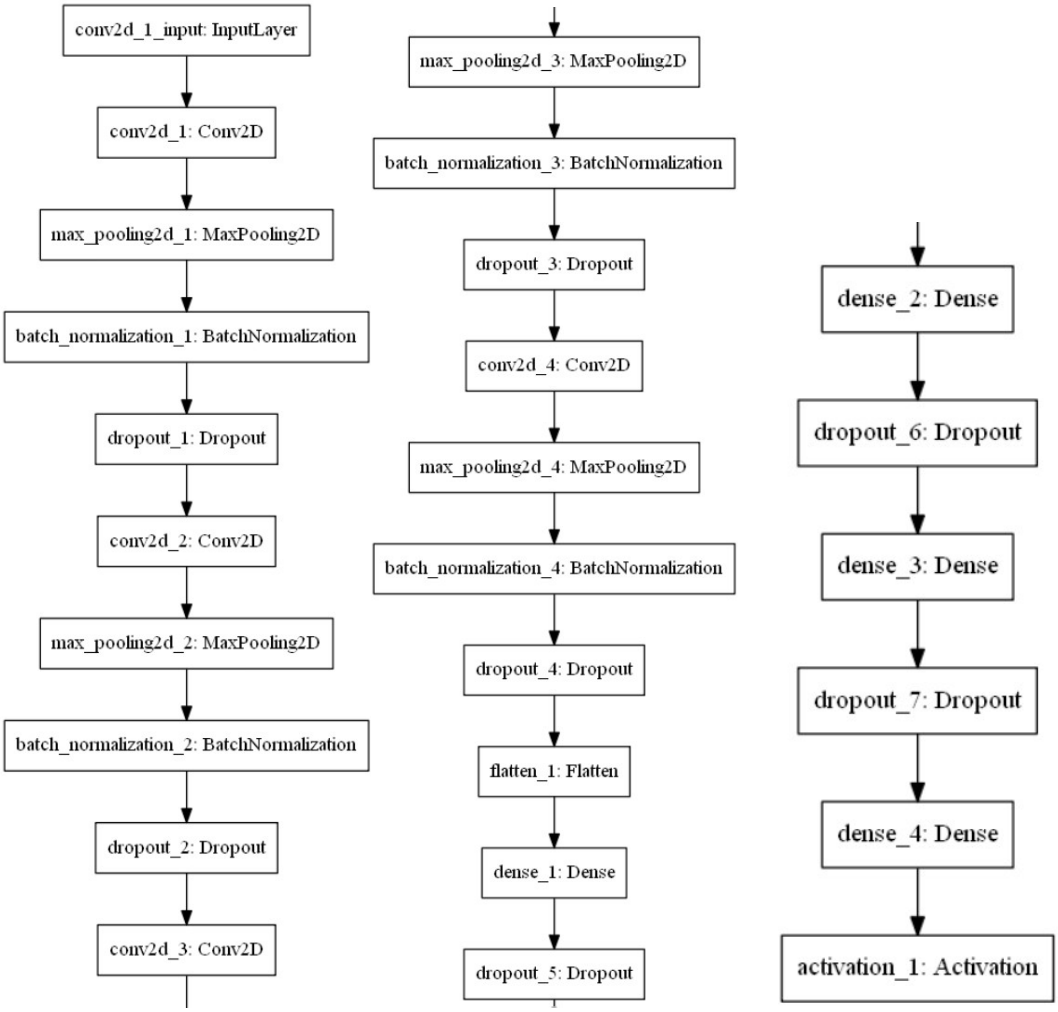
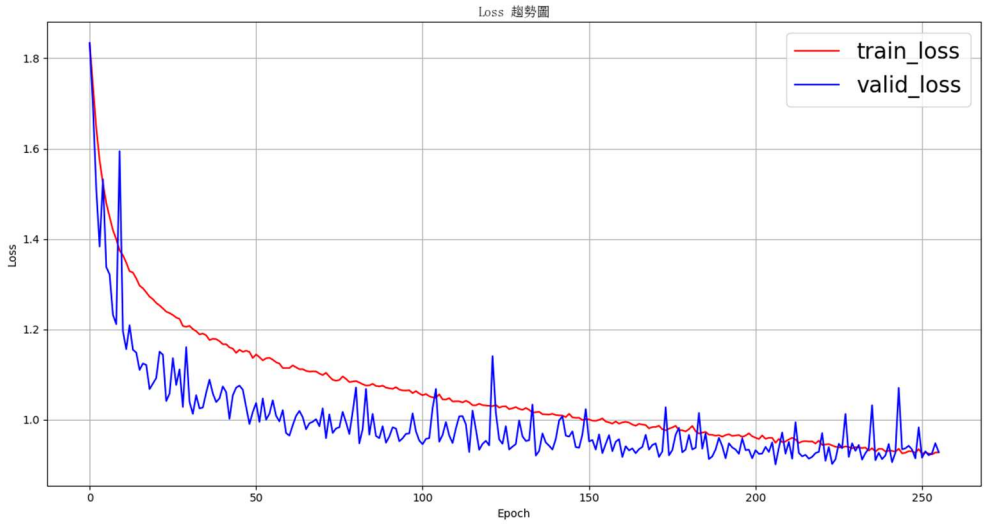
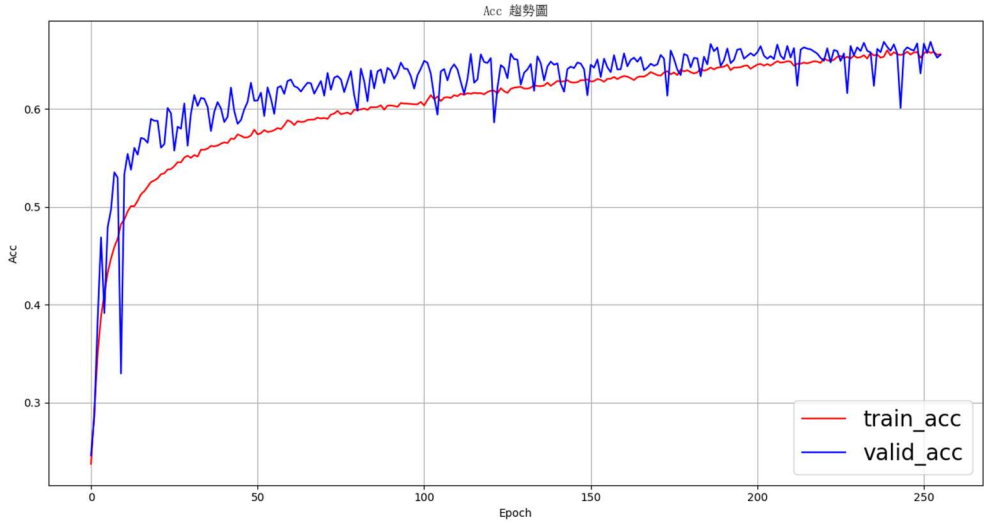
學號：B04902099 系級： 資工三 姓名：黃嵩仁

