

深度學習第二次作業

使用 CNN 做玉米葉影像病害

分類實驗。

學生:

114206109 林睿杰

日期 2025/11/18

目錄

1. 實驗目的
2. 資料介紹
3. 資料處理
4. 實驗方式
5. 實驗結果
6. 結論

實驗目的

使用 CNN 做玉米葉影像病害分類實驗，並討論前處理方式及各種方法及
參數所訓練之模型效果，並用常用指標(accuracy、precision、recall、
F1_score 等)判斷模型成效

資料介紹

資料來源

Corn or Maize Leaf Disease Dataset

<https://www.kaggle.com/datasets/smaranjitghose/corn-or-maize-leaf-disease-dataset>

資料集描述

有 4 類影像資料為

- 0: Common Rust (常見鏽蝕) - 1306 images
- 1: Gray Leaf Spot (灰斑病) - 574 images
- 2: Blight (枯萎病) -1146 images
- 3: Healthy (健康)- 1162 images

資料集筆記

(本資料集基於廣受歡迎的 PlantVillage 和 PlantDoc 資料集建構而成。在資料集建構過程

中，我們移除了一些被認為無用的圖像)

資料處理

由於資料分布為

(Common Rust -1306、 Gray Leaf Spot -574、 Blight -1146、 Healthy -1162) ,

其中 Gray Leaf Spot 的數量相較於其他分類過少導致整體樣本的不平均，所以在此

資料集中針對 Gray Leaf Spot 去做資料擴充到樣本為 1200 筆解決樣本的不平均的問題，資料擴充方法為

題，資料擴充方法為

1. 先複製完整資料一份到新資料夾
2. 隨機從 現有 Gray_Leaf_Spot 圖片挑一張，做以下變化
 - 50% 機率左右翻轉
 - 角度的隨機偏轉(-20 到 +20)
 - 亮度，飽和度對比度的隨機變化(0.8 到 1.2)

擴充完樣本數為

- Common Rust -1306、
- Gray Leaf Spot -1200、
- Blight -1146、
- Healthy -1162

然後統一切成 256*256 pixel 的影像防止輸入錯誤

實驗方式

資料樣本切分

在這次實驗資料擴充完總筆數為 4812 (Common Rust -1306、Gray Leaf Spot - 1200、Blight -1146、Healthy -1162)

切割 訓練:驗證:測試為每類 0.7:0.1:0.2

結果：

Train: 3369

Val: 481

Test: 964

=== 每個子資料集類別統計 ===

Train 分布：{'Blight': 807, 'Common_Rust': 905, 'Healthy': 824, 'Gray_Leaf_Spot': 833}

Val 分布：{'Gray_Leaf_Spot': 115, 'Blight': 113, 'Common_Rust': 124, 'Healthy':

129}

Test 分布 : {'Common_Rust': 277, 'Healthy': 209, 'Gray_Leaf_Spot': 252, 'Blight':

226}

CNN 模型設計

模型設置為 4 個 block，每個 block 分別做一次

- 卷積 層(kernel size=3，padding=1)
- batch normalization
- ReLu
- 池化層

後面接 Fully Connected Layer 全連接層 做分類輸出 4 維 logits

之後由於我 loss funtion 是用 `nn.CrossEntropyLoss()`

- 它會把 logits(輸出) 轉成 *softmax* 的 *log* 機率
- 然後計算正確類別的負對數機率
- 反向傳播讓模型調整參數

再用 `torch.max()` 找出輸出 logits 最大的作為預測答案

維度變化為

$3 > 16 > 32 > 64 > 128$

Piexl 變化為

$$256*256 > 128*128 > 64*64 > 32*32 > 16*16$$

全連接為

$$128*16*16 > 256 > 4$$

實驗項目

實驗各超參數改變影響以及有預訓練權重的 ResNet 18、32 模型的指標(accuracy、precision、recall、F1_score 等)判斷模型成效

實驗項目有

Batch_size

LR

優化器的選擇

最終實驗較好參數的訓練與預訓練權重的 ResNet 18、32 模型比較

實驗結果(2025/11/18)

