學號：R07943107 系級：電子所碩一  姓名：徐晨皓

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: None)**

答：

* CNN model

1. 模型架構：

本次作業使用keras實作，model分為兩部分，CNN與DNN。圖1為本次作業CNN模型架構。首先，由**五層convolution layers**與**三層max pooling layers**交錯組成CNN的部分，每層convolution layer都有使用**zero padding**的技術，讓圖片不會損失原有大小。接著再以**三層fully connected layers**組成DNN部分，最後再接上output layer。除了output layer的activation function為**softmax**，convolution layer與dense layer都使用**relu**作為activation function。此模型有使用**dropout**來避免overfitting。

(有使用data augmentation)

1. 訓練參數：
2. Dropout rate: 0.25
3. Optimizer: Adam (default)
4. Epochs: 100
5. Batch size: 100
6. **Number of parameters: 7,255,815**
7. 準確率：

Private score: 0.70103

Public score: 0.71217

* DNN model

1. 模型架構：

圖2為DNN模型架構。使用了共**九層的dense layers** (含output layer)。除了output layer的activation function為**softmax**，convolution layer與dense layer都使用**relu**作為activation function。此模型有使用**dropout**。

1. 訓練參數：
2. Dropout rate: 0.25
3. Optimizer: Adam (default)
4. Epochs: 100
5. Batch size: 100
6. **Number of parameters: 7,268,615**
7. 準確率：

Private score: 0.38032

Public score: 0.36639

* 觀察結果

雖然CNN model與DNN model的參數量接近，但由public score與private score來看，CNN model的預測結果比DNN model的預測結果相當多。

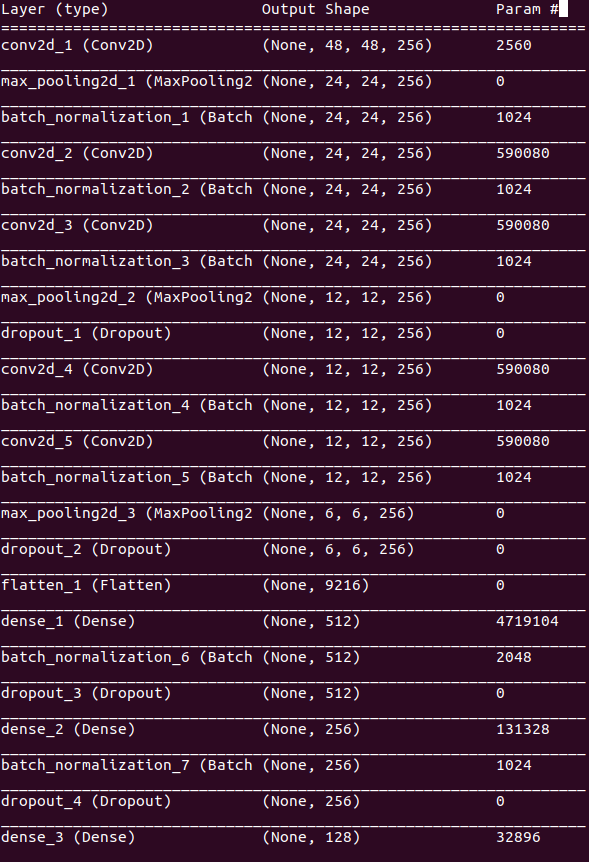
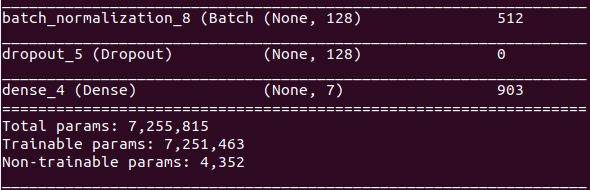


