

HW3 Report

學號：b07901039

系級：電機二

姓名：劉知穎

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%)

(1) 模型架構

總共使用 5 層 convolution layer -> batch normalization -> ReLU -> Max Pooling。每層 convolution layer 都使用 3*3 的 filters、stride = 1、padding = 1，第一層 input channel 為 3、output channels 為 64，接下來二、三、四、五層 output channels 分別為 128、256、512 和 512。

Max Pooling layer 都使用 kernel size = 2、stride = 2、padding = 0。

最後的 feedforward network 的結構為 linear->ReLU->linear->ReLU->linear，第一層 linear input 維度為 512*4*4、output 維度為 1024，二、三層 linear 的 output 維度分別為 512 和 11。

Data augmentation 則在每次取用一個 batch 的資料時隨機對圖像水平翻轉、隨機旋轉 0 至 15 度、隨機調整影像的亮度、隨機將影像剪裁(crop)。

使用 Adam，learning rate 設為 0.001，epoch 設為 120。

(2) 參數量：大約 12.8M。

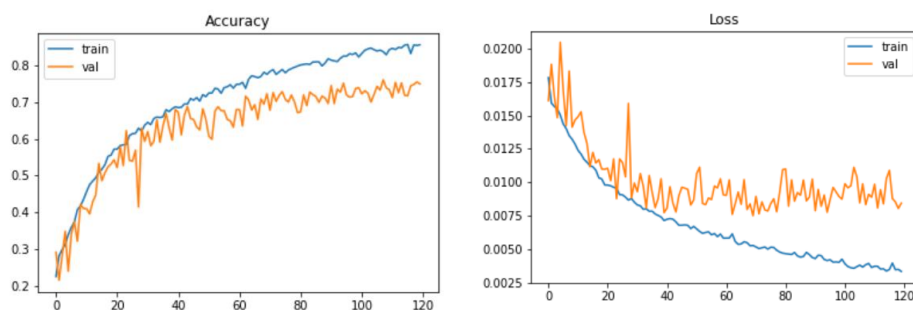
(3) 準確率

training accuracy : 0.857186、loss : 0.003333

validation accuracy : 0.750437、loss : 0.008438

將 training set 和 validation set 合併後的 training accuracy : 0.890117、
loss : 0.002628

kaggle 上 public accuracy : 0.82785



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%)

(1) 模型架構

將 CNN 層數減半為 2 層，batch normalization、ReLU、Max Pooling 的架構都和第一題相同。convolution layer 都使用 3*3 的 filters、stride = 1、padding =

1，第一層 input channels 為 3、output channels 為 64，第二層 output channels 為 128。

將所有 channels flatten 輸入和第一題架構相同的 feedforward network。其中第一層 linear layer 的 input 維度為 $128 \times 32 \times 32$ 、output 維度為 100，第二層的 output 維度為 64、第三層的 output 維度為 11。

Data augmentation 的部分和第一題的模型完全相同。

Optimizer、learning rate、epoch 都和第一題模型相同。

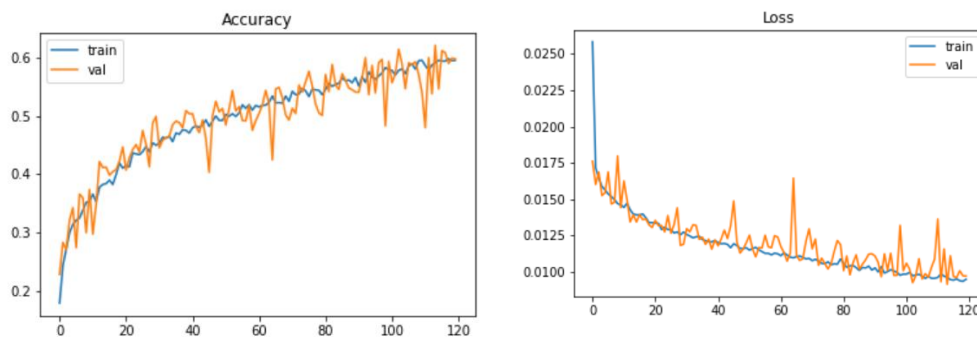
(2) 參數量：大約 13.1M。

(3) 準確率：

training accuracy : 0.595277、loss : 0.009469

validation accuracy : 0.597668、loss : 0.009701

將 training set 和 validation set 合併後的 training accuracy : 0.607852、
loss : 0.009059



