資料處理&分析

我先將各個資料類別編號，並將資料夾重新命名，後續方便進行訓練，下列是編號結果:

編號	菜名	編號	菜名	編號	菜名	編號	菜名	編號	菜名
0	三杯雞	10	柳丁	20	番茄炒蛋	30	葡萄	40	香酥魚排
1	什錦炒麵	11	棗子	21	白米飯	31	蒜泥白肉	41	馬鈴薯燉肉
2	咖哩雞	12	橘子	22	白菜滷	32	蒸蛋	42	高麗菜
3	塔香海茸	13	沙茶肉片	23	福山萵苣	33	蓮霧	43	鳳梨
4	大陸妹	14	油菜	24	空心菜	34	螞蟻上樹	44	鵝白菜
5	客家小炒	15	洋蔥炒蛋	25	糖醋雞丁	35	西瓜	45	鹽酥雞
6	小番茄	16	滷蛋	26	紅蘿蔔炒蛋	36	豆芽菜	46	麥克雞塊
7	有機小松菜	17	滷雞腿	27	義大利麵	37	關東煮	47	麻婆豆腐
8	有機青松菜	18	玉米炒蛋	28	芥藍菜	38	青江菜	48	麻油雞
9	木瓜	19	瓜仔肉	29	菠菜	39	香蕉	49	黑胡椒豬柳

資料處理&分析

接著我將拿到的資料拆分Train set 跟 Test set，按照投影要求保持在8:2的狀態，另外我是將每一類的所有圖片 個別拆成8:2，所以Train set跟Test set會有類似的分布，另外我將每一類都挑出125張做成測試資料集(共6250張)，因為我要事先拿來做模型效能分析，這樣就不需要每個模型都看過將近10萬筆的資料來比較效能，既能節省時間又符合經濟效益。那Data Augmentation也是有做，主要就是做旋轉、上下顛倒等，沒做顏色抖動或對比度調整是因為我覺得應該要讓食物原來的狀況能夠呈現，對於一些菜品來說，顏色應該是重要的資訊不應該隨便亂更動。

完整的資料集數量: Train set: 84282 images, Test set: 21096 images

測試的資料集數量: Train set: 5000 images, Test set: 1250 images

Model 架構探討

我選了5個模型來做效能分析，分別是AlexNet、VGG19、DenseNet201、ResNet152、ResNeXt101，這邊我盡量使用大的模型架構是因為會有10萬筆左右的資料，模型要大一些才可能會有較佳的泛化能力，這邊做模型效能分析使用的資料集是我整理過後的6250張圖片的測試資料集，我將使用以下的指標來評估最後要拿所有資料來訓練的模型。

1. Top1-Accuracy: 模型對輸入樣本的預測中，最高概率的預測結果是否正確。
2. Top5-Accuracy: 模型對輸入樣本的預測中，最高概率的 前5個類別 中是否包含正確類別。
3. Precision: 在模型預測為正類的樣本中，有多少是真正的正類。(TP/(TP+FP))
4. Recall: 在所有實際為正類的樣本中，有多少被模型正確預測為正類。(TP/(TP+FN))

Model 架構探討

Model	AlexNet	VGG19	DenseNet201	ResNet152	EfficientNetB5
Top1-Accuracy	0.5824	0.6664	0.7448	0.6968	0.7224
Top5-Accuracy	0.8760	0.9424	0.9576	0.9232	0.9496
Precision	0.6215	0.6836	0.7584	0.7436	0.7325
Recall	0.5824	0.6664	0.7448	0.6968	0.7224

這邊只比較模型架構的Performance，所以每個模型的訓練超參數都是一樣的，只是因為分類是分50類，所以我都在最後一層額外加上有50類的線性層。

超參數:

Epochs: 20, Batch Size: 32, Learning rate: 1e-4, Image size: 224x224

Loss function: Cross Entropy, Optimizer: Adam

測試的資料集: Train set: 5000 images, Test set: 1250 images

Model 架構探討

根據實驗結果我發現DenseNet201、ResNet152、EfficientNetB5的效果較好，但我最後選用ResNet152為正式訓練的模型，因為ResNet152在影像分類的任務上表現的最好，尤其是在大型資料集(Ex. ImageNet)上，那這次營養午餐的資料集有將近10萬張的影像，所以我評估應該ResNet152的效果會最好，雖然DenseNet201、 EfficientNetB5在5000張的表現上較好，但DenseNet201比較適合拿來處理醫學影像，EfficientNetB5參數量比較少泛化性較差但計算效率會較佳，然後依照ppt要做transfer learning的要求，所以我這裡使用的都是Pretrained Model。

