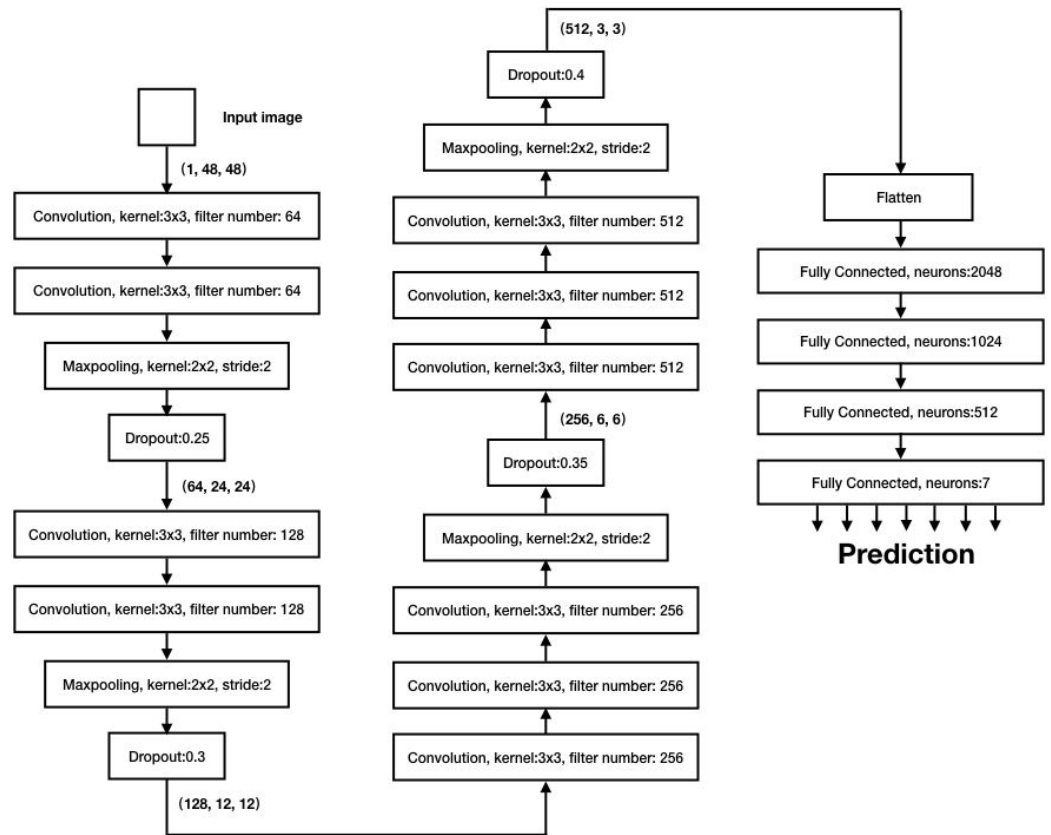
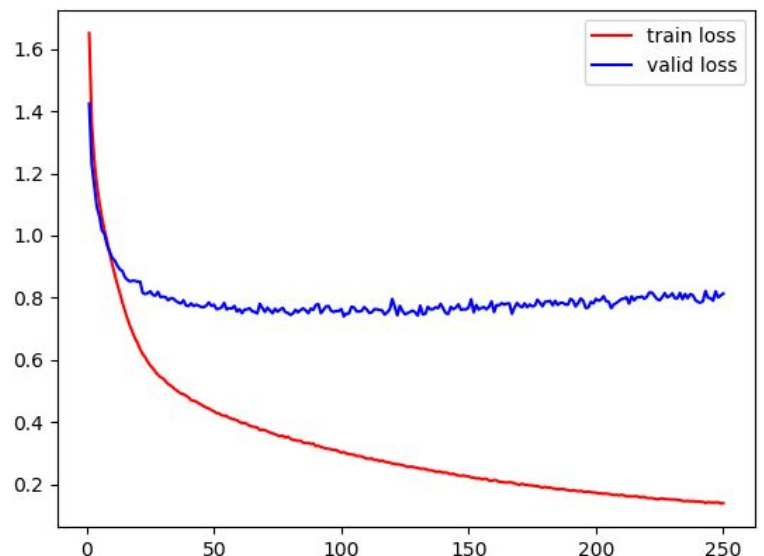
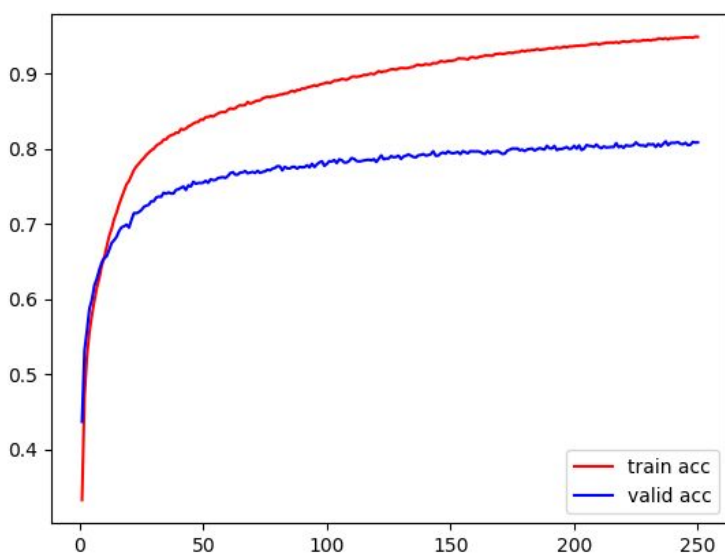


