

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%)

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 64, 128, 128]	1,792
BatchNorm2d-2	[-1, 64, 128, 128]	128
ReLU-3	[-1, 64, 128, 128]	0
MaxPool2d-4	[-1, 64, 64, 64]	0
Conv2d-5	[-1, 128, 64, 64]	73,856
BatchNorm2d-6	[-1, 128, 64, 64]	256
ReLU-7	[-1, 128, 64, 64]	0
MaxPool2d-8	[-1, 128, 32, 32]	0
Conv2d-9	[-1, 256, 32, 32]	295,168
BatchNorm2d-10	[-1, 256, 32, 32]	512
ReLU-11	[-1, 256, 32, 32]	0
Conv2d-12	[-1, 256, 32, 32]	590,080
BatchNorm2d-13	[-1, 256, 32, 32]	512
ReLU-14	[-1, 256, 32, 32]	0
MaxPool2d-15	[-1, 256, 16, 16]	0
Conv2d-16	[-1, 512, 16, 16]	1,180,160
BatchNorm2d-17	[-1, 512, 16, 16]	1,024
ReLU-18	[-1, 512, 16, 16]	0
Conv2d-19	[-1, 512, 16, 16]	2,359,808
BatchNorm2d-20	[-1, 512, 16, 16]	1,024
ReLU-21	[-1, 512, 16, 16]	0
MaxPool2d-22	[-1, 512, 8, 8]	0
Conv2d-23	[-1, 512, 8, 8]	2,359,808
BatchNorm2d-24	[-1, 512, 8, 8]	1,024
ReLU-25	[-1, 512, 8, 8]	0
MaxPool2d-26	[-1, 512, 4, 4]	0
Linear-27	[-1, 1024]	8,389,632
ReLU-28	[-1, 1024]	0
Dropout2d-29	[-1, 1024]	0
Linear-30	[-1, 1024]	1,049,600
ReLU-31	[-1, 1024]	0
Dropout2d-32	[-1, 1024]	0
Linear-33	[-1, 11]	11,275
=====		
Total params: 16,315,659		
Trainable params: 16,315,659		
Non-trainable params: 0		
=====		

我採用了類似 VGG-11 的模型，CNN 模型中我使用了 7 層 Convolution layer, 在每一組都用了 ReLU, BatchNormalization, Filter 數量是越來越多個。最後再將圖壓平，進入 3 層的 Fully-connected-network, DropOut rate 設為 0.5，產生 11 種分類。在訓練的過程中使用 Adam 當作 optimizer，整個模型大致用了 16,000,000 個參數。Convolution layer 中大概有 6500000 個參數。在分 training set 和 testing set 的測試當中，training accuracy 為 0.94，validation set accuracy 為 0.78。而最後合併 training set 和 testing set 的 best model 最後在 Kaggle 的到的分數為 Public:0.81231

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%)

```
~/Downloads/hw3 admin@1080ti 2m 10s
I > python hw3_cnn2.py
Reading data
Size of training data = 9866
Size of validation data = 3430
Size of Testing data = 3347
-----
Layer (type)                Output Shape                Param #
-----
Conv2d-1                    [-1, 256, 128, 128]        7,168
BatchNorm2d-2               [-1, 256, 128, 128]        512
ReLU-3                      [-1, 256, 128, 128]        0
MaxPool2d-4                 [-1, 256, 64, 64]         0
Conv2d-5                    [-1, 512, 64, 64]         1,180,160
BatchNorm2d-6               [-1, 512, 64, 64]         1,024
ReLU-7                      [-1, 512, 64, 64]         0
MaxPool2d-8                 [-1, 512, 32, 32]         0
Conv2d-9                    [-1, 512, 32, 32]         2,359,808
BatchNorm2d-10              [-1, 512, 32, 32]         1,024
ReLU-11                    [-1, 512, 32, 32]         0
MaxPool2d-12                [-1, 512, 16, 16]         0
Conv2d-13                   [-1, 512, 16, 16]         2,359,808
BatchNorm2d-14              [-1, 512, 16, 16]         1,024
ReLU-15                    [-1, 512, 16, 16]         0
MaxPool2d-16                [-1, 512, 4, 4]           0
Linear-17                   [-1, 1024]                 8,389,632
ReLU-18                    [-1, 1024]                 0
Dropout2d-19                [-1, 1024]                 0
Linear-20                   [-1, 1024]                 1,049,600
ReLU-21                    [-1, 1024]                 0
Dropout2d-22                [-1, 1024]                 0
Linear-23                   [-1, 11]                   11,275
=====
Total params: 15,361,035
Trainable params: 15,361,035
Non-trainable params: 0
```