接下來會用全部的資料訓練

完整的資料集數量: Train set: 84282 images, Test set: 21096 images

Model 架構探討

ResNet152架構(用summary呈現)，中間略。

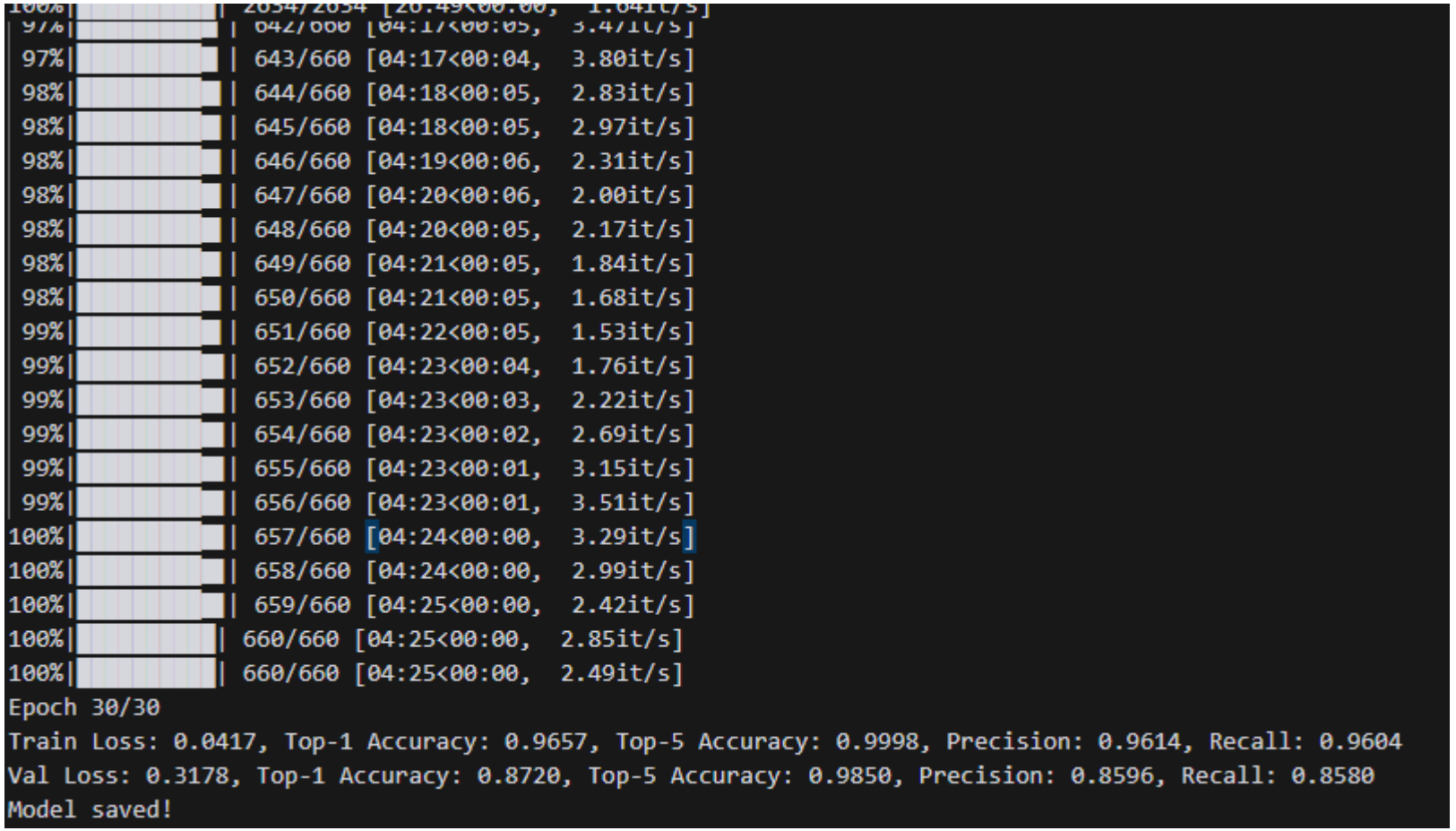
Layer (type)	Output Shape	Param #
Conv2d-1	[-1, 64, 112, 112]	9,408
BatchNorm2d-2	[-1, 64, 112, 112]	128
ReLU-3	[-1, 64, 112, 112]	0
MaxPool2d-4	[-1, 64, 56, 56]	0
Conv2d-5	[-1, 64, 56, 56]	4,096
BatchNorm2d-6	[-1, 64, 56, 56]	128
ReLU-7	[-1, 64, 56, 56]	0
Conv2d-8	[-1, 64, 56, 56]	36,864
BatchNorm2d-9	[-1, 64, 56, 56]	128
ReLU-10	[-1, 64, 56, 56]	0
Conv2d-11	[-1, 256, 56, 56]	16,384
BatchNorm2d-12	[-1, 256, 56, 56]	512
Conv2d-13	[-1, 256, 56, 56]	16,384
BatchNorm2d-14	[-1, 256, 56, 56]	512
ReLU-15	[-1, 256, 56, 56]	0
Bottleneck-16	[-1, 256, 56, 56]	0
Conv2d-17	[-1, 64, 56, 56]	16,384
BatchNorm2d-18	[-1, 64, 56, 56]	128
ReLU-19	[-1, 64, 56, 56]	0
Conv2d-20	[-1, 64, 56, 56]	36,864