1. (2%) 請說明你實作的 CNN model，其模型架構、訓練參數和準確率為何？並請用與上述 CNN 接近的參數量，實做簡單的 DNN model，同時也說明其模型架構、訓練參數和準確率為何？並說明你觀察到了什麼？(Collaborators:)

答：



- 每一層 Convolution 都有做 zero padding，使得 input 和 output 長寬一樣
- 每一層 Convolution 後都有接 ReLU 和 Batch normalization
- 每一層 Fully connected 後都有接 ReLU
- Output Layer 後不再接 softmax，因為 Pytorch 在計算 Cross entropy 時輸入只需要 raw output 即可，它會自動幫忙作 softmax 再計算 loss
- training set:validation set = 8:2

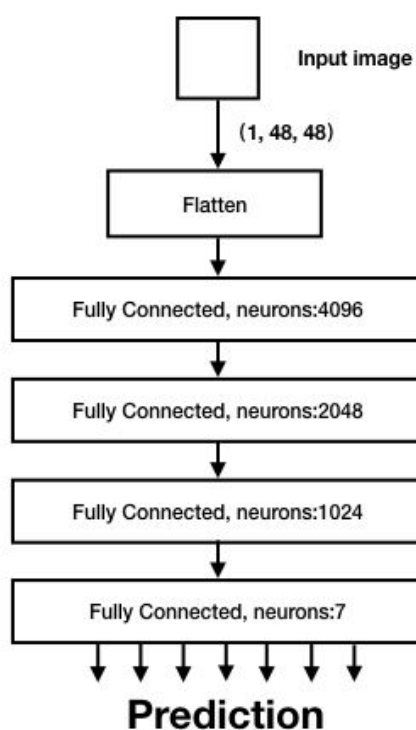


(Data是有先經過 normalization、data augmentation 再切成 8:2 的 training set/validation set)

