學號：B08902075　　系級：資工一　　姓名：林耘平

1. 請說明你實作的 CNN 模型(best model)，其模型架構、訓練參數量和準確率為何？(1%)

我採用的模型架構9層Convolution Layer、其中穿插5層Max Pooling，最後接上兩層Fully Connected Feed-Forward Hidden Layer，這兩層都套用機率為0.5的Dropout，optimizer是adam，learning rate是0.001。所有我使用的Convolution Layer參數都一致，kernel_size=3，stride=1，padding=1。

Layer Type	Input Size	Output Size	Number of Parameters
Convolution	3*128*128	64*128*128	64*3*3*3
Convolution	64*128*128	128*128*128	128*64*3*3
Max Pooling	128*128*128	128*64*64	0
Convolution	128*64*64	128*64*64	128*128*3*3
Convolution	128*64*64	256*64*64	256*128*3*3
Convolution	256*64*64	512*64*64	512*256*3*3
Max Pooling	512*64*64	512*32*32	0
Convolution	512*32*32	512*32*32	512*512*3*3
Convolution	512*32*32	512*32*32	512*512*3*3
Max Pooling	512*32*32	512*16*16	0
Convolution	512*16*16	512*16*16	512*512*3*3
Max Pooling	512*16*16	512*8*8	0
Convolution	512*8*8	512*8*8	512*512*3*3
Max Pooling	512*8*8	512*4*4	0
Fully-Connected	512*4*4	1024	(512*4*4+1)*1024
Fully-Connected	1024	512	(1024+1)*512
Fully-Connected	512	11	(512+1)*11

CNN的參數量約為1.1*10e7，全連接層的參數量約為總參數量大約是8.9*10e6，總參數量約為2*10e7。

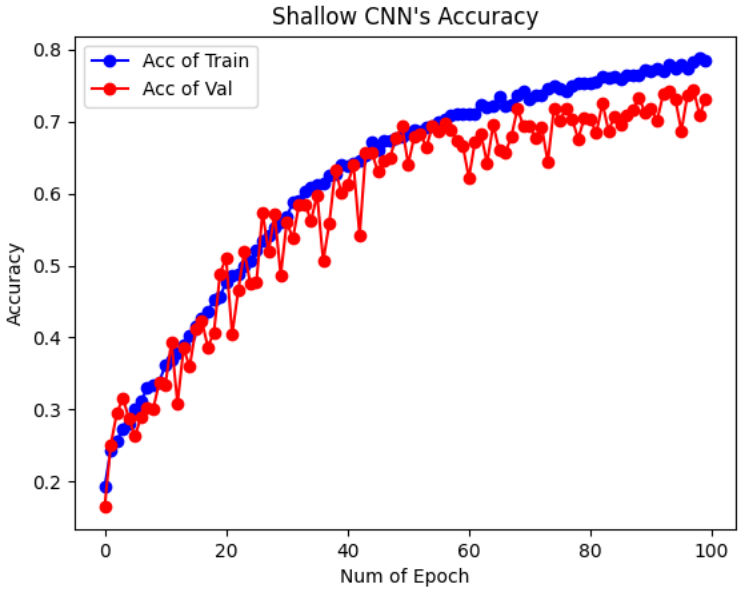
由於train的時候是將training set和validation set一起訓練，所以並沒有兩者準確率的比較，但若以kaagle的public set當作validation set，我在105個epoch，本地訓練資料77%的時候丟過一次kaggle，得到78%的正確率，最後在180個epoch，本地訓練資料83%的時候，在kaggle也得到83%的準確率。

2. 請實作與第一題接近的參數量，但 CNN 深度（CNN 層數）減半的模型，並說明其模型架構、訓練參數量和準確率為何？(1%)

以下是我設計的CNN架構，參數量約為9.5*10e6，和上題的CNN部份接近，至於全連接層和其他各種參數都和上題一致。

Layer Type	Input Size	Output Size	Number of Parameters
Convolution	3*128*128	512*128*128	512*3*3*3
Max Pooling	512*128*128	512*64*64	0
Convolution	512*64*64	512*64*64	512*512*3*3
Max Pooling	512*64*64	512*32*32	0
Convolution	512*32*32	512*32*32	512*512*3*3
Max Pooling	512*32*32	512*16*16	0
Convolution	512*16*16	512*16*16	512*512*3*3
Max Pooling	512*16*16	512*8*8	0
Convolution	512*8*8	512*8*8	512*512*3*3
Max Pooling	512*8*8	512*4*4	0

