

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

CNN 模型架構如下表所示，input 為 $3 * 256 * 256$ 之 image

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 32, 256, 256]	896
BatchNorm2d-2	[-1, 32, 256, 256]	64
ReLU-3	[-1, 32, 256, 256]	0
MaxPool2d-4	[-1, 32, 128, 128]	0
Conv2d-5	[-1, 64, 128, 128]	18,496
BatchNorm2d-6	[-1, 64, 128, 128]	128
ReLU-7	[-1, 64, 128, 128]	0
MaxPool2d-8	[-1, 64, 64, 64]	0
Conv2d-9	[-1, 128, 64, 64]	73,856
BatchNorm2d-10	[-1, 128, 64, 64]	256
ReLU-11	[-1, 128, 64, 64]	0
MaxPool2d-12	[-1, 128, 32, 32]	0
Conv2d-13	[-1, 256, 32, 32]	295,168
BatchNorm2d-14	[-1, 256, 32, 32]	512
ReLU-15	[-1, 256, 32, 32]	0
MaxPool2d-16	[-1, 256, 16, 16]	0
Conv2d-17	[-1, 512, 16, 16]	1,180,160
BatchNorm2d-18	[-1, 512, 16, 16]	1,024
ReLU-19	[-1, 512, 16, 16]	0
MaxPool2d-20	[-1, 512, 8, 8]	0
Conv2d-21	[-1, 512, 8, 8]	2,359,808
BatchNorm2d-22	[-1, 512, 8, 8]	1,024
ReLU-23	[-1, 512, 8, 8]	0
MaxPool2d-24	[-1, 512, 4, 4]	0
Linear-25	[-1, 1024]	8,389,632
Dropout-26	[-1, 1024]	0
ReLU-27	[-1, 1024]	0
Linear-28	[-1, 512]	524,800
Dropout-29	[-1, 512]	0
ReLU-30	[-1, 512]	0
Linear-31	[-1, 11]	5,643

訓練參數量：12851467

Validation accuracy：77.3178 %

Training accuracy：96.1180 %

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%)

實作 CNN 層數減半的模型如下，最後呈現為跑 150 次 epoch 之結果，input 為 $3 * 256 * 256$ 之 image。

Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 64, 256, 256]	4,864
BatchNorm2d-2	[-1, 64, 256, 256]	128
ReLU-3	[-1, 64, 256, 256]	0
MaxPool2d-4	[-1, 64, 64, 64]	0
Conv2d-5	[-1, 256, 64, 64]	409,856
BatchNorm2d-6	[-1, 256, 64, 64]	512
ReLU-7	[-1, 256, 64, 64]	0
MaxPool2d-8	[-1, 256, 16, 16]	0
Conv2d-9	[-1, 512, 16, 16]	3,277,312
BatchNorm2d-10	[-1, 512, 16, 16]	1,024
ReLU-11	[-1, 512, 16, 16]	0
MaxPool2d-12	[-1, 512, 8, 8]	0
Linear-13	[-1, 280]	9,175,320
Dropout-14	[-1, 280]	0
ReLU-15	[-1, 280]	0
Linear-16	[-1, 11]	3,091

訓練參數量：12872379

Validation accuracy：63.8776 %

Training accuracy：81.3298 %

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%)

實作 DNN 的模型如下，最後呈現為跑 150 次 epoch 後的結果，input 為 3*128*128 影像拉直過後的 vector。

Layer (type)	Output Shape	Param #
Linear-1	[-1, 260]	12,779,780
ReLU-2	[-1, 260]	0
Linear-3	[-1, 11]	2,871
Dropout-4	[-1, 11]	0

訓練參數量：12782651

Validation accuracy：25.4227 %

Training accuracy：24.7314 %

4. 請說明由 1 ~ 3 題的實驗中你觀察到了什麼？(1%)

- 不同 CNN 深度之 model 比較

比較不同 CNN 深度的 model，會發現深度較深的 CNN 在 training 上準確率可以高達 96%，跟深度較淺的 CNN 比起來高出許多，同時在 validation 上，深度較深的 CNN 也表現得較佳，似乎可以推測出在同樣參數的情況下，深度較深的 model 表現會比較好。(撇除層數太深造成 gradient vanishing 的問題)