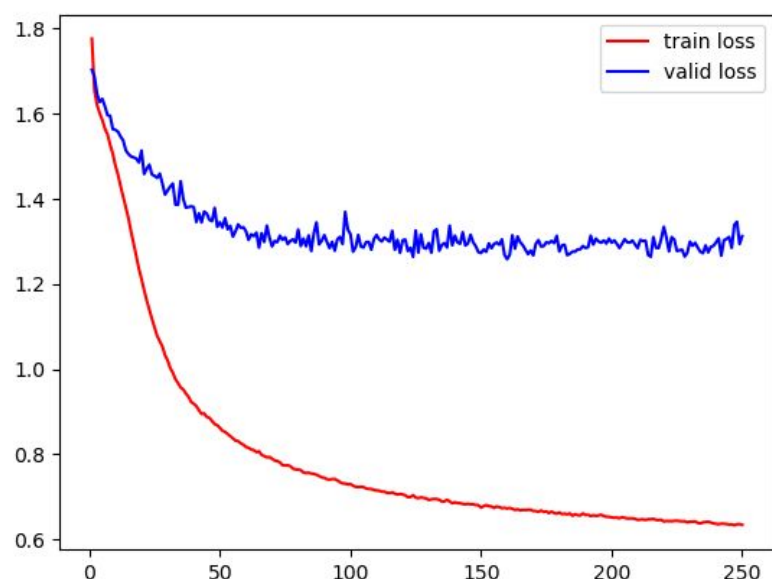
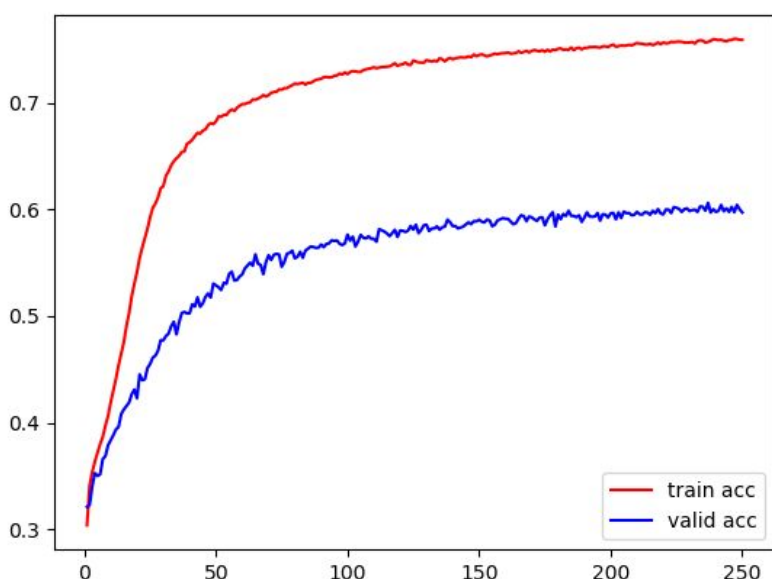
由圖片觀察到，training set 上的 accuracy 隨著 epoch 增加而一直變大，而且貌似沒有停止的跡象，只是變大的速度比較慢而已，很顯然已經 overfitting 了，可能的原因很多，例如我 network 接太深了而導致，而在 validation set 上的 accuracy 則很早就不再一直上升了，在 0.8 左右徘徊。在 loss 上面的表現也是類似，training set 一直下降沒停止的跡象，而 validation set 的 loss 則很快就收斂了。最後採用的 model 是在 validation set 上 accuracy 最高的，其數據表現如下

epoch	train acc	valid acc	Public
234	0.94706	0.80844	0.70353

整個 CNN model 所用到的參數數量為 19705287，建立一個 DNN model 使用參數數量為 19951623，架構圖如下



- 和 CNN 不一樣的是，DNN 每層 Fully connected 後有接 Batch normalization
- 和 CNN 一樣每層 Fully connected 後都有接 ReLU
- 這邊 output layer 也沒有接 softmax，因為 pytorch 的 cross entropy 會自動幫忙實作
- training set:validation set = 8:2



從圖片觀察到，在同樣的 data 下，model 架構的不同會大大的影響 performace，在 accuracy 上，training/validation set 在 0.75/0.6 左右就停止成長了，很快就 overfitting，而 loss 的表現，training/validation set 也都在相較於 CNN 的收斂值還要高的數值收斂，並且 validation set 的 loss 還具有較大的不穩定震盪現象。

2. (1%) 承上題，請分別畫出這兩個model的訓練過程 (i.e., loss/accuracy v.s. epoch)

(Collaborators:)