CNN 模型中我改使用了 4 層 Convolution layer, 在每一組一樣都用了 ReLU, BatchNormalization, Filter 數量是越來越多個。最後再將圖壓平，進入 3 層的 Fully-connected-network, DropOut rate 設為 0.5，產生 11 種分類。在訓練的過程中一樣使用 Adam 當作 optimizer，整個模型大致用了 15,000,000 個參數。Convolution layer 中大概有 6000000 個參數。training accuracy 為 0.91, validation set accuracy 為 0.74。而最後合併 training set 和 testing set 的 best model 最後在 Kaggle 的到的分數為 Public:0.76210

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%)

Layer (type)	Output Shape	Param #
Linear-1	[-1, 128]	6,291,584
ReLU-2	[-1, 128]	0
BatchNorm1d-3	[-1, 128]	256
Dropout2d-4	[-1, 128]	0
Linear-5	[-1, 128]	16,512
ReLU-6	[-1, 128]	0
BatchNorm1d-7	[-1, 128]	256
Dropout2d-8	[-1, 128]	0
Linear-9	[-1, 11]	1,419
Total params: 6,310,027		
Trainable params: 6,310,027		
Non-trainable params: 0		

DNN 模型中我用了 3 層 Fully connected network, 每層都有用 Relu, BatchNormalization 和 Dropout, 總共參數量約為 650 萬。DNN 在 Kaggle 上的分數為 Public:0.31679

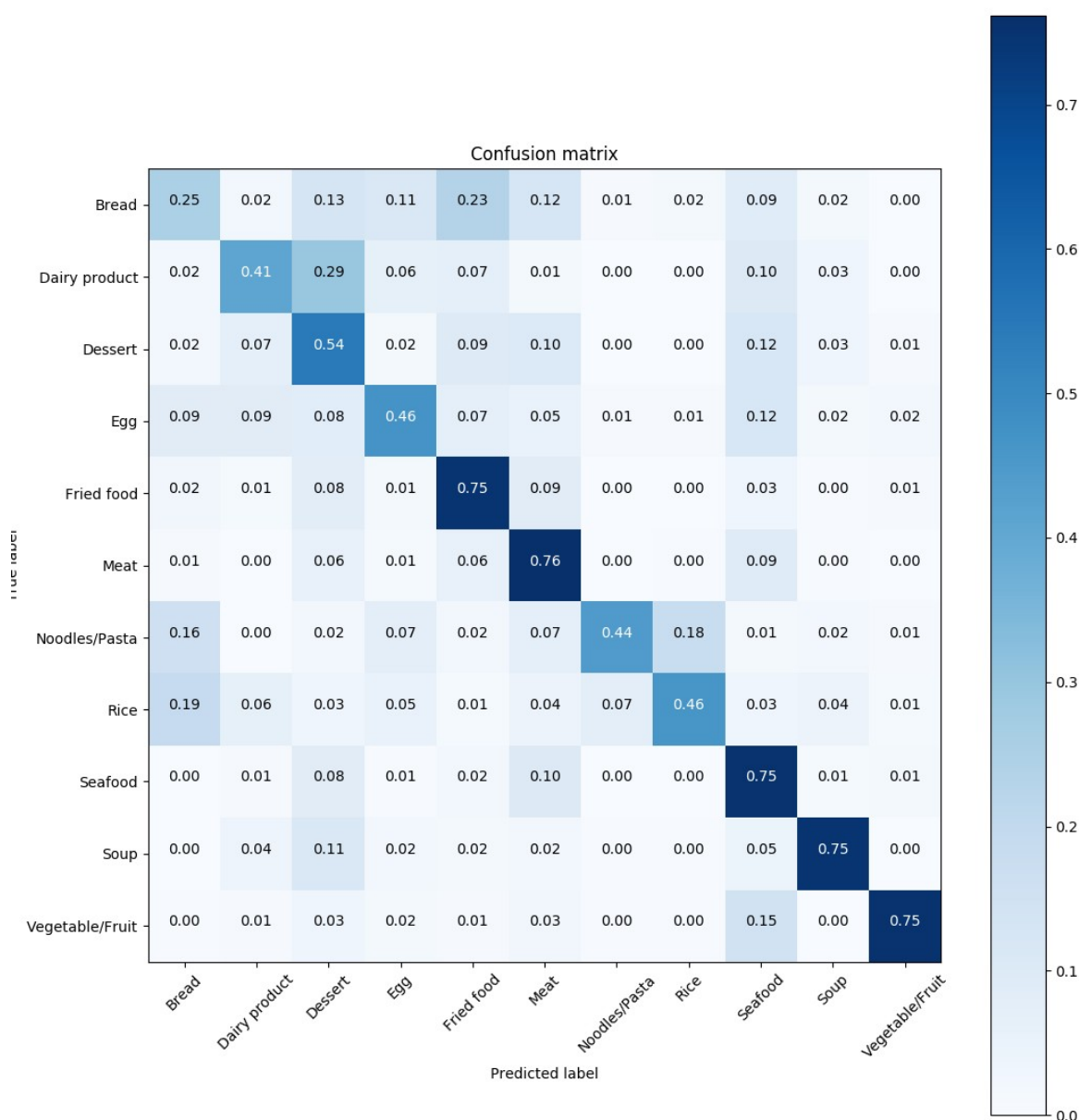
4. 請說明由 1 ~ 3 題的實驗中你觀察到了什麼？(1%)