- CNN 跟 DNN 之 model 比較

在此可以明顯看出 CNN 的 performance 比 DNN 好太多，如同老師影片所說，CNN 在處理影像時有許多優點，兩者比較起來，DNN 只是把 image 所有 pixel 拉成很長的 vector，而且單一 pixel 的數值對影像分類問題來說能夠參考的資訊非常低，因此 DNN 的結果自然就不會到那麼好。

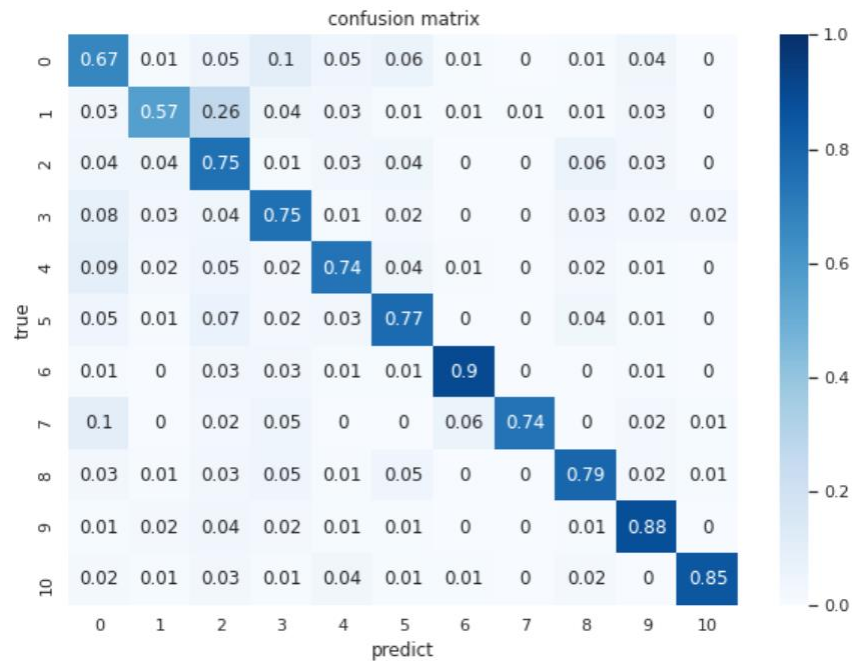
5. 請嘗試 data normalization 及 data augmentation，說明實作方法並且說明實行前後對準確率有什麼樣的影響？(1%)

以下針對 best model 去嘗試是否實作 data normalization 以及 data augmentation 的影響，此處不做 data normalization 我是直接將[0,1]的 tensor 轉換成[0,255]，以下準確率為 validation set

	有 augmentation	無 augmentation
有 normalization	77.3178 %	65.0146 %
無 normalization	76.7347 %	67.0554 %

從上面結果可以發現是否實作 augmentation 會對準確率造成較大的影響，最主要是因為 data augmentation 可以透過旋轉、調整大小的方式，創造更多圖片讓 model 學習，可以解決原本 model 出現 overfit 的問題，至於 normalization 的影響不大，我認為因為 normalization 最大的好處是可以針對不同維度且不同的 scale 做統一標準化的處理，而在影像裡每個 pixel 都是介於 0-255 之間，scale 的差距並不大，因此 normalization 的影響才不像之前的作業有太大的影響。

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%)



從上圖來看會發現我的 model 在 label 1 的圖片滿容易把它認為是 label 2，因此我特地挑出圖片來看看



True : 1、Predict : 2

label 1 我自己是判斷奶類的製品，例如牛奶、奶油等等，而 label 2 像是甜點類的食物，例如蛋糕、馬卡龍、冰淇淋等等，然而實際上有些 label 1 的圖我也會分不出其跟 label 2 的差異，因此 model 會混淆也是合情合理。上圖是其中一個認錯的圖片，我認為上圖三角形部分在 model 裡很容易會被認成蛋糕的形狀，很可能是這個原因導致 model 預測錯誤，但說真的這張圖我認為這張圖片歸類在 label 2 應該也不算答錯。