BatchNorm2d-504	[-1, 512, 7, 7]	1,024
ReLU-505	[-1, 512, 7, 7]	0
Conv2d-506	[-1, 512, 7, 7]	2,359,296
BatchNorm2d-507	[-1, 512, 7, 7]	1,024
ReLU-508	[-1, 512, 7, 7]	0
Conv2d-509	[-1, 2048, 7, 7]	1,048,576
BatchNorm2d-510	[-1, 2048, 7, 7]	4,096
ReLU-511	[-1, 2048, 7, 7]	0
Bottleneck-512	[-1, 2048, 7, 7]	0
AdaptiveAvgPool2d-513	[-1, 2048, 1, 1]	0
Linear-514	[-1, 50]	102,450

Total params: 58,246,258
Trainable params: 58,246,258
Non-trainable params: 0

Input size (MB): 0.57
Forward/backward pass size (MB): 606.58
Params size (MB): 222.19
Estimated Total Size (MB): 829.35

Model Training 過程



Model Training 結果

	Training	Validation
Top-1 Accuracy	0.9657	0.8720
Top-5 Accuracy	0.9998	0.9850
Precision	0.9614	0.8596
Recall	0.9604	0.8580

上方是最後一個Epoch的結果

使用的模型: 預訓練的ResNet152(Transfer Learning)

完整的資料集數量: Train set: 84282 images, Test set: 21096 images

超參數:

Epochs: 30, Batch Size: 32, Learning rate: 1e-4, Image size: 224x224

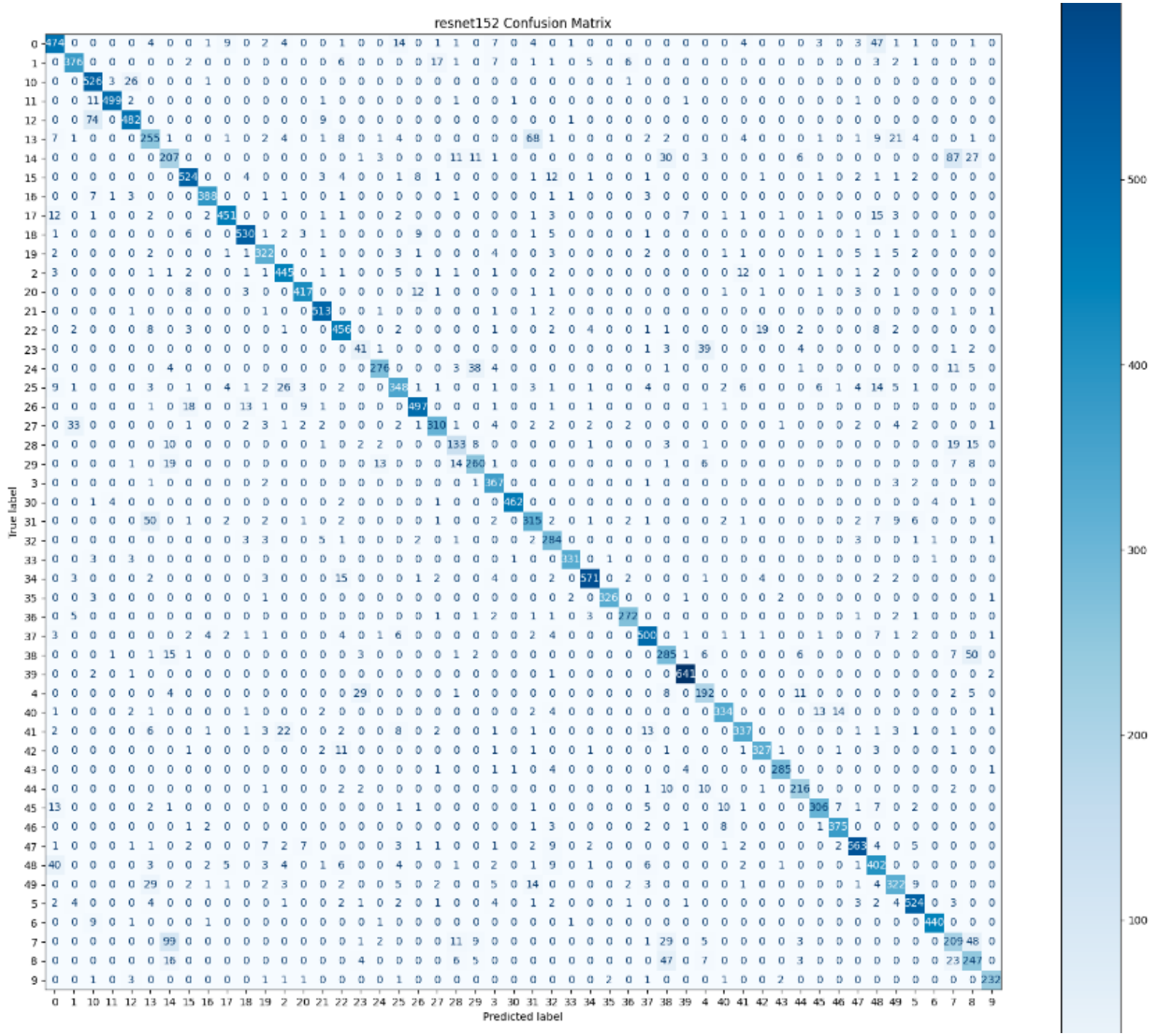
Loss function: Focal Loss, Optimizer: Adam