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？
(Collaborators: 有跟資工三 林政豪同學討論過做法，也有參照上學期修課的宋子維同學的做法)
答：模型架構如下圖：疊了 4 層的 Convolution layer (filter 數依序為 64, 128, 256, 512，activation function 皆為 relu，除了第一層的 filter size 為(5, 5)外，其餘 filter size 都為(3, 3))，每層 Convolution 後都緊接著 Maxpooling (pool_size 為(2, 2))跟 BatchNormalization，並設定 dropout 為 0.3。再 flatten 完後，加上 4 層的 Dense (units 分別為 512, 512, 64, 7，activation function 皆為 relu)，並設定 dropout 為 0.3。



訓練過程方面，訓練前，在將資料處理成可接受格式後，做 normalization (除以 255)，並取出最後 5000 筆作為 validation data (取之前有先 shuffle 過整個 training data)，之後將 training data 經過左右翻轉(照常理而言左右翻轉後情緒應該一樣)後，concatenate 在原 training data 後方。並使用 Keras 的 ImageDataGenerator 處理照片 (旋轉、平移、翻轉等功能)。
準確率方面，可參見下方之 Loss 與 Accuracy 趨勢圖，由圖可觀察到 validation data 的 loss 與 acc 震盪幅度都比 training data 來的大。(最後一個 epoch 的 validation loss = 0.928413, validation accuracy = 0.655600)