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%)

(1) 模型結構

使用八層 linear layer、activation function 都使用 ReLU。

第一層 input 維度是 $128 \times 128 \times 3$ 、output 維度是 128。接下來二到八層的 output 維度分別是 2048、1024、1024、1024、1024、512、11。

Data augmentation 的部分和第一題的模型完全相同。

Optimizer、learning rate、epoch 都和第一題模型相同。

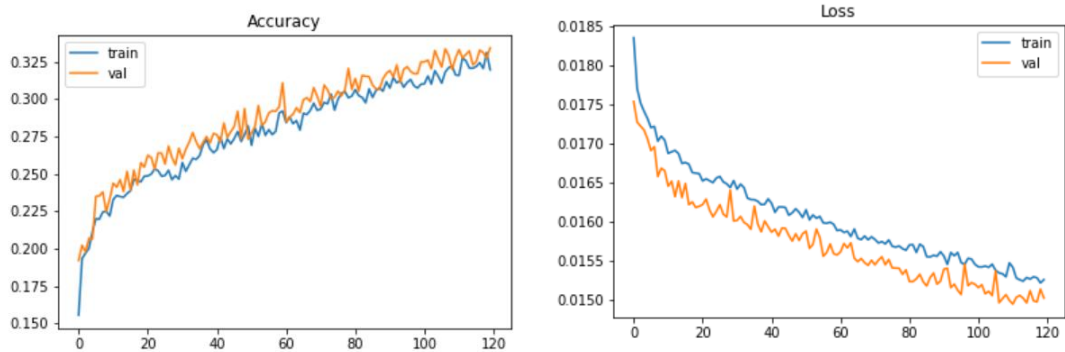
(2) 訓練參數：大約 12.3M。

(3) 準確率

training accuracy : 0.319481、loss : 0.015253

validation accuracy : 0.334110、loss : 0.015019

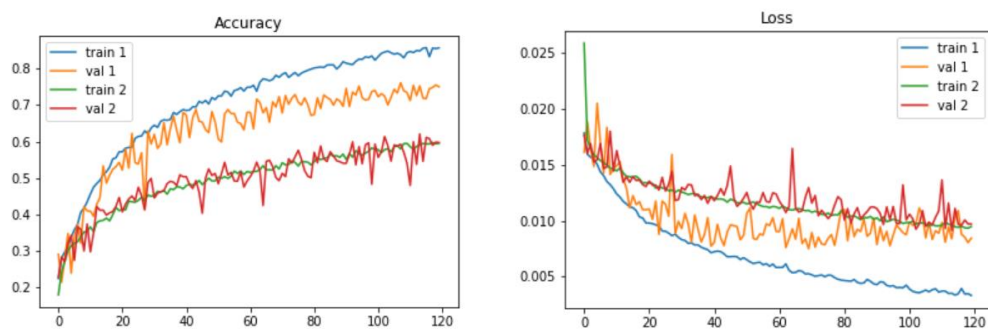
將 training set 和 validation set 合併後的 training accuracy : 0.320310 、 loss : 0.015198



4. 請說明由 1 ~ 3 題的實驗中你觀察到了什麼？(1%)

(1) CNN 深度減半

	Best model	CNN 深度減半
參數量	12.8M	13.1M
training accuracy	0.857186	0.595277
validation accuracy	0.750437	0.597668
合併後 training accuracy	0.890117	0.607852