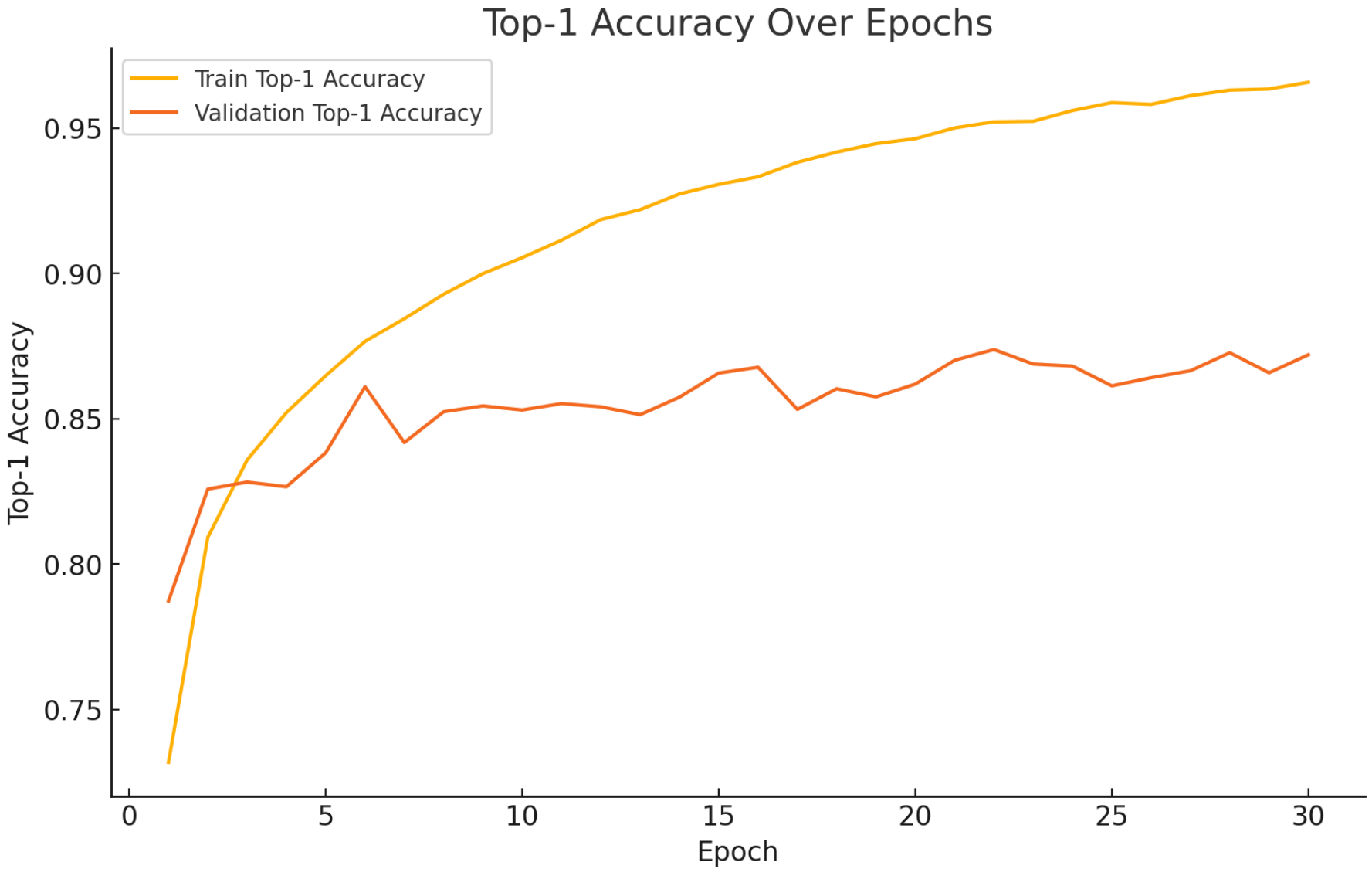
這邊使用Focal Loss是為了解決資料不平衡的問題

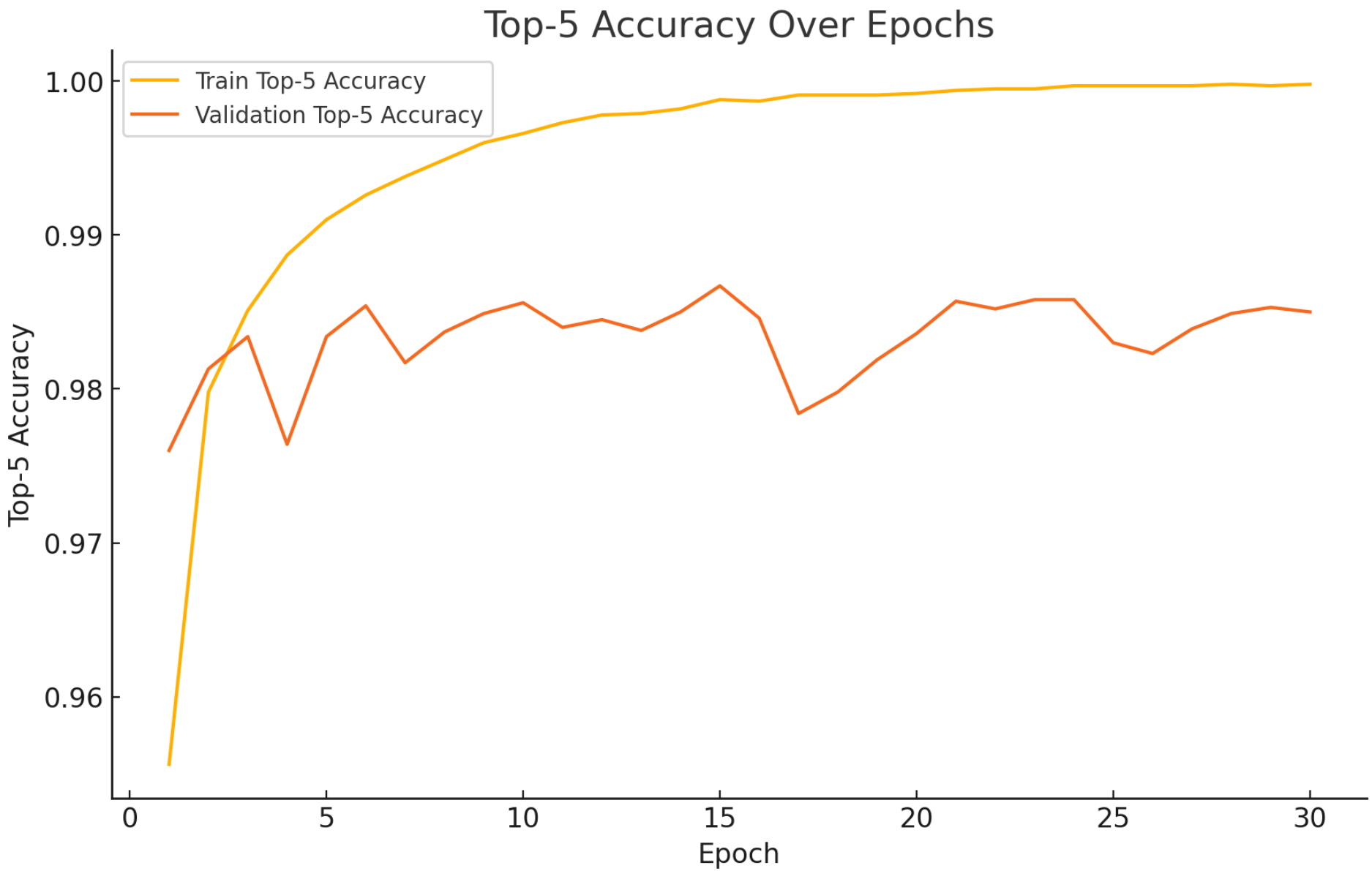