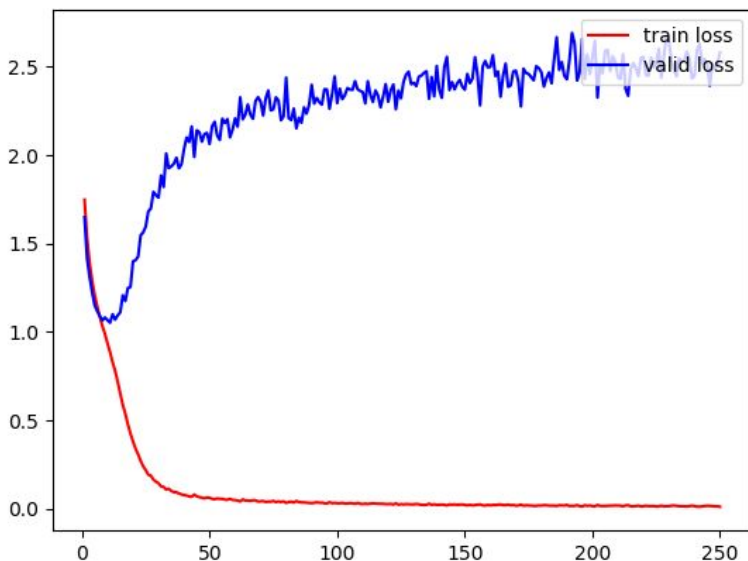
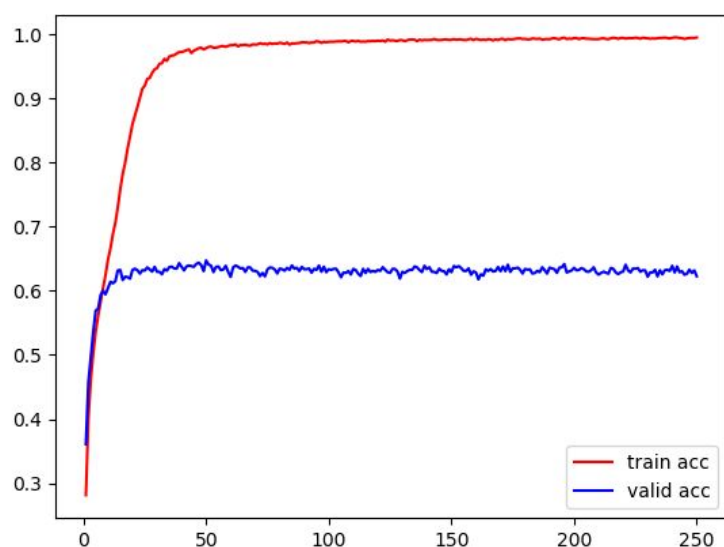
答：如上題圖所示

3. (1%) 請嘗試 data normalization, data augmentation, 說明實作方法並且說明實行前後對準確率有什麼樣的影響？

(Collaborators:)

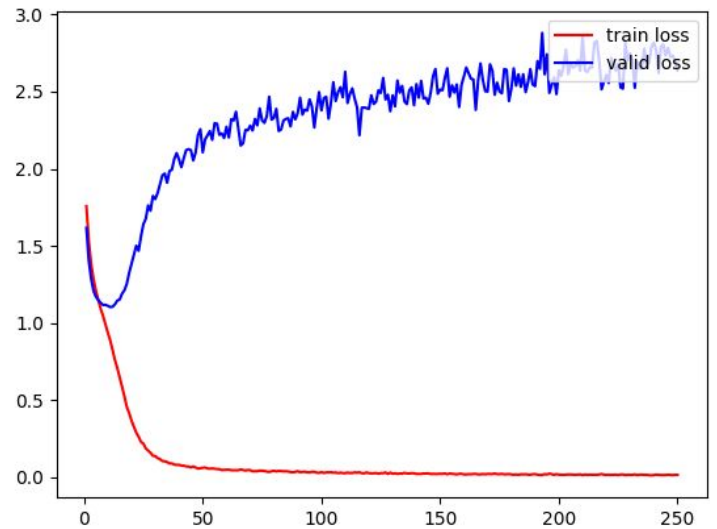
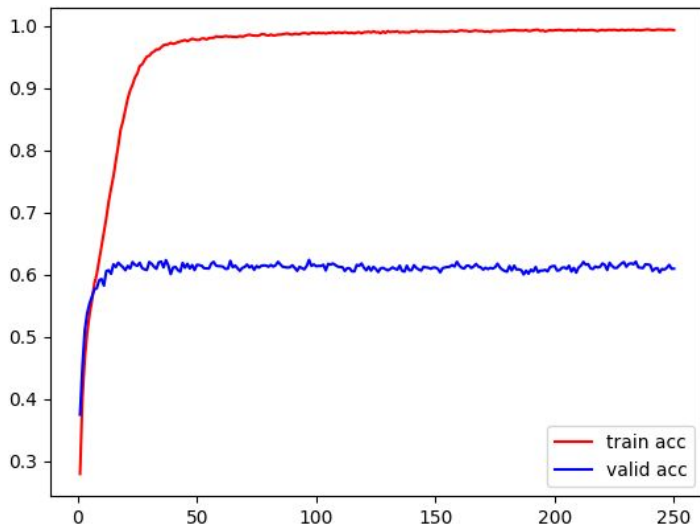
答：實驗分成三種類型的討論，如下

