

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%)

以下為我的模型架構：

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 64, 128, 128]	1,792
Conv2d-2	[-1, 64, 128, 128]	36,928
BatchNorm2d-3	[-1, 64, 128, 128]	128
ReLU-4	[-1, 64, 128, 128]	0
MaxPool2d-5	[-1, 64, 64, 64]	0
Dropout-6	[-1, 64, 64, 64]	0
Conv2d-7	[-1, 128, 64, 64]	73,856
BatchNorm2d-8	[-1, 128, 64, 64]	256
ReLU-9	[-1, 128, 64, 64]	0
MaxPool2d-10	[-1, 128, 32, 32]	0
Dropout-11	[-1, 128, 32, 32]	0
Conv2d-12	[-1, 128, 32, 32]	147,584
BatchNorm2d-13	[-1, 128, 32, 32]	256
ReLU-14	[-1, 128, 32, 32]	0
MaxPool2d-15	[-1, 128, 16, 16]	0
Dropout-16	[-1, 128, 16, 16]	0
Conv2d-17	[-1, 512, 16, 16]	590,336
BatchNorm2d-18	[-1, 512, 16, 16]	1,024
ReLU-19	[-1, 512, 16, 16]	0
MaxPool2d-20	[-1, 512, 8, 8]	0
Dropout-21	[-1, 512, 8, 8]	0
Conv2d-22	[-1, 512, 8, 8]	2,359,808
BatchNorm2d-23	[-1, 512, 8, 8]	1,024
ReLU-24	[-1, 512, 8, 8]	0
MaxPool2d-25	[-1, 512, 4, 4]	0
Dropout-26	[-1, 512, 4, 4]	0
Linear-27	[-1, 1024]	8,389,632
ReLU-28	[-1, 1024]	0
Dropout-29	[-1, 1024]	0
Linear-30	[-1, 512]	524,800
ReLU-31	[-1, 512]	0
Dropout-32	[-1, 512]	0
Linear-33	[-1, 512]	262,656
ReLU-34	[-1, 512]	0
Linear-35	[-1, 11]	5,643
Total params: 12,395,723		

- Train Acc: 0.945145 | Val Acc: 0.802569

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%)

以下模型架構：

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 64, 128, 128]	1,792
MaxPool2d-2	[-1, 64, 64, 64]	0
Conv2d-3	[-1, 128, 64, 64]	73,856
BatchNorm2d-4	[-1, 128, 64, 64]	256
ReLU-5	[-1, 128, 64, 64]	0
MaxPool2d-6	[-1, 128, 32, 32]	0
Dropout-7	[-1, 128, 32, 32]	0
MaxPool2d-8	[-1, 128, 16, 16]	0
Conv2d-9	[-1, 512, 16, 16]	590,336
BatchNorm2d-10	[-1, 512, 16, 16]	1,024
ReLU-11	[-1, 512, 16, 16]	0
MaxPool2d-12	[-1, 512, 8, 8]	0
Dropout-13	[-1, 512, 8, 8]	0
MaxPool2d-14	[-1, 512, 4, 4]	0
Linear-15	[-1, 1024]	8,389,632
ReLU-16	[-1, 1024]	0
Dropout-17	[-1, 1024]	0
Linear-18	[-1, 512]	524,800
ReLU-19	[-1, 512]	0
Dropout-20	[-1, 512]	0
Linear-21	[-1, 512]	262,656
ReLU-22	[-1, 512]	0
Linear-23	[-1, 11]	5,643
Total params: 9,849,995		
Trainable params: 9,849,995		
Non-trainable params: 0		

- Train Acc: 0.901034 | Val Acc: 0.752799

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%)

以下為模型架構：

Layer (type)	Output Shape	Param #
Linear-1	[-1, 2048]	16,779,264
ReLU-2	[-1, 2048]	0
Dropout-3	[-1, 2048]	0
Linear-4	[-1, 128]	262,272
ReLU-5	[-1, 128]	0
Dropout-6	[-1, 128]	0
Linear-7	[-1, 512]	66,048
ReLU-8	[-1, 512]	0
Dropout-9	[-1, 512]	0
Linear-10	[-1, 512]	262,656
ReLU-11	[-1, 512]	0
Dropout-12	[-1, 512]	0
Linear-13	[-1, 11]	5,643
Total params: 17,375,883		
Trainable params: 17,375,883		
Non-trainable params: 0		
Input size (MB): 0.19		
Forward/backward pass size (MB): 0.07		
Params size (MB): 66.28		
Estimated Total Size (MB): 66.54		

- Train Acc: 0.201034 | Val Acc: 0.192799

4. 請說明由 1 ~ 3 題的實驗中你觀察到了什麼？(1%)

