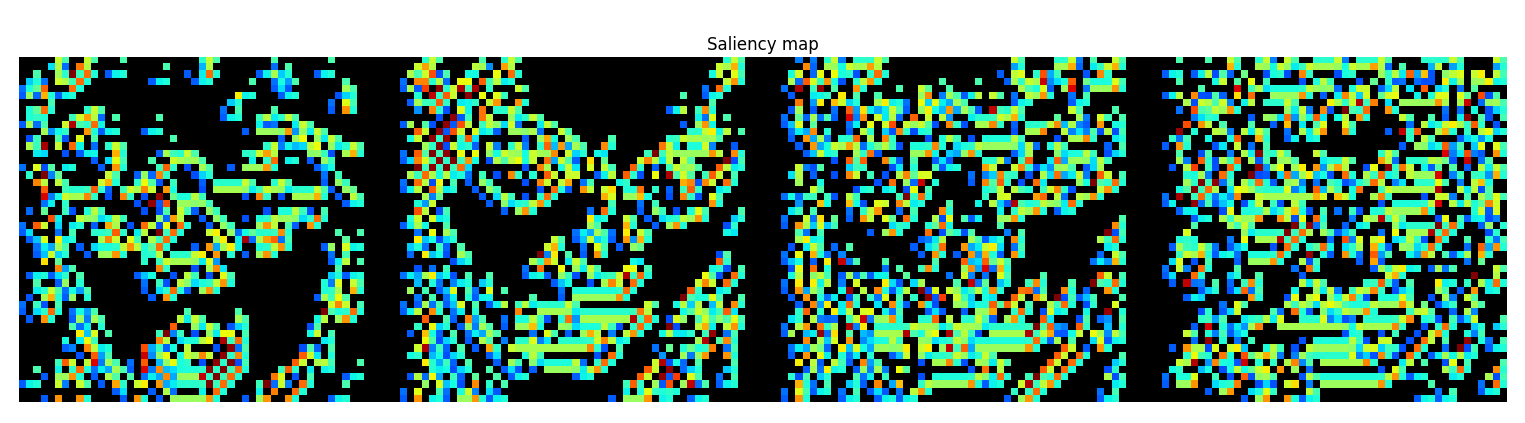
相較之下，fear和sad都有一定數量，如果有更多的data，這兩者容易用混的程度可能比disgust和angry要高。

1. **(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？**

**答：**

我使用keras vis的套件，針對第一層convolution layer畫saliency map，畫出的結果如下：

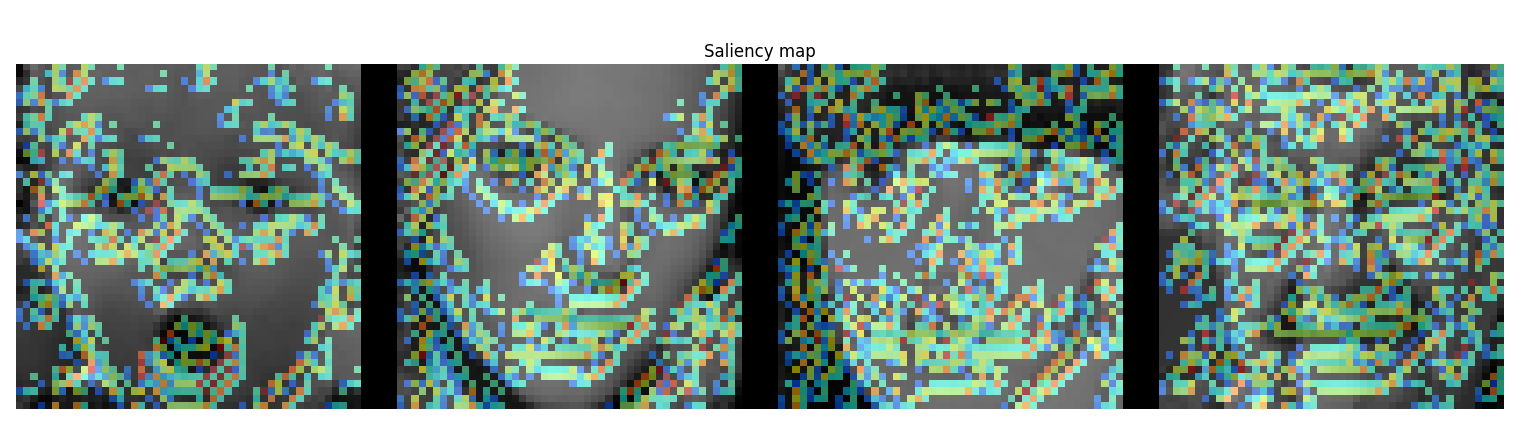
* 畫出的saliency map：



* 原圖：



* 將saliency map和原圖結合：

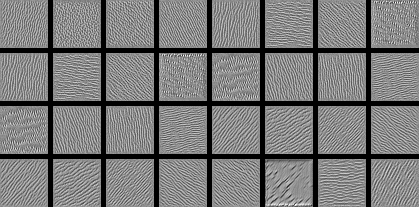


觀察上面的圖可以發現：我的模型在做classification時，主要會focus在圖片裡面人的五官以及臉部輪廓。但是如果圖片上被蓋了浮水印、或是臉部皺紋很多，就會影響到模型focus的部分。