DNN 明顯低了 CNN 不少。在經過相同的 Epoch, CNN 的準確率已經遠高於 DNN, 但是雖然用的參數量差不多, CNN 的訓練速度卻比 DNN 慢許多, 可能是因為 CNN 在用卷積層取出圖片特徵時的運算量較大。而我認為我做了 Data augmentation 也是讓 DNN 準確率不高的因素之一, 因為 DNN 並沒有像 CNN 能過濾圖片特徵, 可能是經過 augmentation 後產生的 noise 影響 DNN 的判斷。

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%)

因為 training 時間太常，我用了一個 convolution layer 三層，以及三層的 Fully-connected-network 的 CNN 做訓練。由於圖片中的每個元素皆為 0~255 的數值，因此我將每個元素同除以 255 做 Normalization，如此可以避免某些偏差太大的元素影響結果，在沒有經過標準化 前的分數為 Public:0.68709 Private:0.67372，經過標準化之後為 Public:0.69239 Private:0.69462，可以發現未標準化結果較差，尤其在 Private 的分數也有下降。至於 data augmentation 我是使用 `transforms.RandomHorizontalFlip()`, `transforms.RandomRotation(15)`，將圖片進行旋轉、翻轉、推移，產生更多的 data 來訓練 CNN 模型，如此可以增加抗躁性，並可以避免 overfitting。我使用 `BatchSize = 128`, `num_epoch = 80`。在沒有經過 data augmentation 的分數為 Public:0.68877 Private:0.68514，在經過 data augmentation 的分數為 Public:0.69239 Private:0.69462。

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%)



由於 train 時間過長，我將原本的 80 epoch 改為 epoch = 50，並畫在 validation set 上。由圖可看出 Bread、Dairy product、Egg、Noodle/Pasta 和 Rice 的準確率較低，尤其從 Bread 所在的 row 和 column 來看，Bread 最容易誤判成 Fried food；Dairy product 最容易誤判成 Dessert；Egg 容易誤判成很多；Noodle/Pasta 最容易誤判成 Bread 和 Rice；Rice 最容易誤判成 Bread。我認為是因為在 Training Data 中有些照片的，Bread、Fried food、Seafood 裡的 Rice、Noodle/Pasta，確實都白白黃黃一顆一顆的，對機器來說可能較困難去分辨。Egg 和其他相像，可能是因為蛋料理實在太多種，故分辨確實困難。