1) Raw data



2) 只採用 data normalization

a) 把 $[0, 255]$ 的 pixel 值除以 255 normalize 到 $[0, 1]$



3) 採用 data normalization 也採用 data augmentation 的模型

a) Data normalization

i) 把 $[0, 255]$ 的 pixel 值除以 255 normalize 到 $[0, 1]$

b) Data augmentation :

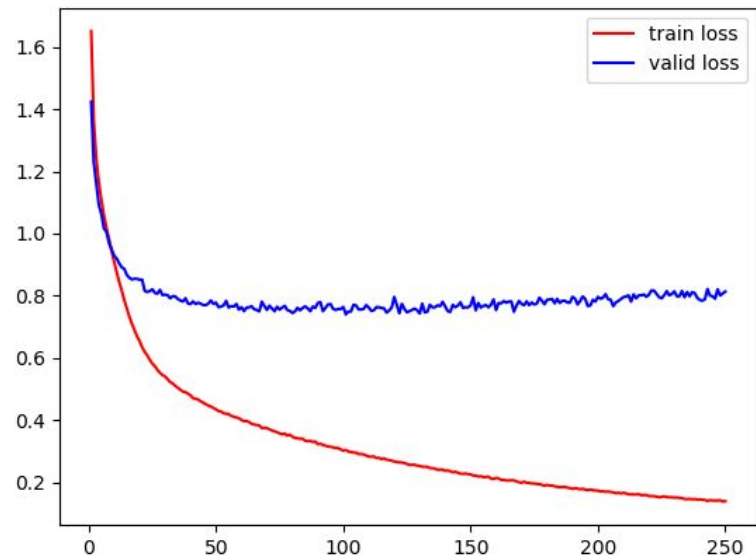
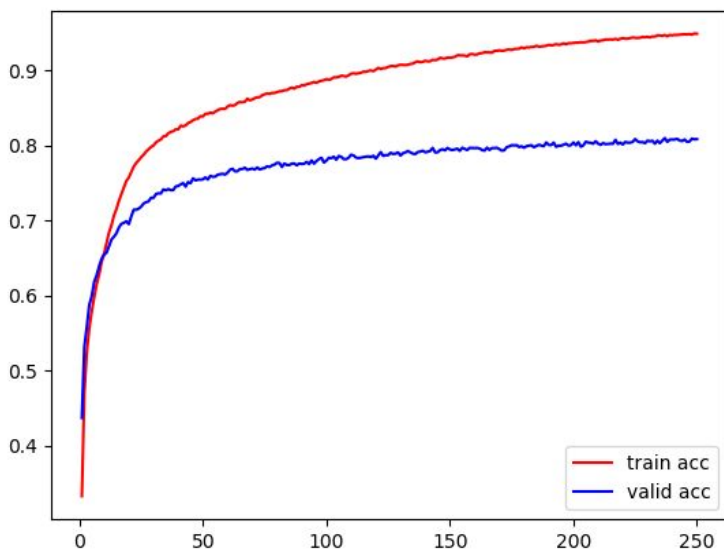
i) Random crop 成 (32,32) 大小的圖片，再 Resize 成 (48,48)

ii) 把照片水平翻轉

iii) 先 Random crop 後水平翻轉，再 Resize 成 (48,48)