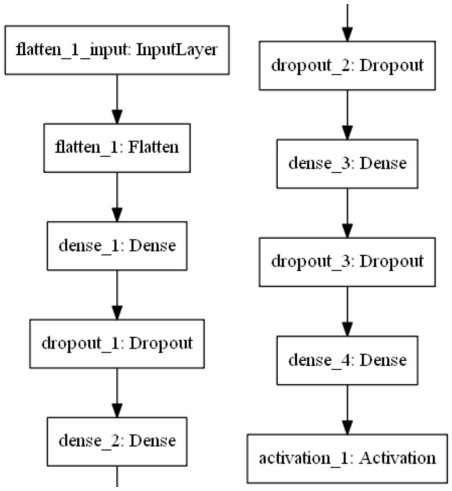
2. (1%) 承上題，請用與上述 CNN 接近的參數量，實做簡單的 DNN model。其模型架構、訓練過程和準確率為何？試與上題結果做比較，並說明你觀察到了什麼？

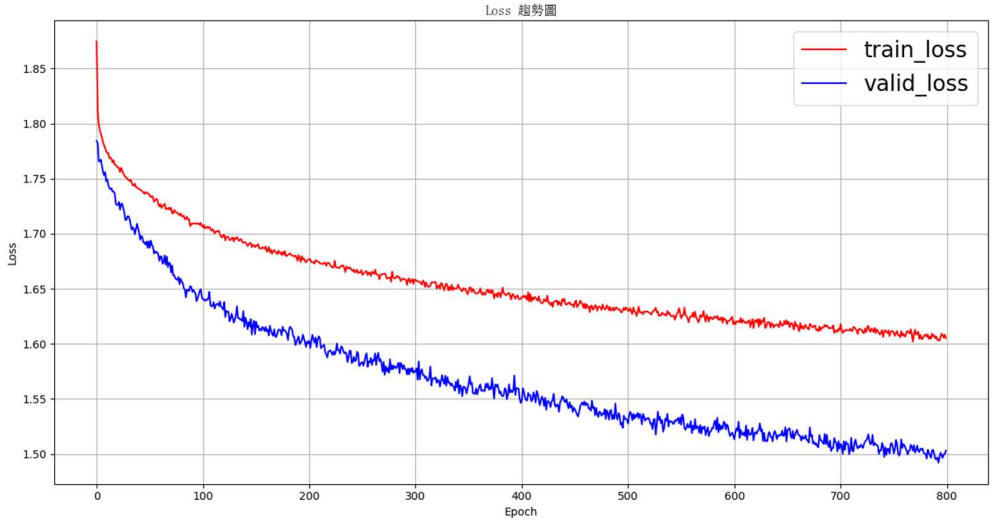
   (Collaborators: 有跟資工三 林政豪同學討論過做法)

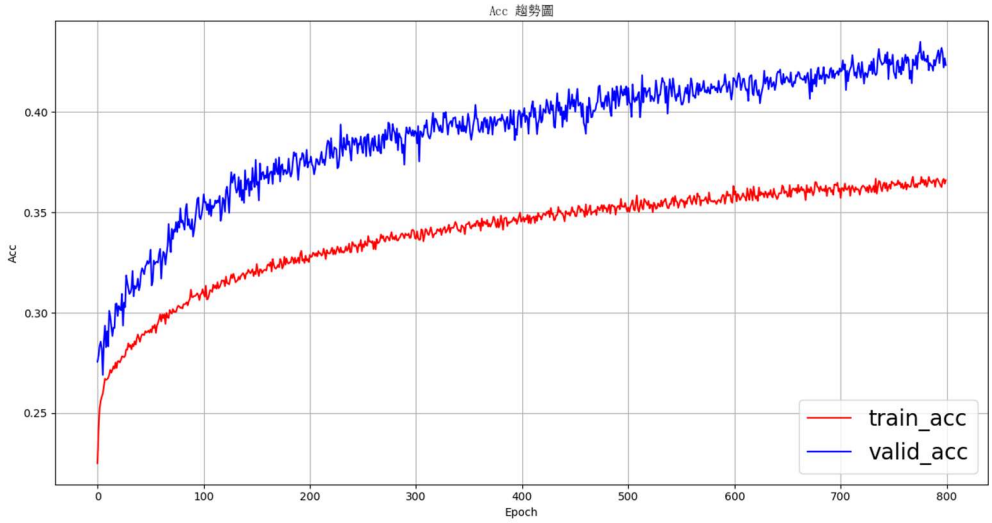
   答：上題中的 CNN 模型中總參數量為 2,113,287 (Trainable：2,111,367；Non-trainable：1920)，實作的 DNN model 總參數量為 2,051,647 (Trainable：2,051,647；Non-trainable：0)。模型架構如下圖，將 Input flatten 後，建立 4 層的 Dense (units 數分別為 720, 360, 360, 7，activation function 為 relu)，並設定 Dropout 為 0.3。

