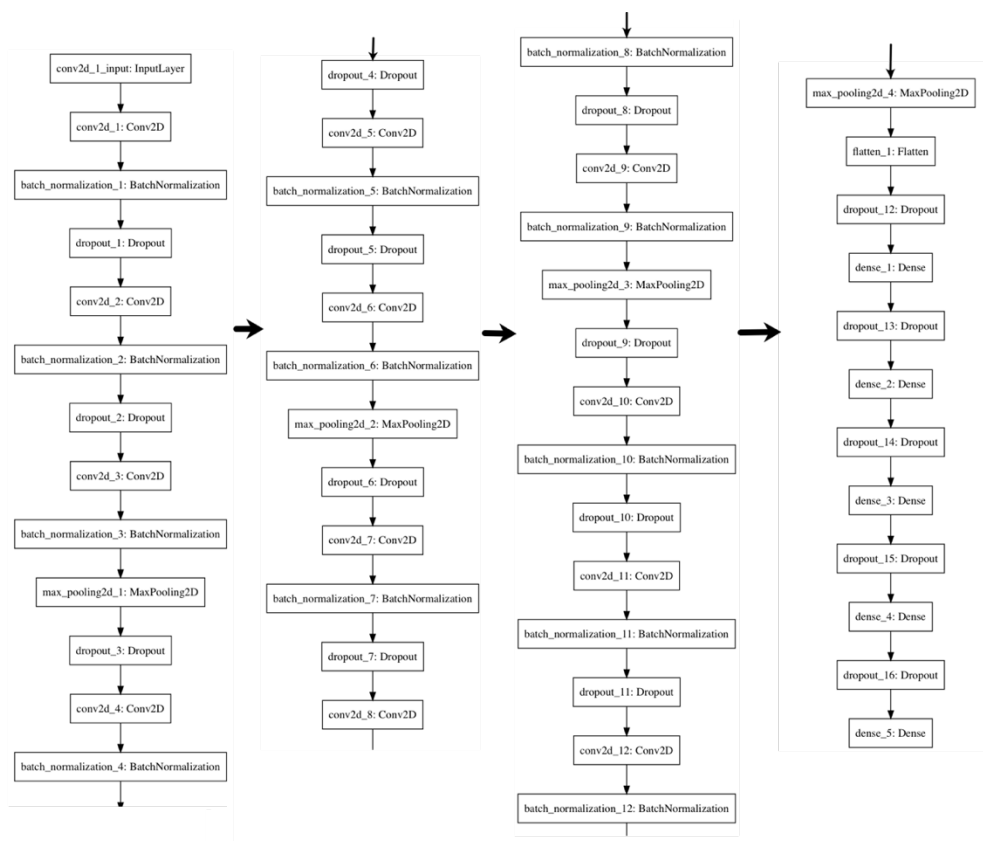


1. (1%) 請說明你實作的 CNN model，其模型架構、訓練過程和準確率為何？

(Collaborators:和紹志宇、呂承洋討論過如何改進，請教過修 ADL 的黃泓硯，)

答：

模型架構：12 層的 3x3 的 CNN 接 5 層 dense，activation 用 Relu。12 層 CNN 中上到下分別 3 層是 32 個 filter、3 層是 64 個 filter、3 層是 128 個 filter、3 層是 256 個 filter，最後的 5 層 dense 分別各是 256、128、64、32、7 個 node。每一層 CNN 都有加 batch_norm，每一 layer 都有加 dropout, dropout 從 0.05~0.3 慢慢增大。



