

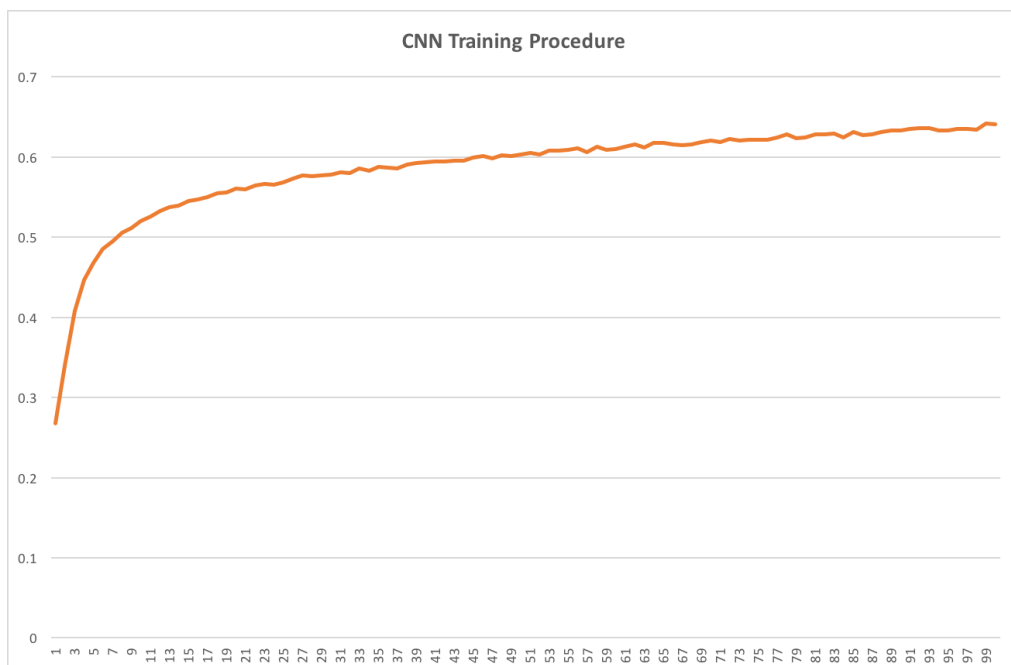
1. 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何? (1%)

答：

使用的 CNN 模型架構如下表：

層次	Type	Param
第 1 層	Conv2D	320
第 2 層	Conv2D	18496
第 3 層	MaxPooling2D	0
第 4 層	Conv2D	36928
第 5 層	MaxPooling2D	0
第 6 層	Conv2D	73856
第 7 層	MaxPooling2D	0
第 8 層	Dropout	0
第 9 層	Flatten	0
第 10 層	Dense	4196352
第 11 層	Dropout	0
第 12 層	Dense	4196352
第 13 層	Dropout	0
第 14 層	Dense	14343

利用 ImageDataGenerator 來做圖片的變化 / 處理，train model 的 optimizer 則用 adam，samples_per_epoch 則為 feature 數量的 2 倍。下圖為 epochs = 100 的訓練過程（選擇 100 是因為發現 100 後其實已經幾乎收斂，繼續訓練不會有太大幫助），使用上述方法訓練出來的 CNN Model 在 public leaderboard 其辨識率約略微 66.89 %。