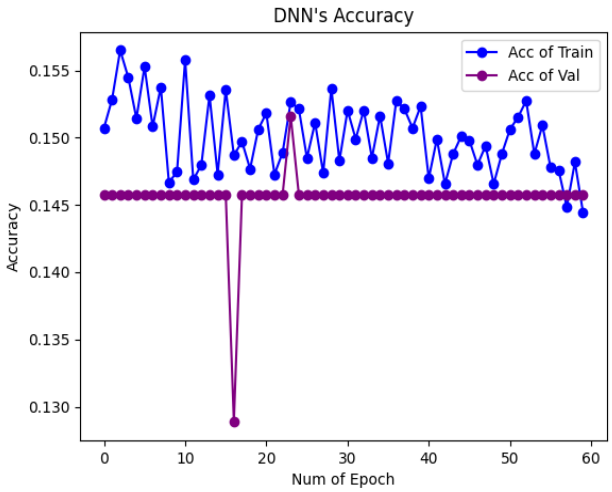
下圖是實驗數據，礙於計算資源，並沒有讓這個model訓練到收斂，不過可以看出大概在60個epoch之後，validation set跟training set的準確率就開始出現落差，然而第一題的best_model，兩組資料的準確率卻是接近的。



3. 請實作與第一題接近的參數量，簡單的 DNN 模型，同時也說明其模型架構、訓練參數和準確率為何？(1%)

以下是我實驗用的DNN模型，總參數量大約是2*10e7，除了model本身，其餘參數、訓練方式都和best model相同。接下來則是實驗得出的準確率數據，由此可知純DNN在這種影像辨識的問題並不在行。

Input Size	Output Size	Num of parameters
3*128*128	400	19660800
400	300	120000
300	11	3300



4. 請說明由 1 ~ 3 題的實驗中你觀察到了什麼？(1%)

首先，影像辨識問題的CNN深度頗重要，可以讓model整理出更關鍵的image features，然而全連接層的參數量或深度較不重要，除了上題將每層DNN的neuron數量變多之外，我也試過加深層數，不過效果都沒有不慎理想。

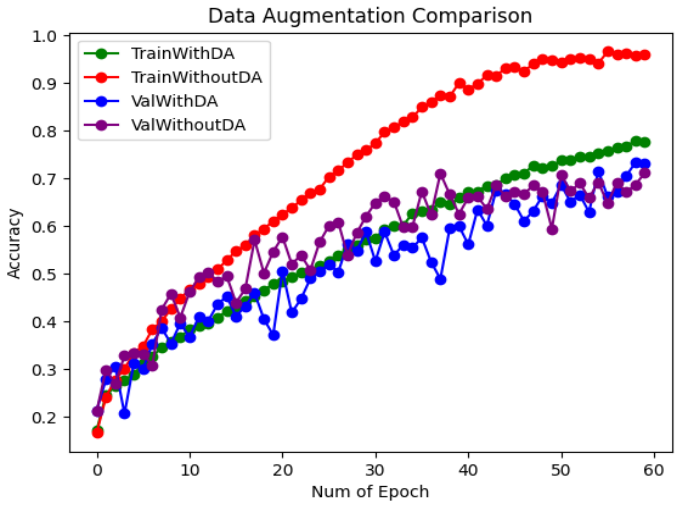
另外，在optimizer和各種參數固定的情況下，較淺的CNN訓練時間會比較短，loss下降的速度比較快，然而我的best mode（9層），train到160個epoch，準確率才正要突破8成，我推測是因為function set比較大和參數比較多，才導致這種緩慢的training速度，不過另一方面，在多參數和大function set的加持之下，比較容易找到一個好的function，也比較不容易overfit，。

5. 請嘗試 data normalization 及 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？(1%)

由於transforms.ToTensor()已經有標準化的功能了，因此再多使用normalization()的幫助很有限，至於我和大部分人的作法都是把被ToTensor映射到[0,1]之間的資料重新轉換到用平均為零，標準為一的數值範圍。

原本助教提供的範例程式碼中，只採用了RandomHorizontalFlip()和RandomRotation(15)兩個函式來增強訓練資料。我覺得食物不論怎麼翻轉，或調整顏色、對比、亮度都不會改變一般人對於食物類型的認知，因此我多增加了RandomVerticalFlip()和ColorJitter()，也把RandomRotation()的角度改為至多45度，來增加訓練資料的多樣性。

將best_model所用的data augmentation取消，和best_model比較準確率，便得到以下圖表，礙於計算資源的匱乏，便只訓練60個epoch。



我們可以得知在沒有施行Data Augmentation的時候，validation set的準確率會低於training set非常多，無論跑多少個epoch，準確率的差距都十分巨大。而有實施Data Augmentation的model，兩組資料間的準確率相對比較接近，較沒有overfitting的問題，我的best model甚至training set、validation set和kaggle public的準確率都差不多。

6. 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%)

以下是只用training set訓練，用validation set求出的Confusion Matrix，而不是由best model所求出，以避免model已經先看過validation set。

由圖可知，Class 1容易被辨認成Class 2，Class 3容易被辨認成Class 0，Class 6容易被辨認成Class 7，而這三個Class恰好也是準確率最低的三個。