

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？

答：

我的 CNN model 有六層 convolution layer，每一層都是 3\*3 的 kernal size，其中在第 1、2、4、6 層有做 2\*2 的 max-pooling。每一層的 Activation function 皆是 LeakyRelu(0.2)，並且在 activation 完後做 batch normalize，之後才在特定層數做 pooling。每一個 layer 的 channel 分別是 64,128,256,512,512,512。在 convolution layer 後則是有三層 fully-connected layer，參數數量分別是 512\*3\*3 -> 1024，1024 -> 512，512 -> 7，其中前兩層有做 0.5 的 dropout。Data 的部分則有做 augmentation，會隨機旋轉正負 30 度，以及隨機縮放 0.8~1.2 倍，還有隨機水平翻轉。

比較特別的是我發現我的 model 如果在 convolution layer 有做 dropout 的話，performance 會變差。而 kernal size 為 3 又比 5 要來的好。

我最好的結果是由這個 model，在 epoch 300~600 次當中，選三個比較好的結果進行投票。Private score 是 0.67539，Public score 是 0.68737。為了方便與 DNN 比較，以下的討論當中，CNN 的 model 是我投票的 3 個 model 當中 performance 最好的一個，這個 model 也有過 strong baseline。

我的 CNN model 大約有 1150 萬個參數，所以我也製造了一個有差不多參數量的 DNN，共有六層，每一層的參數數量分別是 2304 -> 2000，2000 -> 1600，1600 -> 1200，1200 -> 1000，1000 -> 600，600 -> 7。

兩個 model 同時 epoch 200 次的結果為：

	Public score	Private score
CNN	0.66007	0.65143
DNN	0.32738	0.32153

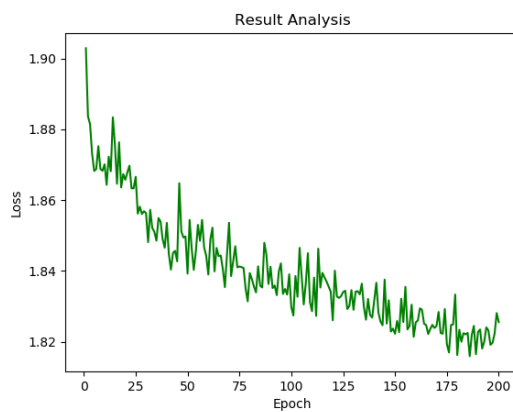
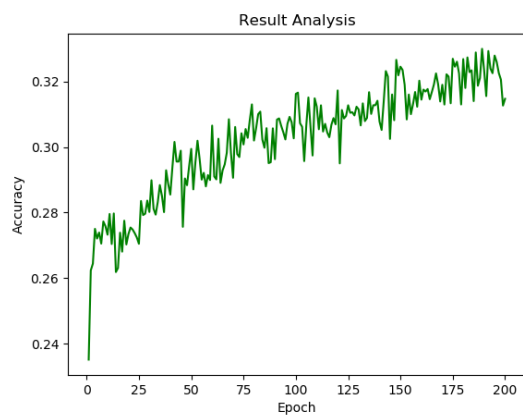
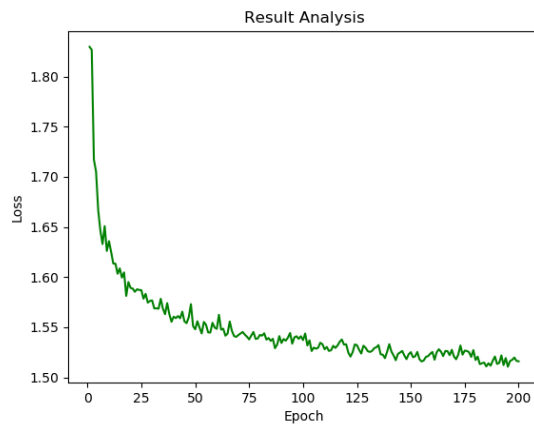
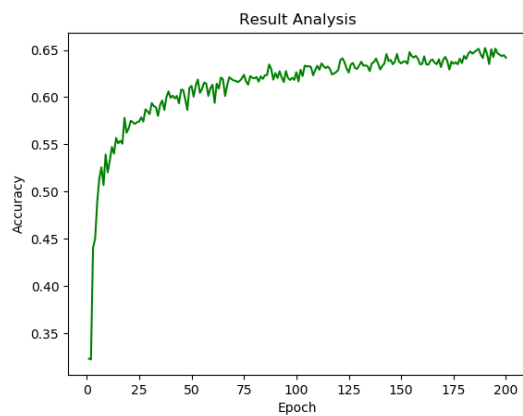
可以發現同樣的參數，CNN 比起 DNN 有著更好的效果。

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)