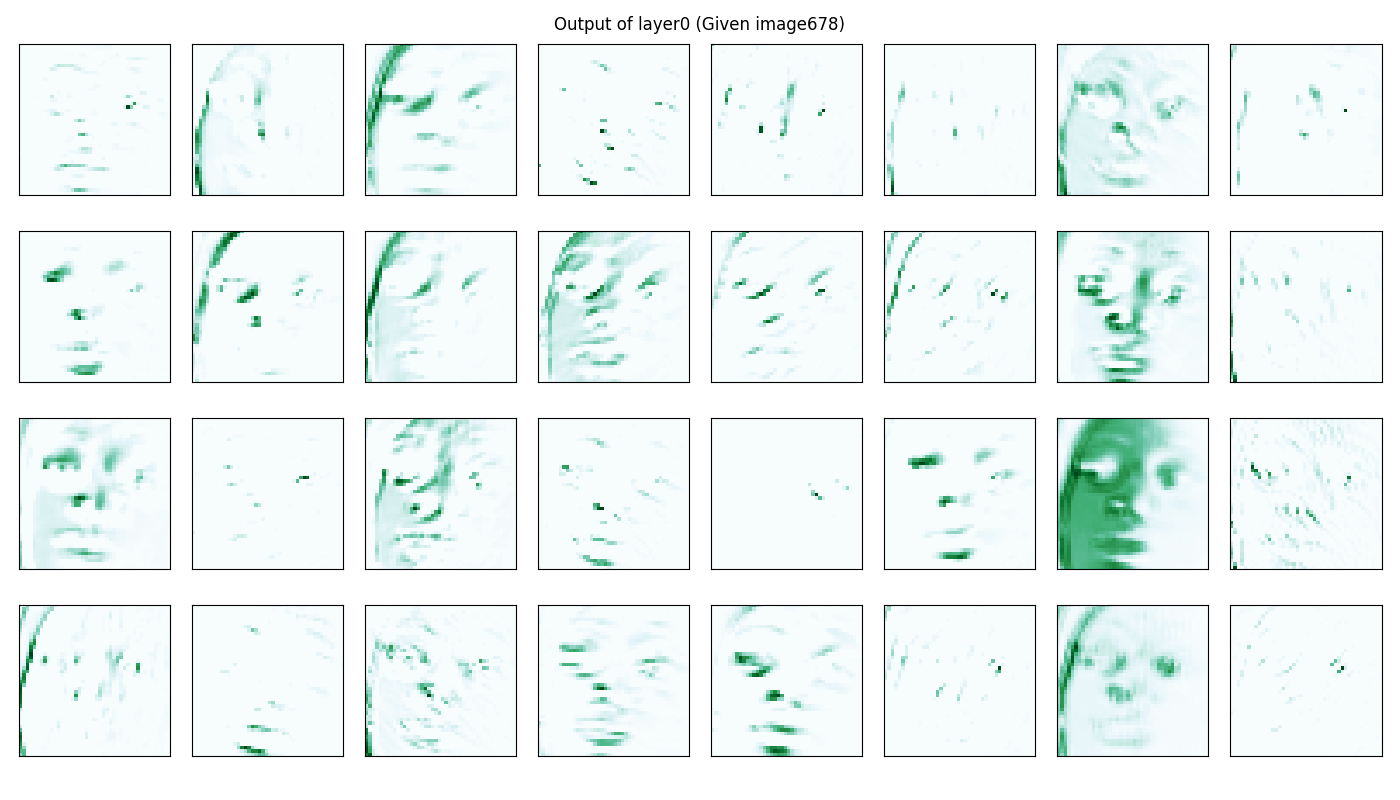
1. **(1%) 承(1)(2)，利用上課所提到的gradient ascent方法，觀察特定層的filter最容易被哪種圖片activate。**

**答：**

我取第2層convolution的filter出來觀察，最容易activate這一層filter的圖片如下圖所示：



把圖放進filter後，產生的結果如下圖所示：



由上面兩張圖，我觀察到由於是前面層數的filter，所以filter基本上還是在辨識直線和曲線。而在不同filter裡面，人臉中不一樣方向的曲線也被辨別出來了。很有可能在後層的filter的時候，就可以讓程式用這些曲線去觀察人臉上不同的特徵。

**[Bonus] (1%) 從training data中移除部份label，實做semi-supervised learning**

我先切1400個圖片作為validation set，剩下的90%的資料去掉label。

只使用training data，再使用CNN訓練100個epoch，在validation set上面的正確率是0.472142。

而如果把沒有label的資料也加進來用semi-supervised learning的話，正確率則上升到0.480714。

semi-supervised learning的訓練過程中，各個epoch的accuracy變化如下圖：

**[Bonus] (1%) 在Problem 5中，提供了3個hint，可以嘗試實作及觀察 (但也可以不限於hint所提到的方向，也可以自己去研究更多關於 CNN 細節的資料)，並說明你做了些什麼？ [完成1個: +0.4%, 完成2個: +0.7%, 完成3個: +1%]**