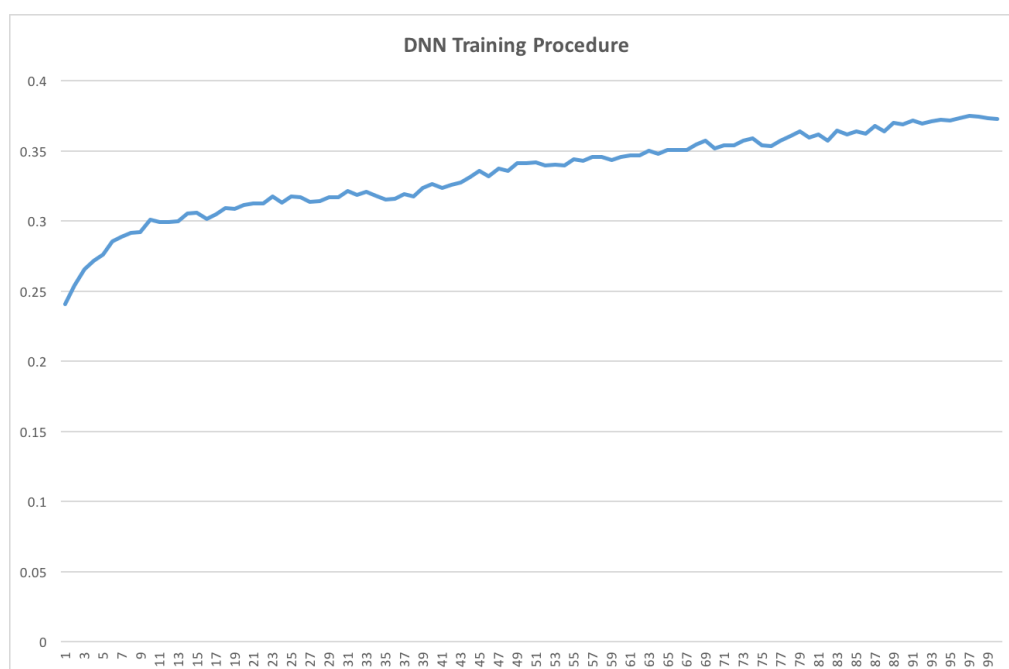
2. 承上題，請用與上述 CNN 接近的參數量，實做簡單的 DNN model，其模型架構、訓練過程和準確率為何？試與上題結果做比較，並說明你觀察到了什麼？ (1%)

答：

使用的 DNN 模型架構如下表：

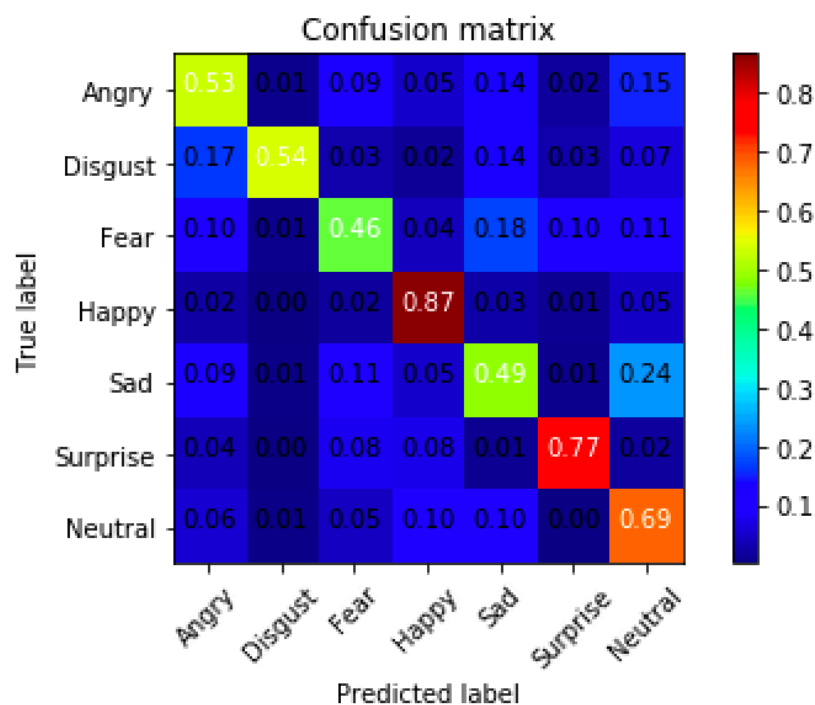
層次	Type	Param
第 1 層	Dense	590080
第 2 層	Dense	65792
第 3 層	Dense	65792
第 4 層	Dense	131584
第 5 層	Dense	262656
第 6 層	Dense	262656
第 7 層	Dense	525312
第 8 層	Dense	1049600
第 9 層	Dense	1049600
第 10 層	Dense	1049600
第 11 層	Dense	1049600
第 12 層	Dense	2099200
第 13 層	Dense	14343