Model Training 結果

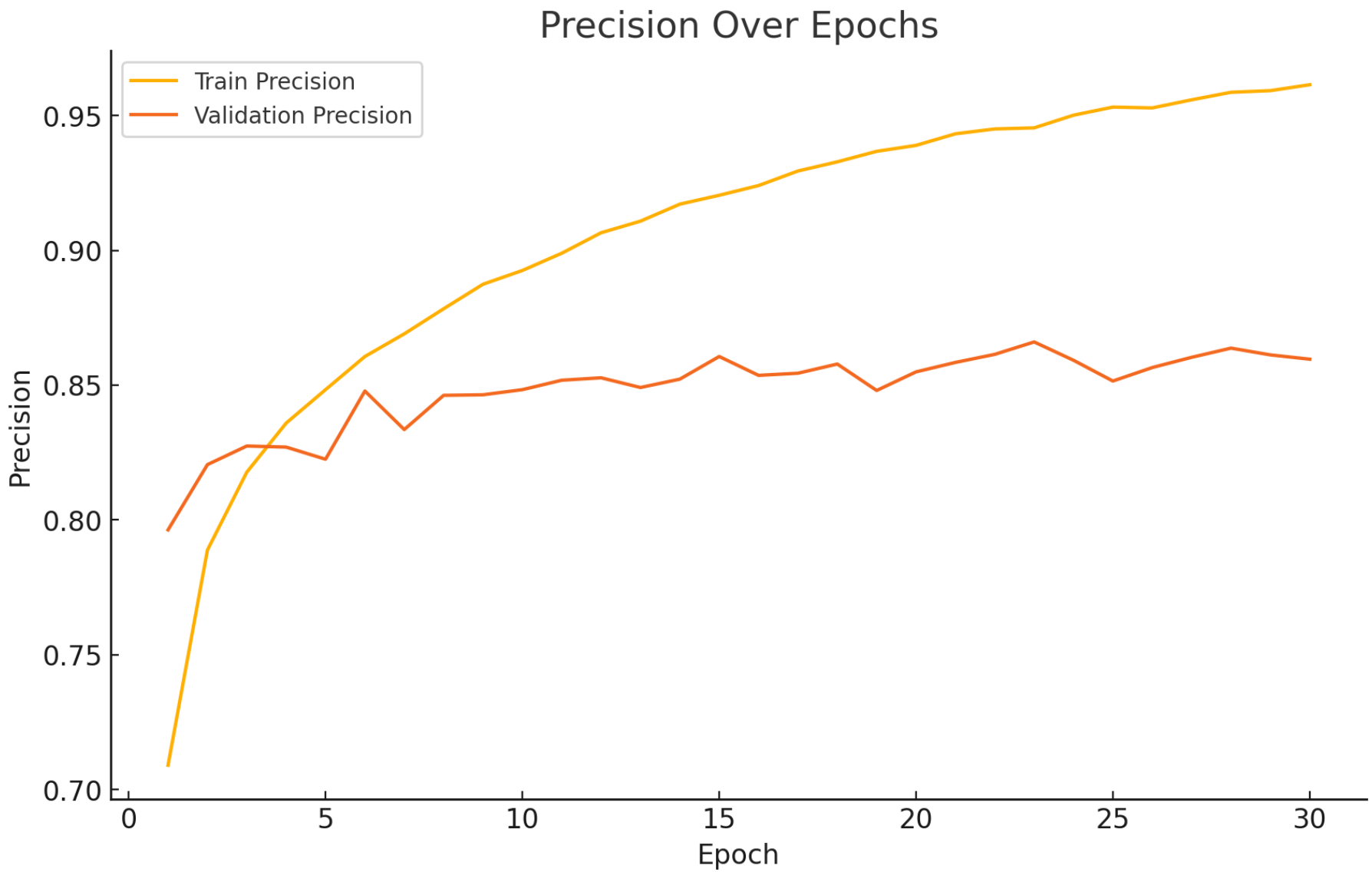
	Training	Validation
Top-1 Accuracy	0.9657	0.8720
Top-5 Accuracy	0.9998	0.9850
Precision	0.9614	0.8596
Recall	0.9604	0.8580