圖1. CNN model

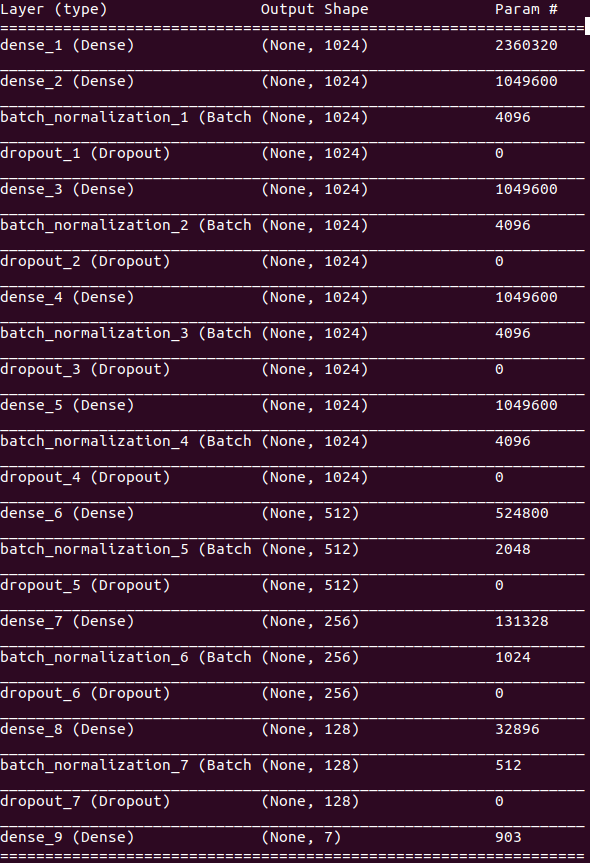
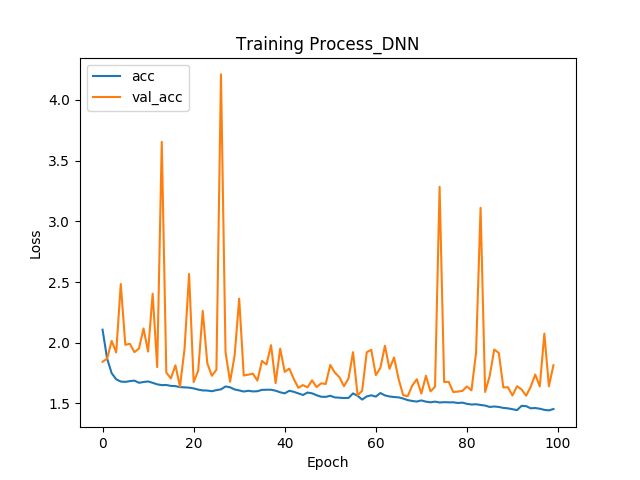
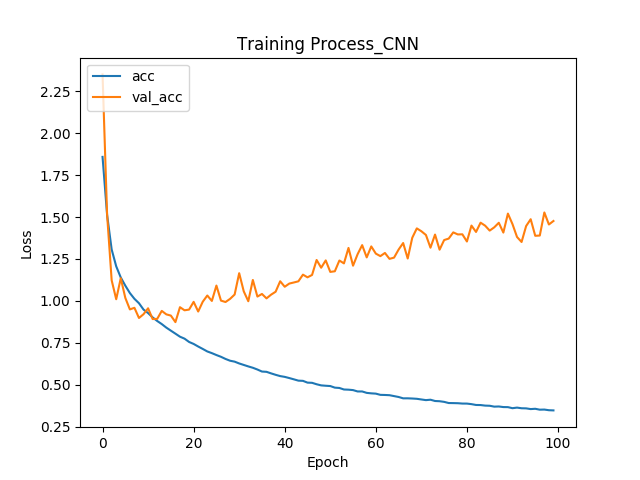
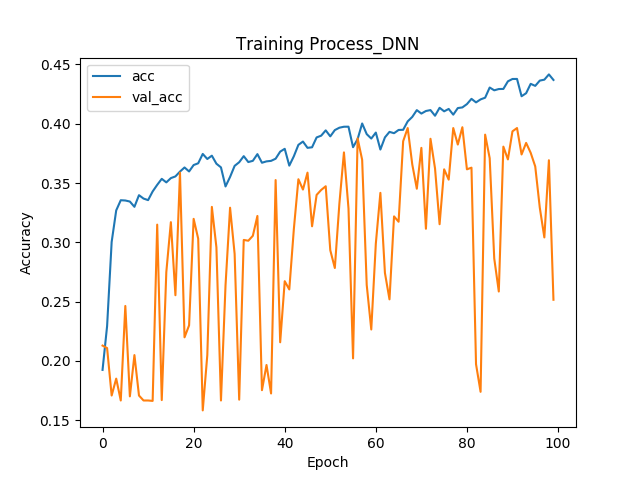
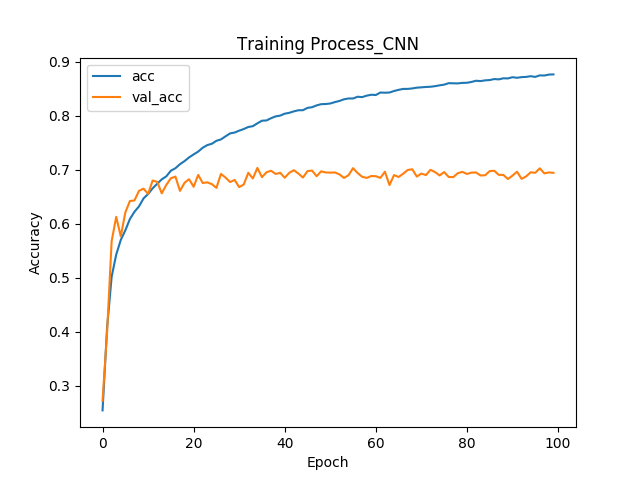