Model.Summary 結果：

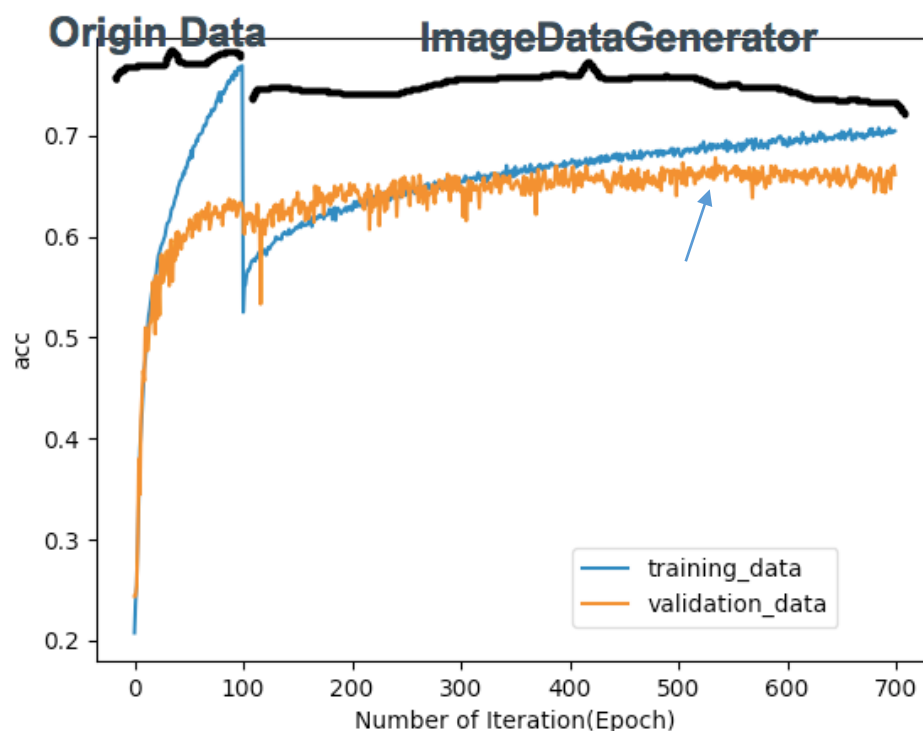
Total params: 2,267,143

Trainable params: 2,264,263

Non-trainable params: 2,880

訓練過程：將 training_data 以 9 : 1 的方式分出 10% 的資料作為 validation data，然後下去 Train 100 個 epoch，此時的 validation 準確率在 0.62，已過 simple baseline，然後再以 Keras 的 imageDataGenerator 的 flow 來生資料繼續 Train 600 個 epoch。根據最高的 validation 的 epoch 來上傳到 kaggle。

準確率：在第 535 的 epoch 中 (ImageDataGenerator 的第 435 個 epoch)
validation acc 到達最高為 0.678161，上傳到 Kaggle，得到 public acc: 0.67874



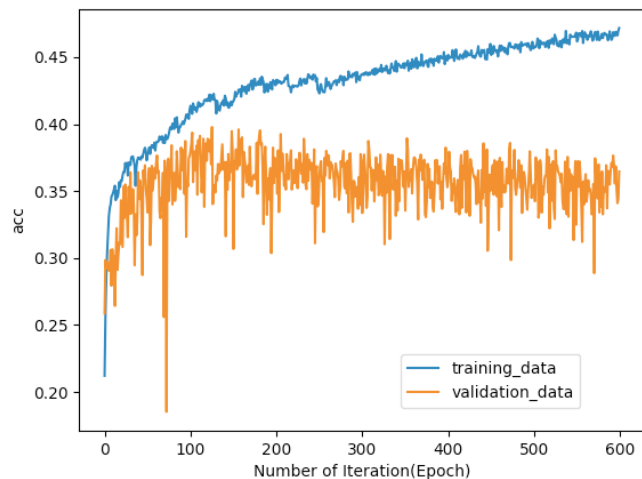
2. (1%) 承上題，請用與上述 CNN 接近的參數量，實做簡單的 DNN model。其模型架構、訓練過程和準確率為何？試與上題結果做比較，並說明你觀察到了什麼？

(Collaborators: 無)

答：

模型架構：按照第一題的 2,267,143 個 parameters 建制差不多數量級的 DNN。總共有 7 層 Dense，每層如下圖從 768、512、256 個 para 遞減到 7 個。

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 768)	1770240
batch_normalization_1 (Batch Normalization)	(None, 768)	3072
dropout_1 (Dropout)	(None, 768)	0
dense_2 (Dense)	(None, 512)	393728
batch_normalization_2 (Batch Normalization)	(None, 512)	2048
dropout_2 (Dropout)	(None, 512)	0
dense_3 (Dense)	(None, 256)	131328
batch_normalization_3 (Batch Normalization)	(None, 256)	1024
dropout_3 (Dropout)	(None, 256)	0
dense_4 (Dense)	(None, 128)	32896
dropout_4 (Dropout)	(None, 128)	0
dense_5 (Dense)	(None, 64)	8256
dropout_5 (Dropout)	(None, 64)	0
dense_6 (Dense)	(None, 32)	2080
dropout_6 (Dropout)	(None, 32)	0
dense_7 (Dense)	(None, 7)	231
Total params: 2,344,903		
Trainable params: 2,341,831		
Non-trainable params: 3,072		
Train on 25838 samples, validate on 2871 samples		