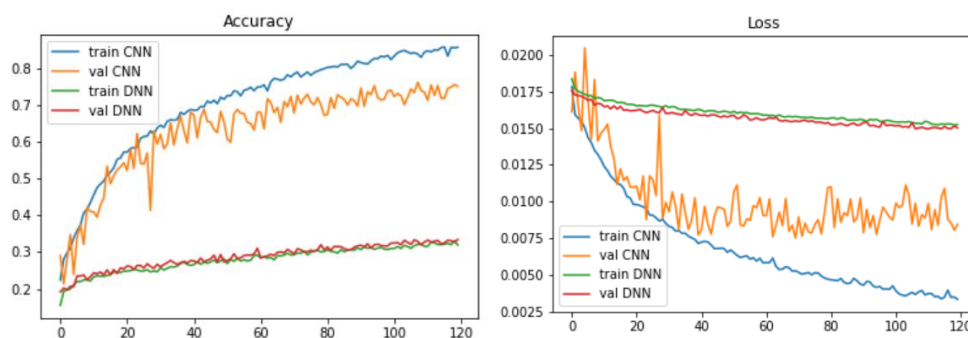


將 CNN 層數減半但保持大約相同的參數量，新模型在 training 和 validation set 上的表現都較差。training accuracy 只有大約 60%，推測這個模型 underfitting。因為 convolution layer 減半，經過的 Max Pooling 也減半，輸入第一層 fully connected feedforward network 的向量維度增加許多，造成大量的參數量消耗在這一層 network。因此在兩層 convolution layer 中能使用的 filter 數與原本相比便少很多。推測是因為 convolution layer 數太少、filter 數太少，造成學習率不佳。

(2) DNN 模型

	Best model	CNN 深度減半
參數量	12.8M	12.3M
training accuracy	0.857186	0.319481
validation accuracy	0.750437	0.334110

合併後 training accuracy	0.890117	0.320310
-----------------------	----------	----------



簡易的 DNN model 在 training 和 validation 上的表現都非常差。因為 CNN 可以學習影像上小範圍的 pattern、不同位置的同樣 pattern、並且用 subsample 的技術減少參數，在學習影像上較有效率。因此在同樣的參數量下，簡易的 DNN 模型的學習率較差。

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%)

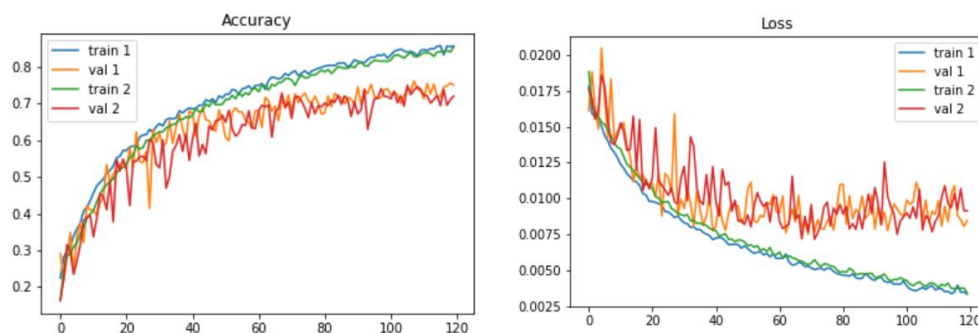
(1) Data normalization

a. 實作方法：

利用 `transform.ToTensor()` 將 image 轉換成 tensor 且 normalize 到 [0, 1] 之間。

b. 準確率比較

	With normalization	Without normalization
training accuracy	0.857186	0.854348
validation accuracy	0.750437	0.720408
training loss	0.003333	0.003436
validation loss	0.008438	0.009137



1: With normalization, 2: Without normalization

說明：使用 data normalization 的 model 的準確率較沒使用的略高一些。

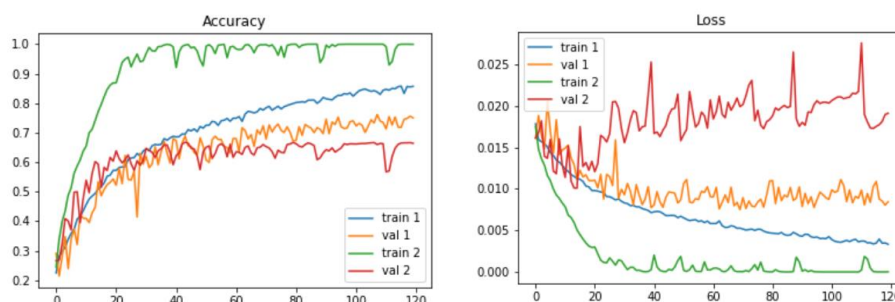
(2) Data Augmentation

a. 實作方法

利用 pytorch 的 transforms 在每次取用一個 batch 的資料時隨機對圖像水平翻轉 (RandomHorizontalFlip())、隨機旋轉 0 至 15 度 (RandomRotation(15))、隨機調整影像的亮度 (ColorJitter(brightness=0.5))、隨機將影像剪裁 (RandomResizedCrop(size=128,scale=(0.1,1.0),ratio=(0.75,1.33),interpolation=2))