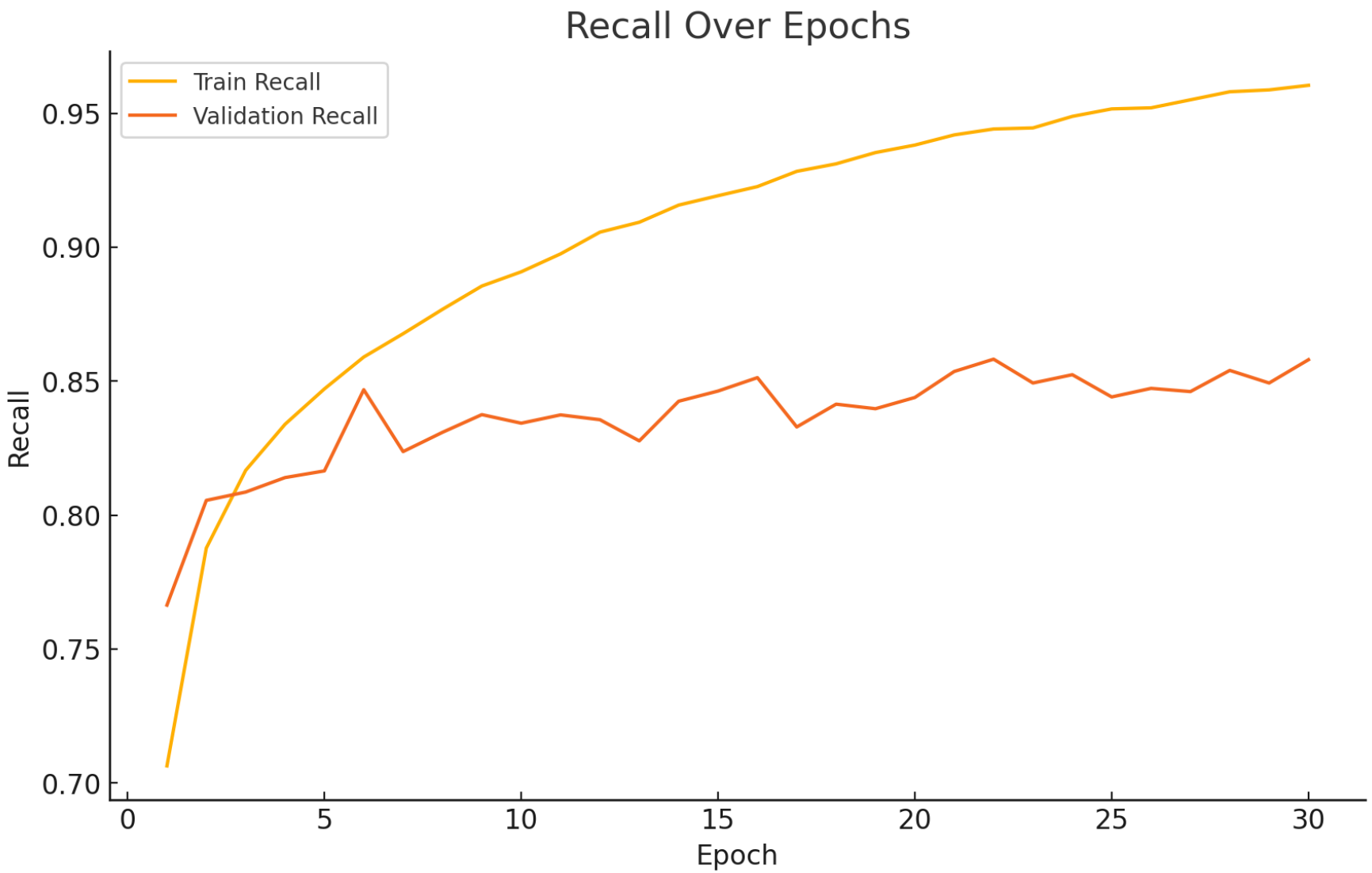
從上述的資訊可以發現這個模型可以正確辨識大多數的營養午餐菜色，Training的Top-1-Accuracy、Precision、Recall在0.96左右， Top-5 Accuracy將近是1 ，Validation的Top-1-Accuracy、Precision、Recall大致上在0.86左右，在Top-5 Accuracy可以達到接近0.99，所以這個用8萬筆訓練資料訓練的模型，可信度還是有的。


resnet152 Confusion Matrix
True label
Predicted label
0 1 10 11 12 13 14 15 16 17 18 19 2 20 21 22 23 24 25 26 27 28 29 3 30 31 32 33 34 35 36 37 38 39 4 40 41 42 43 44 45 46 47 48 49 5 6 7 8 9
500
400
300
200
100


Top-1 Accuracy Over Epochs
Train Top-1 Accuracy
Validation Top-1 Accuracy
Top-1 Accuracy
0.95
0.90
0.85
0.80
0.75
0
5
10
15
20
25
30
Epoch