利用上述 DNN 模型架構可訓練參數約為 8215815(CNN 約為 8536647)，其中 optimizer 使用 adam，samples_per_epoch 則為 feature 數量的 2 倍。下圖為 epochs = 100 的訓練過程：



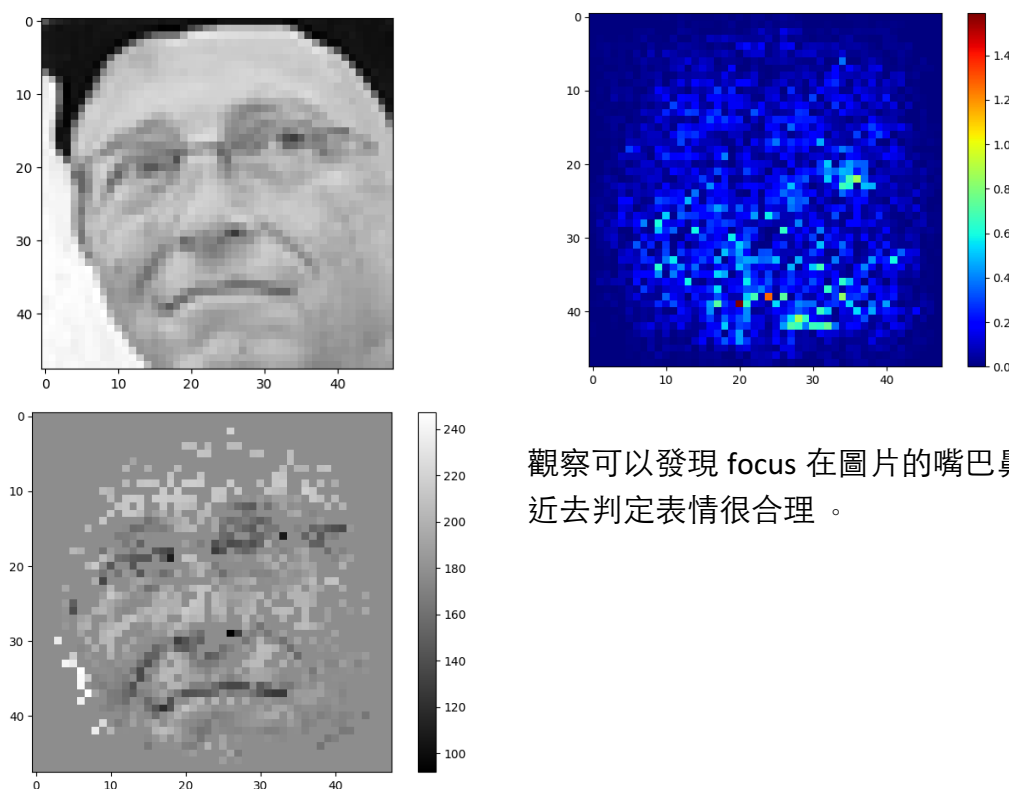
從訓練過程中可以發現雖然 DNN 與 CNN 參數量差不多的，但 DNN 結果一下子就收斂(上升不快)，會有這種差異主要還是如老師上課所說的 filter（讓 Machine 看到 pixel 相對位置關係）。

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



從上圖可以發現 Happy 最好辨認，我想應該是開心會有笑容，較容易捕捉到。而錯誤最嚴重的則是將 Sad 視為 Neutral，這部分我則認為我自己也很難從資料是人其實也很分辨 Neutral 與 Sad，所以機器會誤判也是很合理（容易誤判的 Case 幾乎是如此，這邊就省略不再多談）。

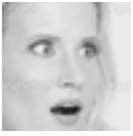
4. 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps 觀察模型在做 classification 時，是 focus 在圖片的哪些部份？



觀察可以發現 focus 在圖片的嘴巴鼻子附近，透過嘴巴鼻子附近去判定表情很合理。

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**。

(input image)



(layer output)