我們可以發現，將CNN層數減半的模型準確率較低，只有75%左右的正確率，但是運算速度快上許多。

而DNN模型做出來的準確率明顯低很多，只有20%左右，把照片拉成一條線再來

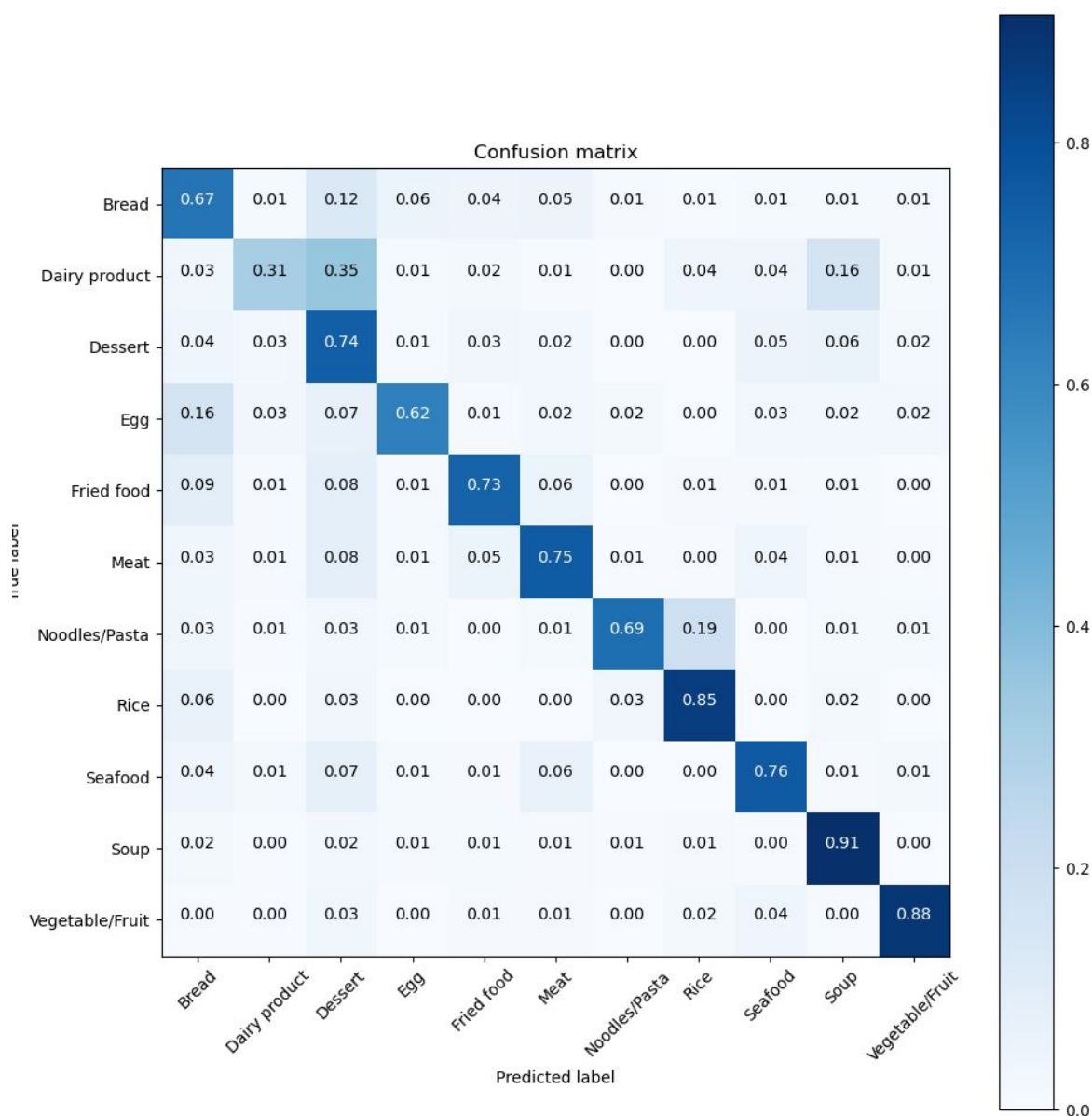
做機器學習果然效果比較不好。

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%)

- data normalization : 本來的to.Tensor就有Normalize的效果了, 所以我把他先拿掉, 用其他方式實作。之後再利用 transforms.Compose 裡面的 transforms.Normalize()來實作, 可以發現如果沒有標準化的話, model的準確率非常低只有30%左右。

- data augmentation : 利用 transforms.Compose 裡面的 transforms.RandomHorizontalFlip 以及 transforms.RandomRotation(15) 來實作, 發現在validation上準確率也差不多是80%, 沒有什麼影響。

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%)

由此圖我們可以發現湯的識別率最高，而乳製品的最低，且常被混淆成點心，我猜測可能是因為起司跟蛋糕的形狀非常相似。