   訓練過程方面，訓練前的處理與第一題的處理方式完全一樣。

   準確率方面，觀察下方的 Loss 與 Accuracy 趨勢圖，可以發現：相同 epoch 數時，不論在 Loss 或是 Accuracy 的表現，CNN 都比 DNN 好；即使在兩者都接近收斂的情況下(CNN：256 epoch；DNN：800 epoch)，CNN 的訓練結果也都比 DNN 好很多。(但以訓練速度而言，跑一個 epoch，DNN model 比 CNN model 快了好幾倍)



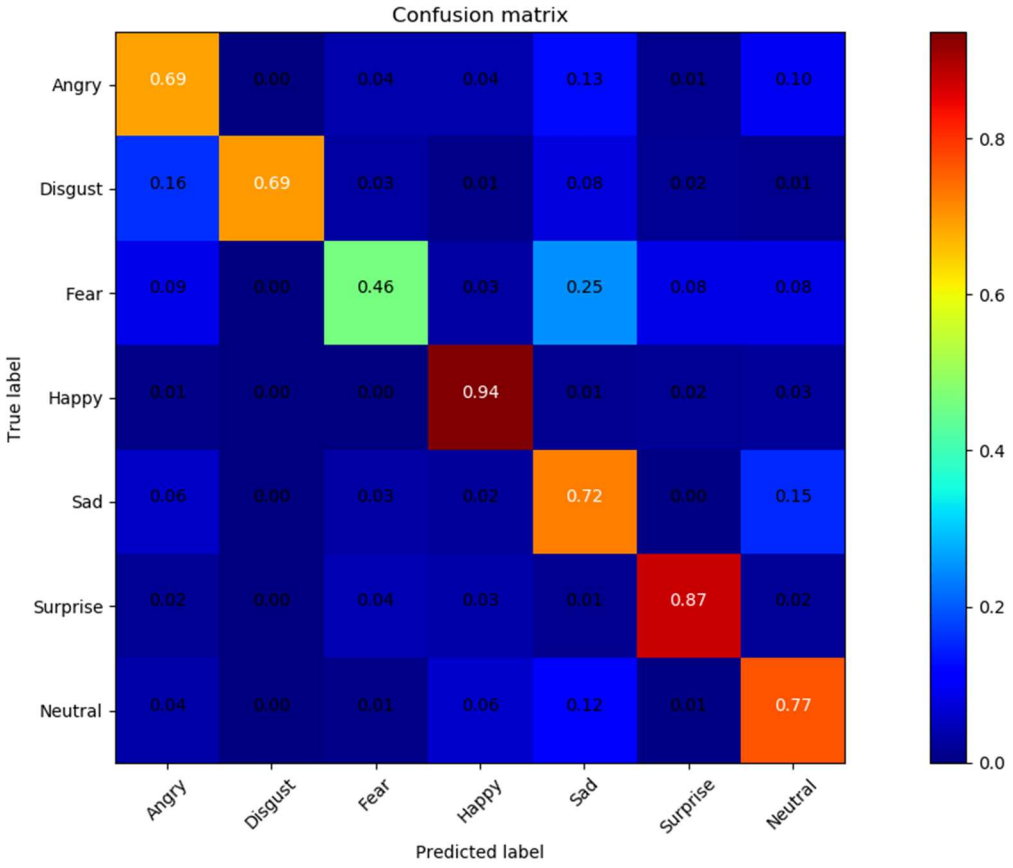




3. (1%) 觀察答錯的圖片中，哪些 class 彼此間容易用混？[繪出 confusion matrix 分析]
(Collaborators: 有參考上學期修課的宋子維同學做答方法)

答：觀察下方的 confusion matrix 可以發現，我所訓練的 model，在判斷"***Fear***"這種情緒的照片時，準確率明顯偏低。當照片的 True label 為 Fear 時，只有 46%的機率被我的 model 判斷為 Fear，且有高達 25%的機率被判斷為 Sad，然而 True label 為 Sad 但 Predict label 被判斷為 Fear

的機率只有 3%，可見我所建立之 model 認為一張照片是 Sad 的機率比是 Fear 的機率高。(此外，也觀察到 Sad 跟 Neutral 兩種 class 之間互相混淆的情況也比其他 class 嚴重，混淆的機率為 12%~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: 參考上學期修課的陳家棋同學的回答)