b. 準確率比較

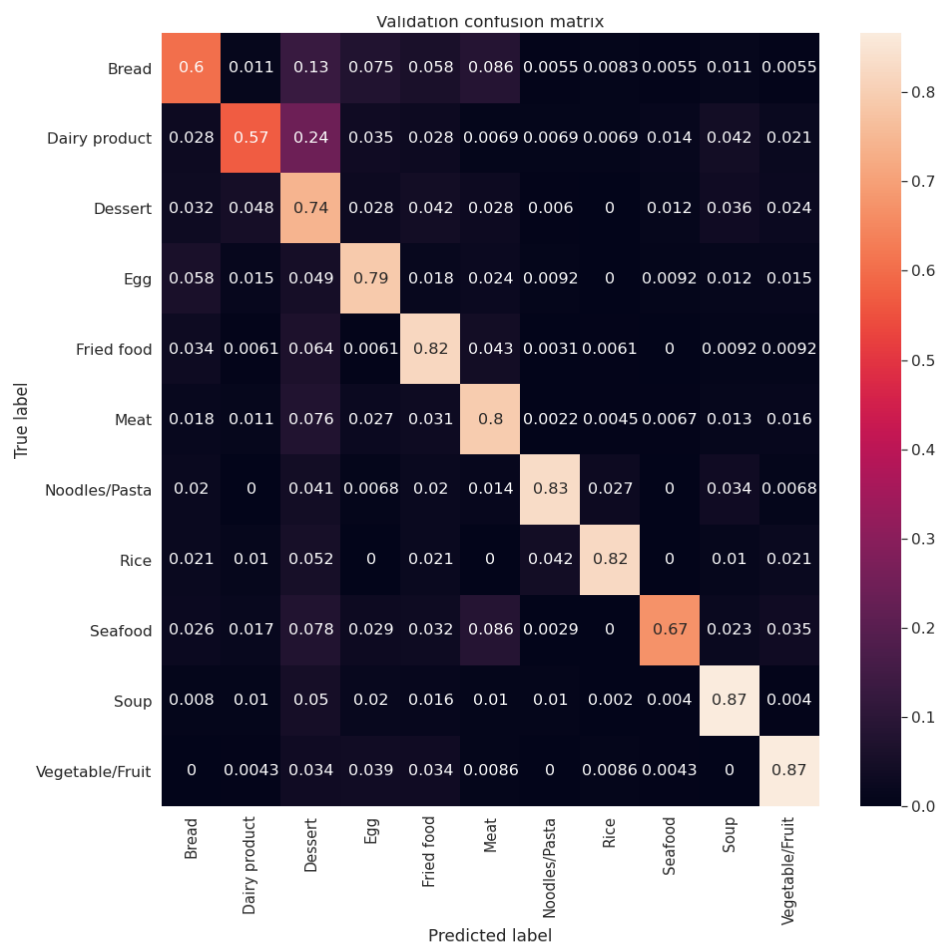
	With Data Augmentation	Without Data Augmentation
training accuracy	0.857186	0.999595
validation accuracy	0.750437	0.664140
training loss	0.003333	0.000018
validation loss	0.008438	0.019098



1: With data augmentation, 2: Without data augmentation

說明：使用 Data Augmentation 的 model 雖然 training accuracy 較低，但 validation accuracy 較高。因為使用 data augmentation training 時，使用的是不同於 training set 的資料，因此 training accuracy 會下降；但因為 train 的資料更多樣化，整體模型會更強，所以 validation accuracy 會上升。觀察 loss v.s. epoch 圖，發現 without data augmentation 的模型在 epoch 大約等於 20 的時候，validation loss 達到最低值，之後便 over-fitting 了。因為 without data augmentation 的模型是重複在相同的資料上 train，資料量較小，因此容易 over-fitting。

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%)



分析：

- (1) Bread 容易辨識成 Dessert(0.13)和 Meat(0.086)。
- (2) Dairy product 容易辨識成 Dessert(0.24)。
- (3) Fried food 容易辨識成 Dessert(0.064)。
- (4) Meat 容易辨識成 Dessert(0.076)。
- (5) Seafood 容易辨識成 Dessert(0.078)、Meat(0.086)。
- (6) Bread(0.6) 和 Dairy product (0.57)的辨識率最差。

結論：

在這次實驗中，Dessert 最容易和其他 class 混淆。