訓練過程： 同樣切出 10% 的資料作為 validation data，train 600 個 epoch。

準確率：training data 的 acc 到 0.46，validation acc 基本在 0.35~0.4 之間抖動。

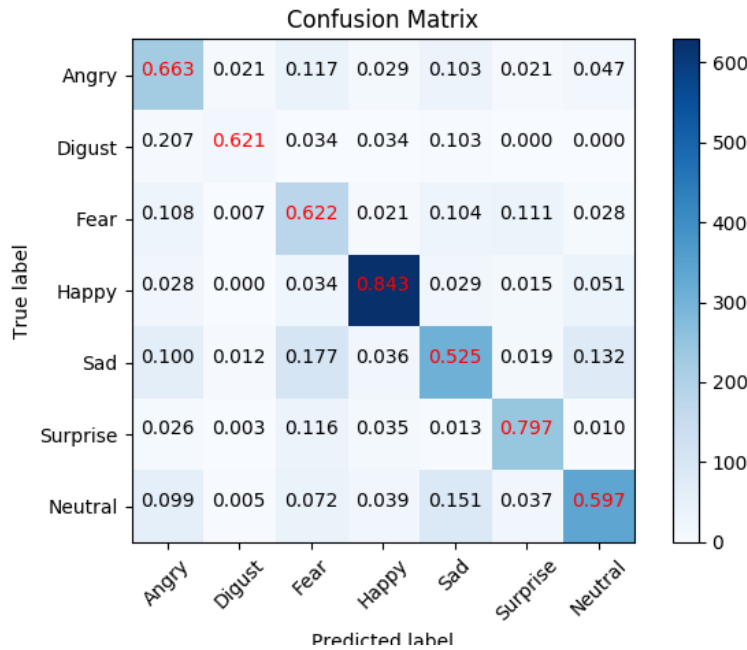
觀察：CNN 的變數比單純 DNN 的少，準確度卻比較好，且在 100 個 epoch 內 val acc 就可以到 0.6，我想這是由於 CNN 在相同的變數下可以做到更多層的 neural，而較深的架構比較能達到好的準確率。

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: 請教袁晟竣畫圖的方法)

答：

根據 Q1 所提的 model 及助教的範例作出 Confusion Matrix，發現較明確的 class 是 Happy，而容易用混的是 Sad 的圖被認作 Fear，其次是 Neutral 被認作 Sad。