答：下圖三排是三張不同圖片之原圖、Saliency Map、Mask 掉 heat 小的部份所生成之圖片。

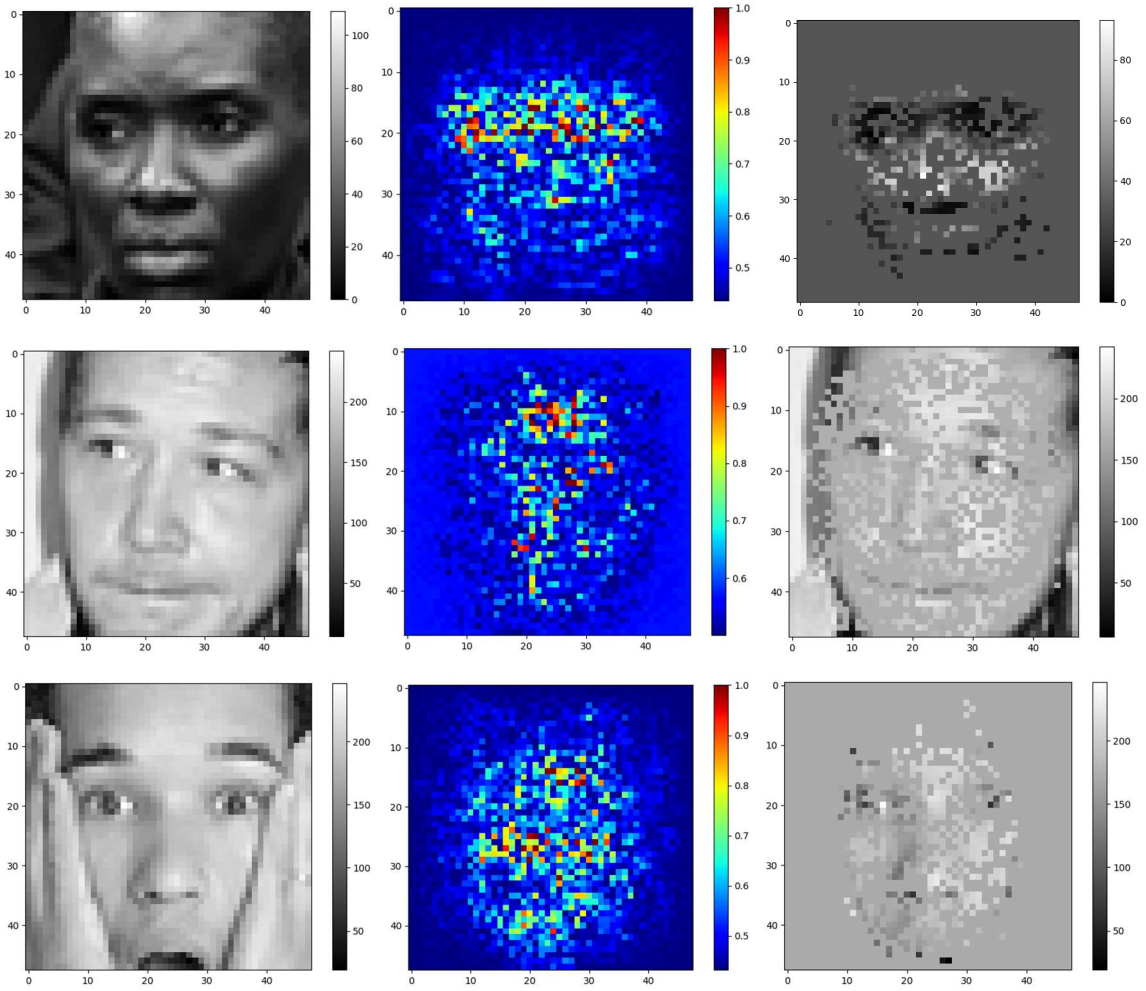
第一排圖片，可清楚的觀察到 Saliency Map 中，眼睛是整張圖片 heat 較高的部位，可猜測 model 會針對眼睛區域進行判斷。

第二排圖片，可看見 Saliency Map 中，雙眼、鼻子與嘴巴構成的 T 字部位 heat 較高，但不知道為甚麼額頭中央區域 heat 也極高。

第三排圖片，可看見 Saliency Map 中，heat 高的區域為眼睛、鼻子，而在 Mask 後的圖片中，除了眼睛與鼻子外，可看見嘴巴的上半部 (可能原因為這張圖中的人臉，嘴巴下半部已經超過邊界，但 model 有辨認出部分嘴型)

