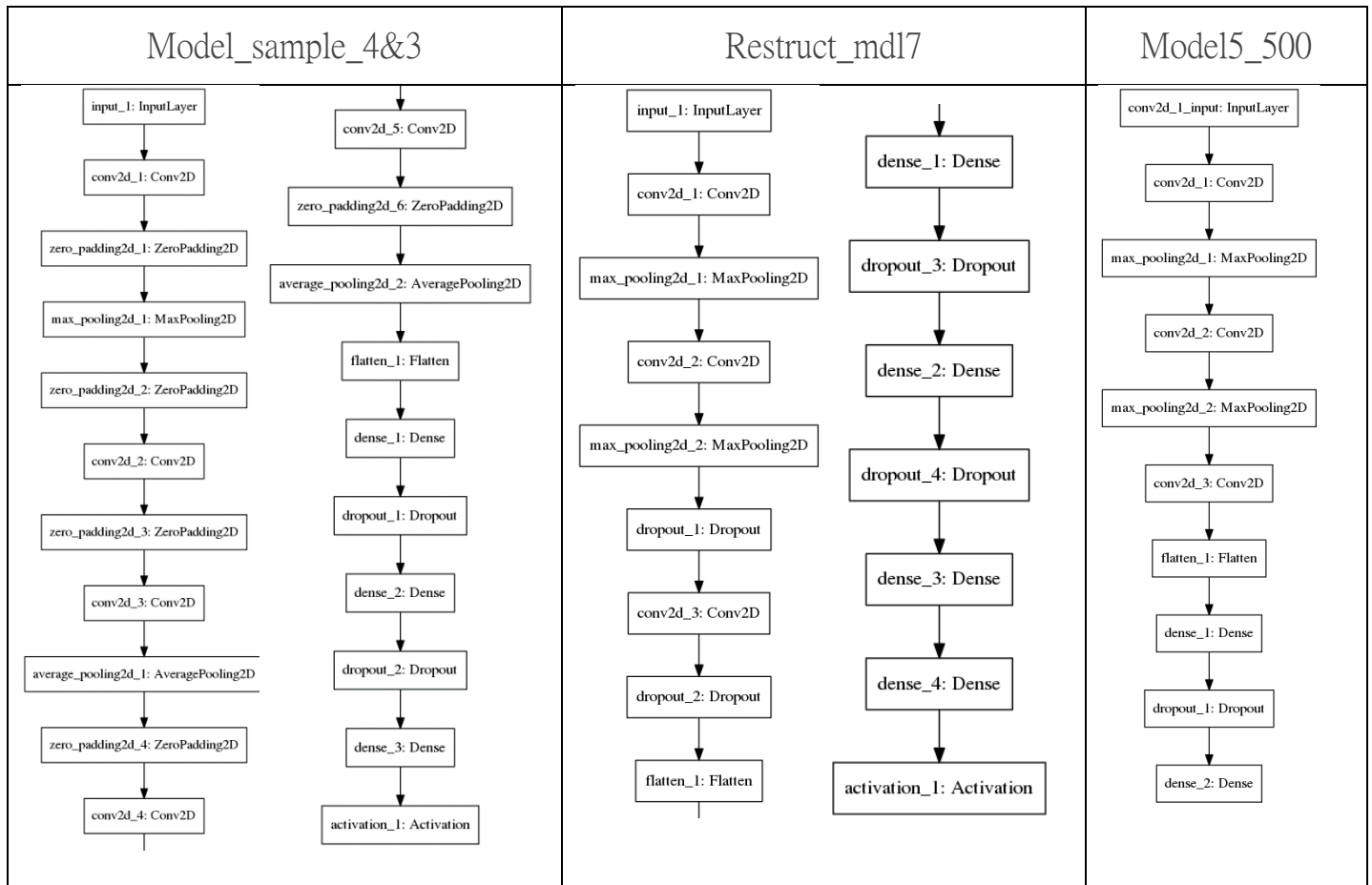


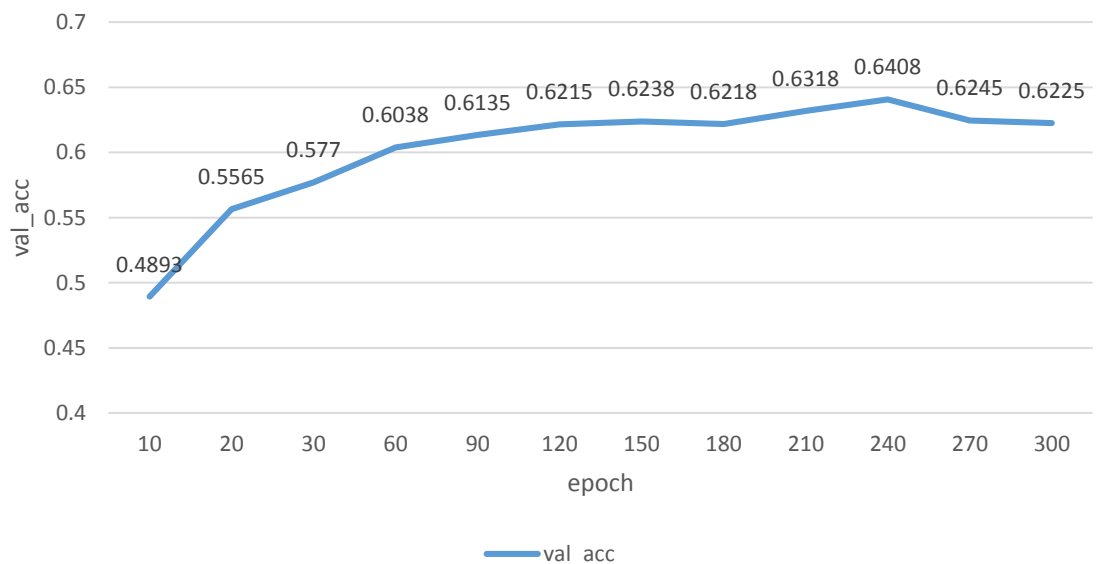
1. (1%) 請說明你實作的 CNN model, 其模型架構、訓練過程和準確率為何？

(Collaborators: TA time sample code)

答：我設計的模型共有四個(註:其中一個參考 sample code), 並依其在 kaggle public 上的分數做加權再 ensemble。以個別準確率最高的 TA sample code 改寫為例: 將模型架出來，然後每隔五個 epoch 更新一次 training data 的順序，另外加上 imgdatagenerator 輔助。以下為四個模型架構與訓練過程:



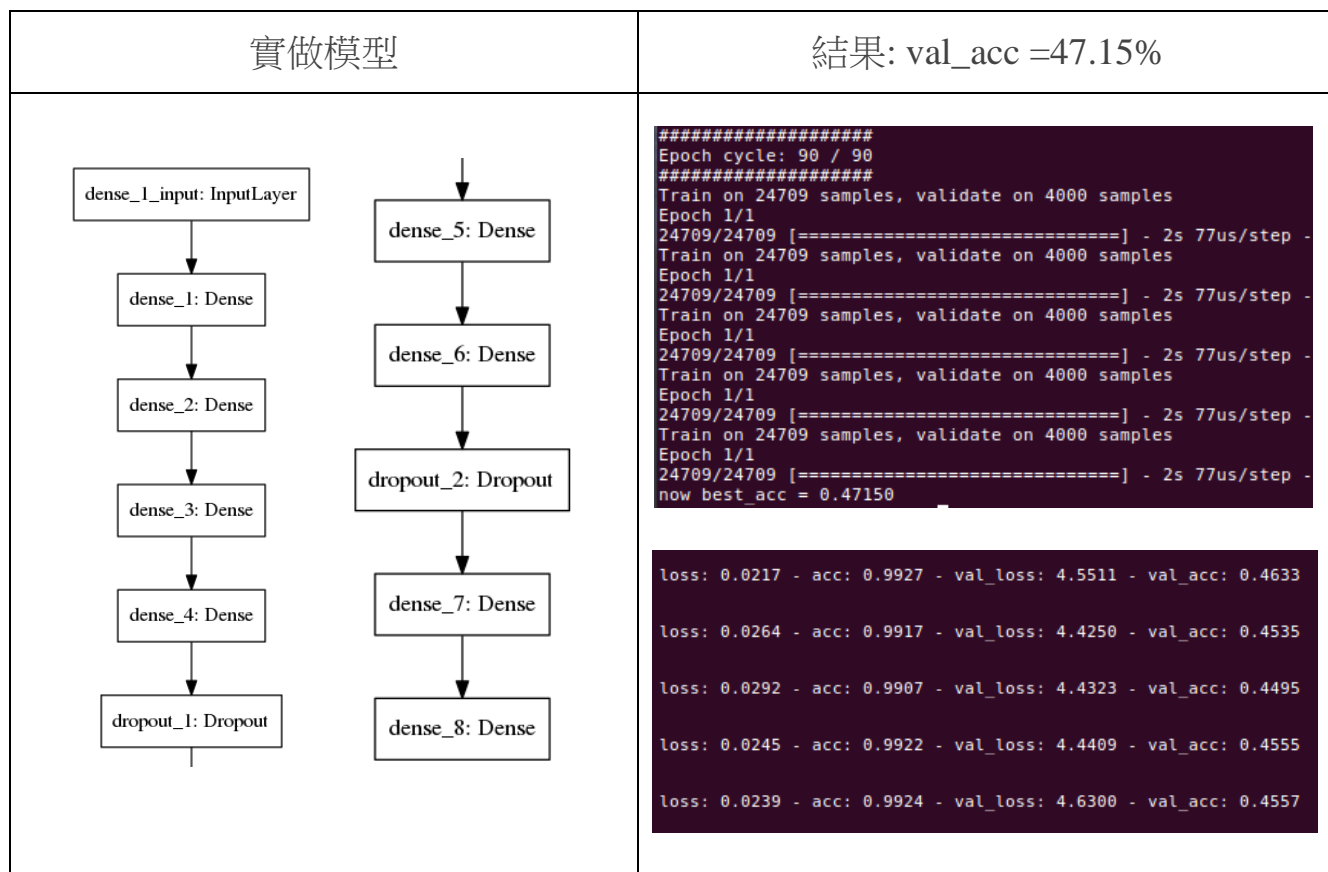
訓練過程: 以model_sample_4為例



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？

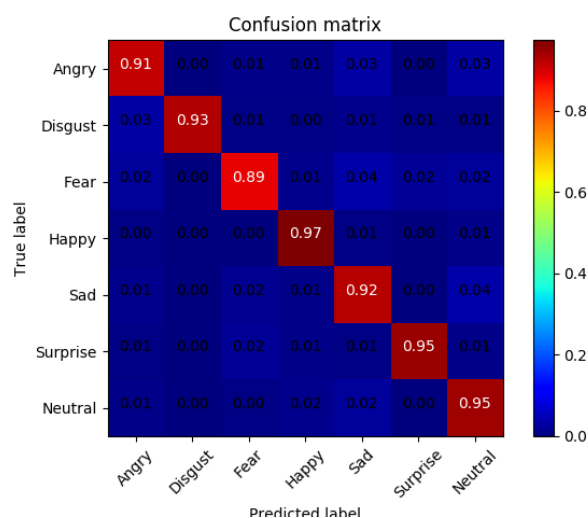
(Collaborators:)

答：實做模型與結果如下圖。訓練過程較容易 **overfit**，**val_acc** 上升緩慢，而正確率只有約 **45%**，遠不及用 **conv2D** 所做出的結果。其因可能為無法判斷二維度的特徵，因而無法抓出最關鍵的判斷依據。不用 **conv2D** 做的話更容易 **overfit**，在訓練相同 **epoch** 下其 **training acc** 很快來到 **90%**，而有 CNN 則會大概一直維持在 **75~80%** 左右。