答：

上面兩個圖為 CNN 的 accuracy/loss 對 epoch 作圖，下面兩張則是 DNN 的圖。

兩張圖皆是切十分之一的 validation set 出來作圖。



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: )

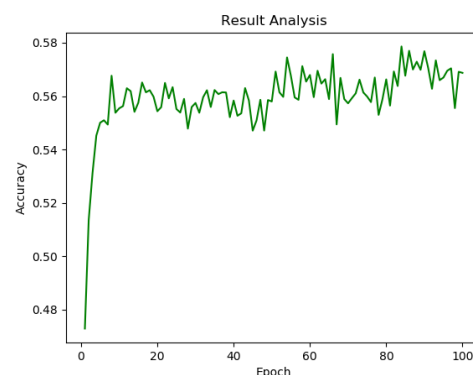
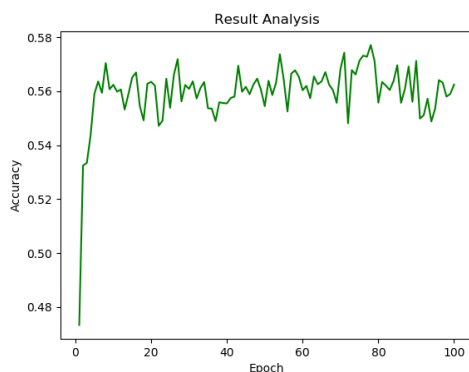
答：

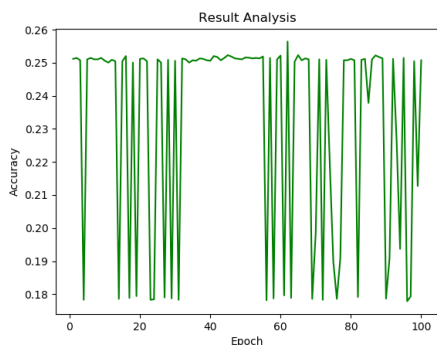
基於我在第一小題的 model，我的 data normalization 是將每個 pixel 都先除以 255，再丟進我的 model。至於 Data augmentation，則是如第一小題所述。

	Public score	Private score
nothing	0.58094	0.57536
data normalization	0.58289	0.57286
data normalization + data augmentation	0.25020	0.24407

可以發現同時做 data normalization 跟 data augmentation 會導致 accuracy 下降許多，有可能是因為 normalization 會讓每個 pixel 的差異變小，如果再做 augmentation 會讓 model 產生混淆，無法抓取特徵。

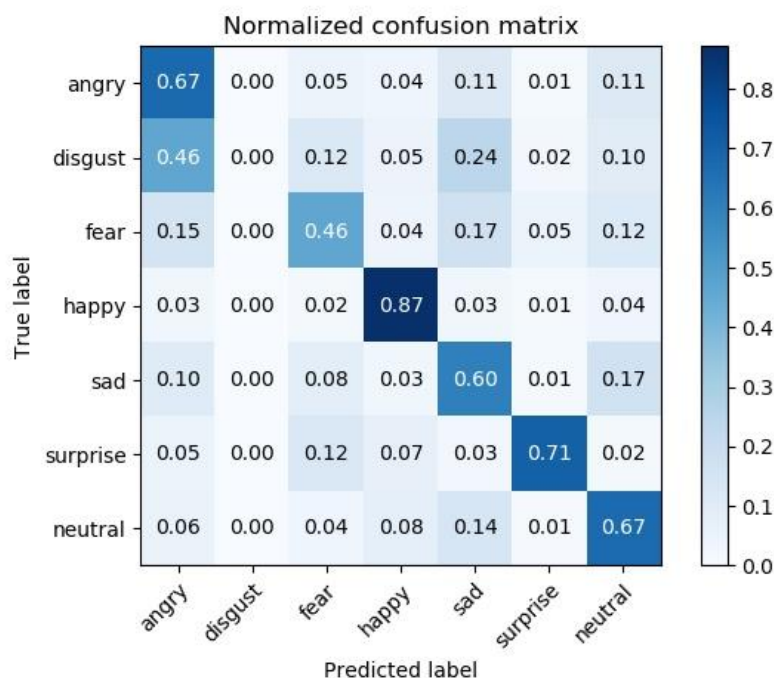
至於在沒有 augmentation 的情況下，有沒有 normalize 的效果是差不多的。





左上圖為兩個都沒有做的 accuracy 對 epoch 的圖，右上則是只有 normalization，左下則是兩個都有做。其中左下的圖一直在 18~25% 變動。

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：

由圖中可以看出，我的 model 比較容易誤認的 class 有：將 disgust 辨識為 angry 和 sad；fear 也有較高的機率被誤認為 angry、sad、neutral；還有 sad 跟 neutral 容易互相混淆。

此外，我的 model 識別 happy 的準確率非常高，較難識別的 class 則是 fear 跟 disgust，其中 disgust 是完全不會被識別出來的。