總結，模型在做 classification 時，眼睛、鼻子、嘴巴都有一定的重要性(若可清楚辨識的話，eg. 沒有被手遮蔽 or 超出邊界)

(a)原圖　　(b) Saliency Map　　(c) Mask 掉 heat 小的部份

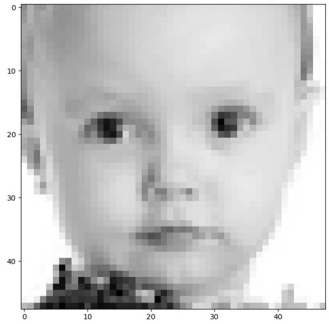


5. (1%) 承(1)(2)，利用上課所提到的 gradient ascent 方法，觀察特定層的 filter 最容易被哪種圖片 activate。

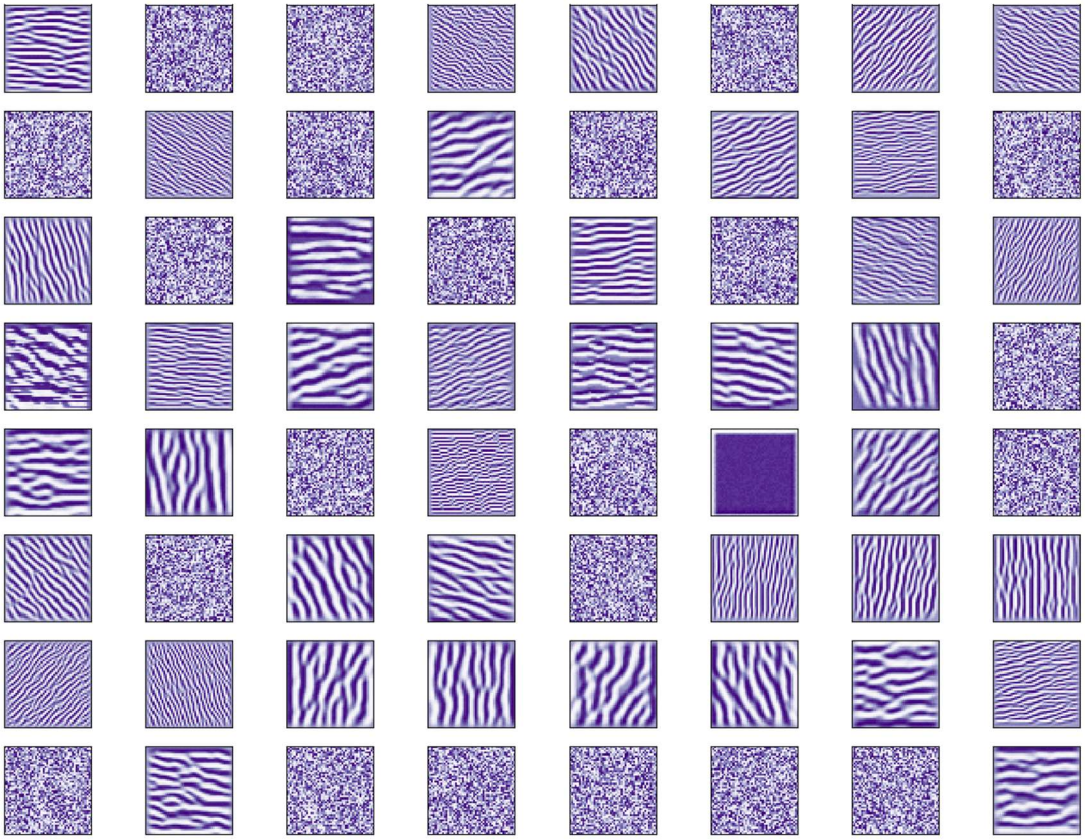
(Collaborators: 參考上學期修課的陳家棋同學的回答)

答：下方圖片分別為原圖、Conv2d_1 之 filters (取第 160 個 epoch)、以及原圖通過 Conv_2d filters 之後的結果，觀察結果可發現該圖通過 filter 的結果中，人臉的輪廓(五官)較為明顯，可推測 Conv2d_1 之 filters 較容易被五官清晰的圖片 activate。(此外，有部分的結果為空白圖片(值皆為 0)，推測為 Convolution2D function 中" relu" 造成的結果。)

原圖：



Conv2d_1 之 filters (取第 160 個 epoch)：



原圖通過 Conv_2d filters 之後的結果：