iv) Random rotation 30度

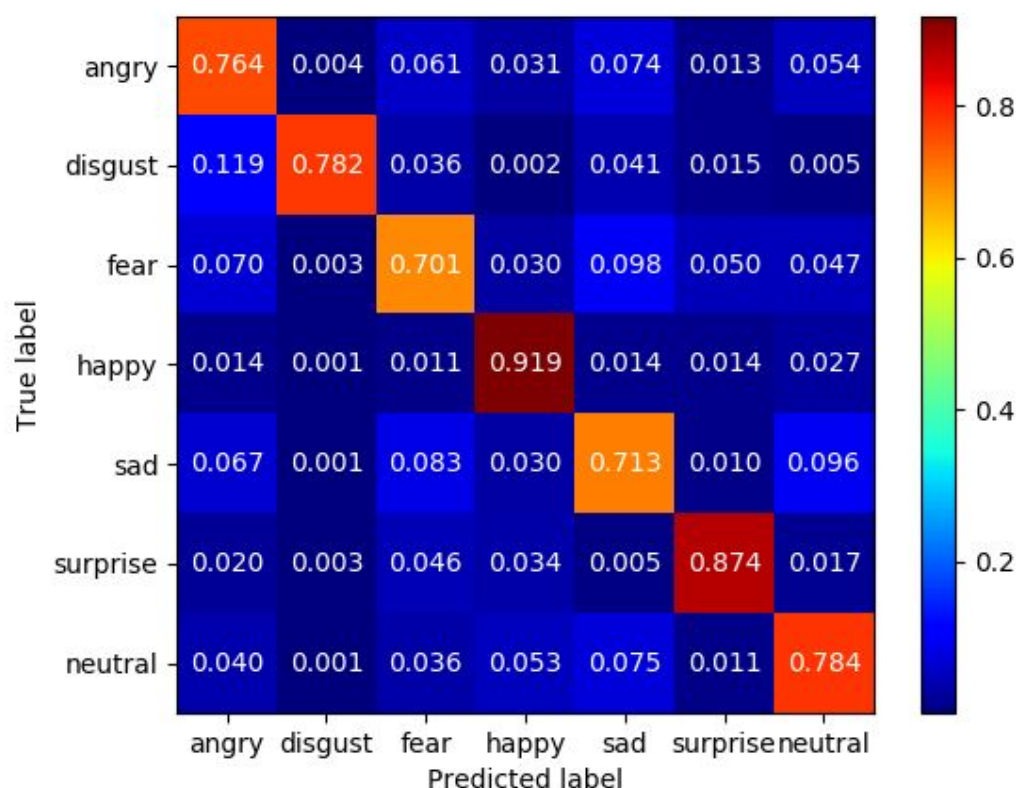
所以在實作 data augmentation 後，本來的 training data 就會從 28709 筆變成 $28709 \times 5 = 143545$ 筆。



在不實作 data augmentation 的狀況下，不管有沒有實作 data normalization 對實際 performance 沒什麼影響，而且兩個 case 的 acc/loss 圖可以說是幾乎快要一樣了，並沒有像我當初所想的「使用沒有 normalize data 的 model 應該連 training acc 都會很低」那樣，原因可能是因為 model 夠複雜，所以連沒 normalize 過的 data 都處理得了，但因為沒有 data augmentation 的關係，overfitting 太過嚴重導致在 validation set 上表現太糟糕，而在加了 data augmentation 後整體表現有大幅的上升，也證明了對 raw data 做一些擾動可以讓整個 model 更加的強健。

4. (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：



由觀察得到，在 validation set 上，「happy」、「surprise」的準確率是相對較高的，可能是因為這兩者的嘴形動作相對其他表情來看比較豐富，也比較好分辨，所以準確率會比較高。

在此matirx上的數據顯示，「sad」、「fear」之間有比較大的機率會分類錯誤，可能的原因是因為CNN在取local feature時，看到sad和fear的嘴唇都是下凹的，所以很有可能會在分類時誤判；另外一組有比較大機會分類錯誤的是「sad」和「neutral」，可能是有某些照片的sad其實是面無表情的而造成模型誤判。