圖2. DNN model

1. **(1%) 承上題，請分別畫出這兩個models的訓練過程 (i.e., loss/accuracy v.s. epoch)  
   (Collaborators: None )**

答：



(取10%的training data當作validation set)

由上四圖可以觀察到在訓練過程中:

1. CNN model的training accuracy可以達到約0.90，但DNN model的training accuracy只能達到約0.45。
2. CNN model的validation accuracy穩定在約0.70，而DNN model的validation accuracy卻在0.16和0.40之間劇烈震盪。
3. CNN model的training loss下降至約0.30，但DNN model的training loss只能達到約1.50。
4. CNN model的validation loss雖然逐漸上升但沒有很大的震盪，可能是為了讓model更fit training data造成validation loss增加。而DNN model的validation loss卻在1.6和4.0之間劇烈震盪。
5. **(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: None)**

答：

* 實作方法

1. Data normalization：由於圖片的pixel是0~255的灰階，因此直接將每個pixel的值除以255，即可達成data normalization。
2. Data augmentation：此作業使用的是keras的ImageDataGenerator，來增加training data。使用的參數為

width\_shift\_range = 0.2,

height\_shift\_range=0.2,

rotation\_flip = 12,

zoom\_range=0.5,

fill\_mode=’nearest’。

* 實驗結果

1. 兩者都不使用：

Private score: 0.59765

Public score: 0.60573

1. 只使用data normalization：

Private score: 0.62412

Public score: 0.63220

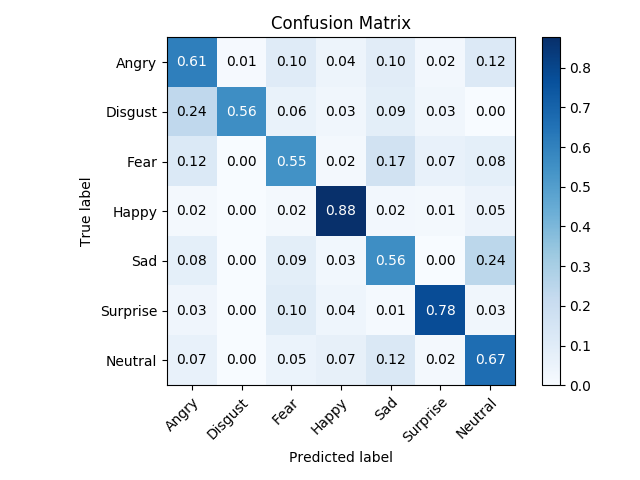
1. 同時做data normalization與data augmentation：

Private score: 0.69908

Public score: 0.69657

* 實驗結果分析

由實驗結果來看，使用data normalization與data augmentation都能使準確率上升。其中以data augmentation的效果最為顯著。

1. **(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: None)**

答：

根據confusion matrix我們可以觀察到以下兩點：

1. Disgust的圖片容易被辨識成angry。由於disgust與angry的圖片都有皺眉與嘴角向下的特徵，容易造成model誤判。
2. Sad的圖片容易被辨識成neutral。因為有些sad圖片的臉部肌肉沒有太大的改變，因此model會將sad辨認成neutral。但neutral卻相對不易被誤判成sad。