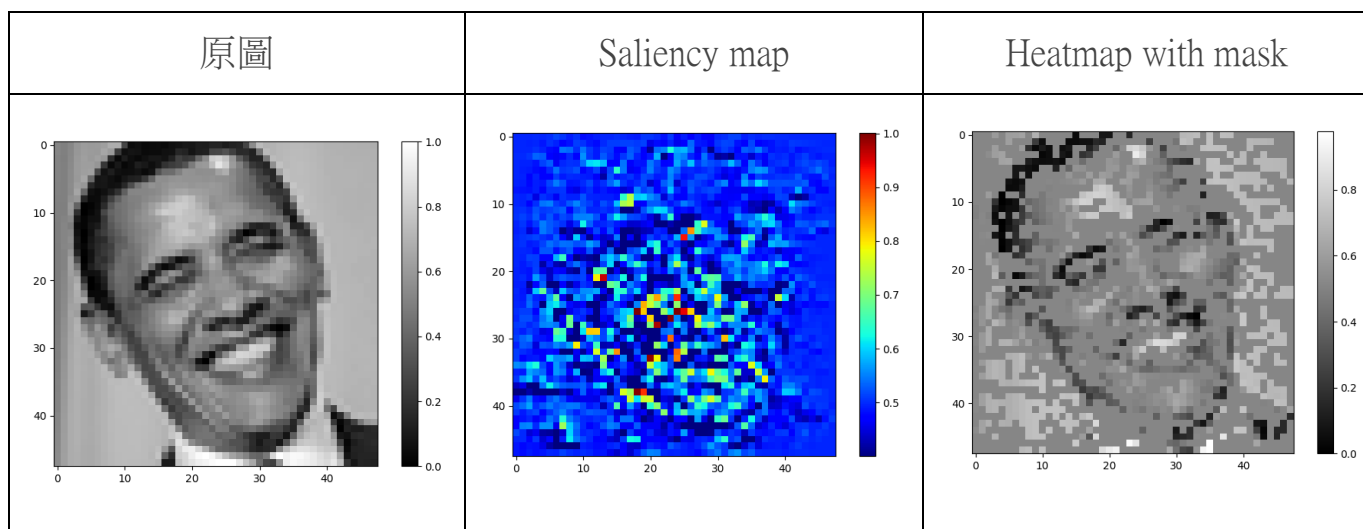
3. (1%) 觀察答錯的圖片中，哪些 **class** 彼此間容易用混？[繪出 **confusion matrix** 分析]
(Collaborators:)

答：以下為我其中一個模型的 confusion matrix。最易混淆的應該是 sad 很容易被誤認為 fear，其誤認比率到答 4%。此外，sad 也很容易被誤認成 neutral，開啟錯誤判別的圖片之後發現確實有許多特徵相似，可能人也分不出來。整體而言，fear 最容易被誤認為別的情緒(錯誤率: 0.11%)，而 happy 的準確率最高，觀察 labeled happy 的圖的 Saliency map 就會發現其特徵相對加權較明顯。



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:)

答：以下是用編號 10000 的圖片進行的 saliency map 和 heatmap mask 分析。依據觀察，辨識加權最重的是嘴巴的形狀(與其邊界)，另外眼睛邊界也是一個重點。此為 labeled happy 的圖，然觀察其他的圖，其重點加權也都擺在面部表情的相關位置(如嘴巴，眼睛等等)，與一般我們認知的辨別非常類似。值得注意的是，其他 label 的 Saliency map 並不像 happy 的一樣這麼區分這麼鮮明，像是 neutral 的分變就沒有這麼明確。



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:)

答：以下以編號 10000 的圖片進行的分析，取 layer_0 與 zero_paddling_4 為分析依據。觀察發現在特定 label 下有關其特徵的 filter 會讓該圖片的相關數直保留下來。而當我們丟入一張隨機的圖形的時候，各個 filter 就會把自己代表的相應的特徵表現出來(觀察到就加權)。以我的圖為例，可以看到第二張圖的最後一排第 8 個 filter，很明顯就是用來偵測笑容用的，其他的 filter 也有他們偵測的特徵。