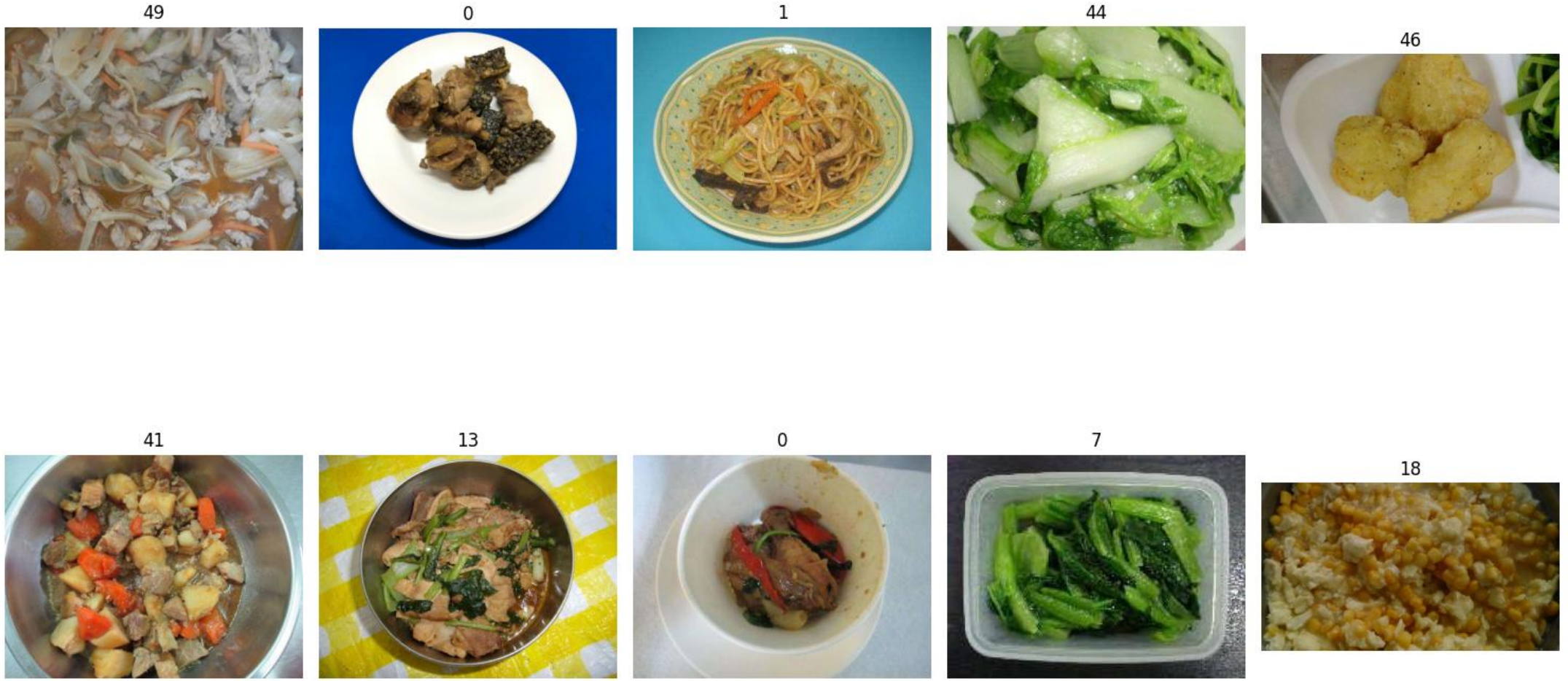

Top-5 Accuracy Over Epochs
Train Top-5 Accuracy
Validation Top-5 Accuracy
1.00
0.99
0.98
0.97
0.96
Top-5 Accuracy
0
5
10
15
20
25
30
Epoch


Precision Over Epochs
Train Precision
Validation Precision
Precision
0.95
0.90
0.85
0.80
0.75
0.70
0
5
10
15
20
25
30
Epoch


Recall Over Epochs
Train Recall
Validation Recall
Recall
0.95
0.90
0.85
0.80
0.75
0.70
0
5
10
15
20
25
30
Epoch


Training and Validation Loss Over Epochs
Train Loss
Validation Loss
Loss
0.5
0.4
0.3
0.2
0.1
0
5
10
15
20
25
30
Epoch

推理效果



推理效果

結果討論

上幾頁是訓練過程的圖和混淆矩陣(Confusion Matrix)的結果，但label的順序不是照數字大小是照字串的順序排所以會是”0”、”1”、”10”、”11”、…….、”2”、”20”、”21”、……、”7”、”8”、”9”，可以發現有些很類似菜色對模型來說是很難分類的，像是難分類的菜有:

- 0(三杯雞),17(滷雞腿),25(糖醋雞丁),45(鹽酥雞),48(麻油雞)
- 24(空心菜),29(菠菜)
- 1(什錦炒麵),27(義大利麵)
- 40(香酥魚排),45(鹽酥雞),46(麥克雞塊)
- 10(柳丁),12(橘子)
- 13(沙茶肉片),31(蒜泥白肉)
- 7(有機小松菜),8(有機青松菜),14(油菜),28(芥藍菜),29(菠菜),38(青江菜)

結果討論

- 15(洋蔥炒蛋),18(玉米炒蛋),20(番茄炒蛋),26(紅蘿蔔炒蛋)
- 2(咖哩雞),25(糖醋雞丁),41(馬鈴薯燉肉)
- 22(白菜滷),34(螞蟻上樹)
- 4(大陸妹),23(福山萵苣)

上述某些菜色的照片就算是人來看的話也不見得可以100%辨識成功。

心得: 這次的影像辨識任務我還訓練蠻久的，把8萬張拿去訓練ResNet152我就跑了兩天左右，所以我後來想要評估各個模型的效能就只能把部分挑出來訓練，不然每個模型都要跑那麼多資料會花很久的時間，總的來說，這次的作業讓我學到該如何正確的預處理大量的資料集，並用有效率的方式進行分析。