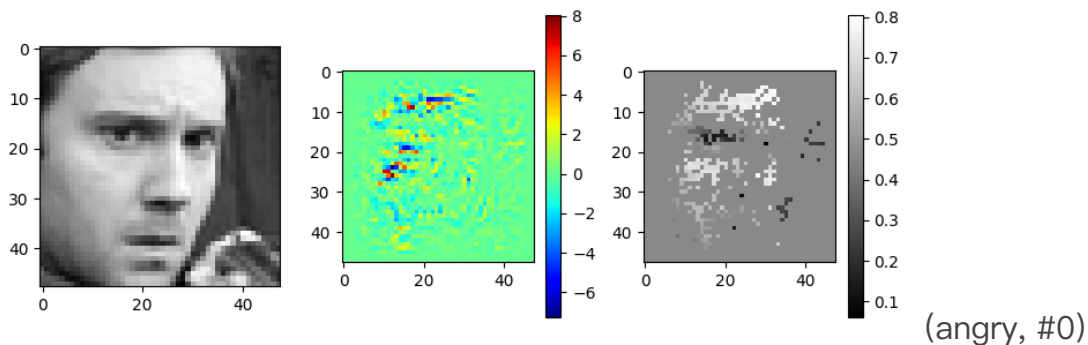


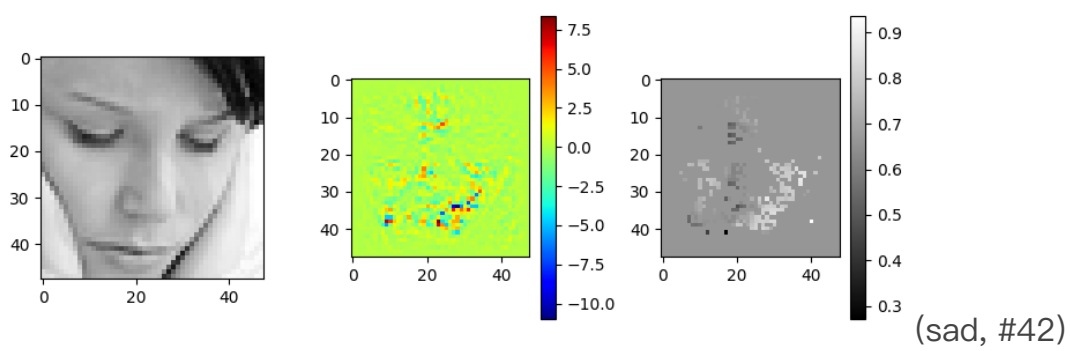
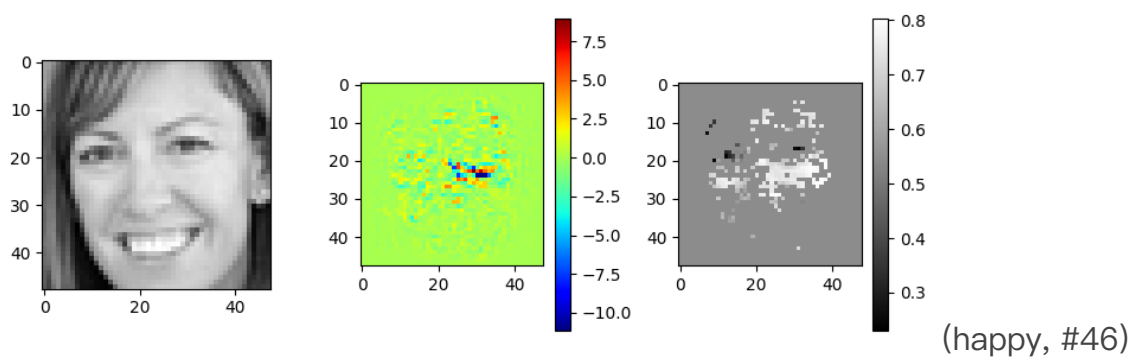
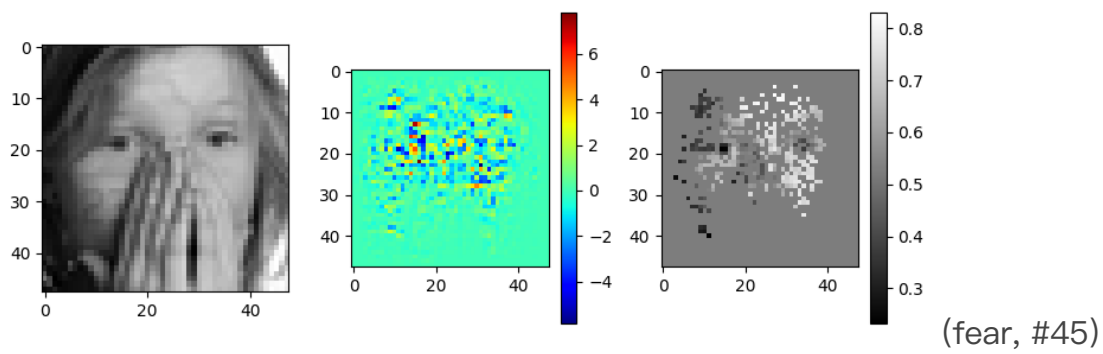
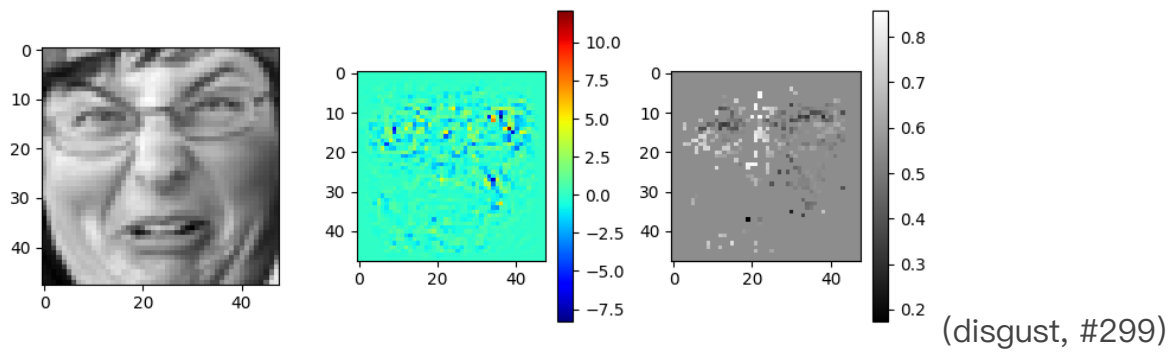
4. (1%) 從(1)(2)可以發現，使用 CNN 的確有些好處，試繪出其 saliency maps，觀察模型在做 classification 時，是 focus 在圖片的哪些部份？

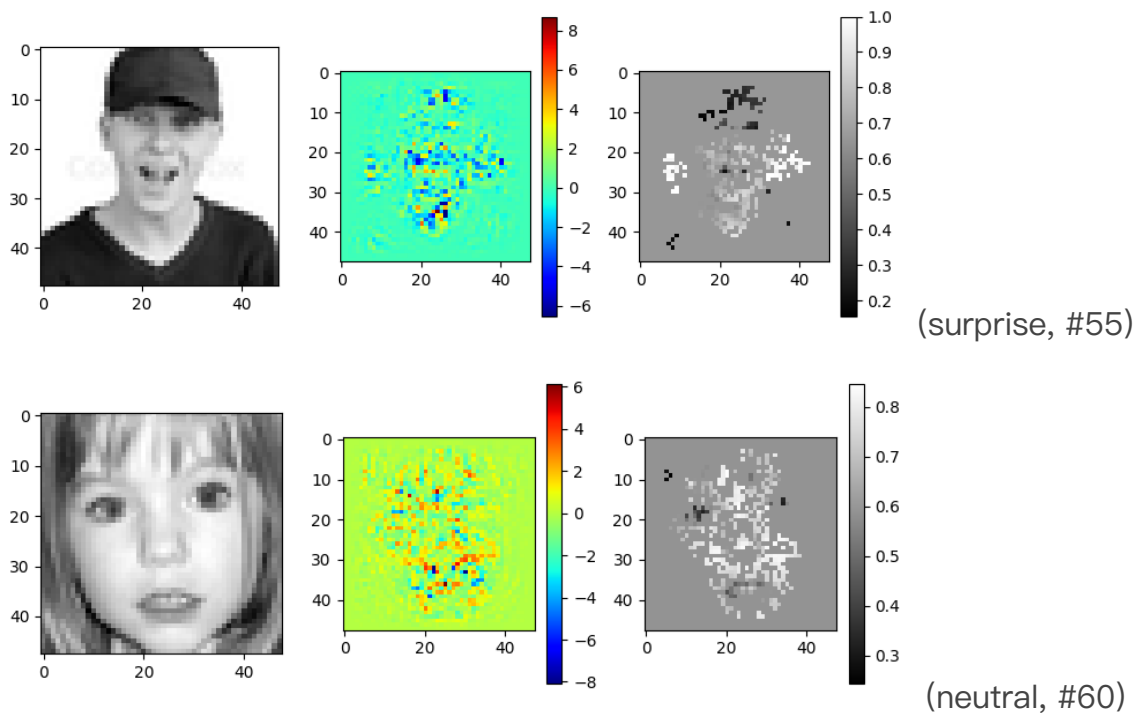
(Collaborators: 無)

答：

在 7 個表情中各選一張圖片作出 saliency map 如下，按照助教提供的 code 就是對圖片做 gradient，然後 normalize 之後畫出圖。觀察發現 model 主要 focus 在臉部的眼睛、鼻子和嘴巴的部分。在(fear #45)中發現，model 對於臉上的手也會避開。







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: 無)

答：分別用 random white noise 和 (disgust, #299)兩張圖對 model 的第一層 filter 作 filter map，在 white noise filter map 中發現有些 filter 反應比較多，而反應多的 filter 在 (disgust, #299)中也會顯示比較多的人臉，其著重在人的表情輪廓上面。