Batch size(以下簡稱 BS) 實驗

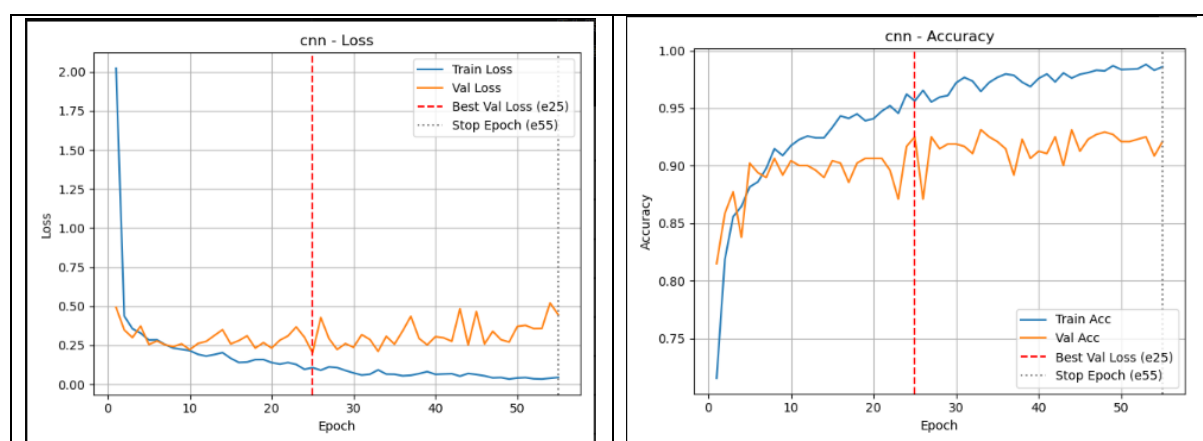
CNN

訓練驗證的準確及損失與回合對照

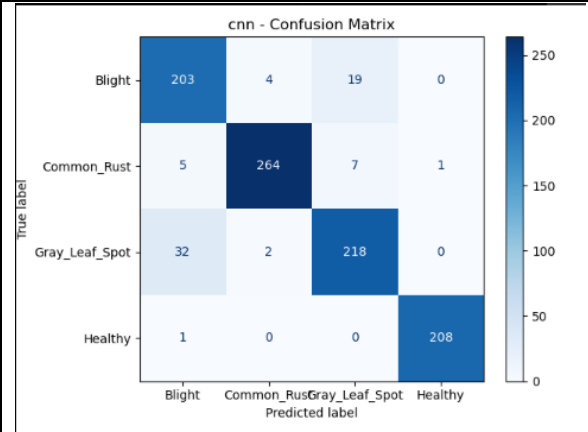
BS=64

參數為

- batch_size=64,
- IR=1e-3
- 優化器: adam
- 權重衰減 weight_decay=1e-4



測試結果(附帶 ResNet18、32 的指標比較)

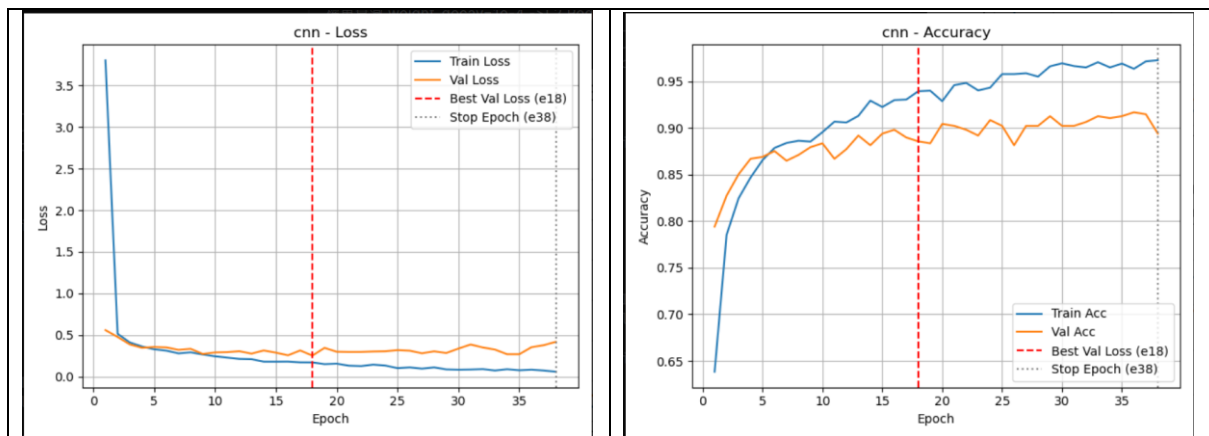


Metric	cnn	resnet18	resnet34
accuracy	0.9263	0.9627	0.9606
precision_macro	0.9272	0.9638	0.9609
recall_macro	0.9279	0.9637	0.9613
f1_macro	0.9272	0.9628	0.9607

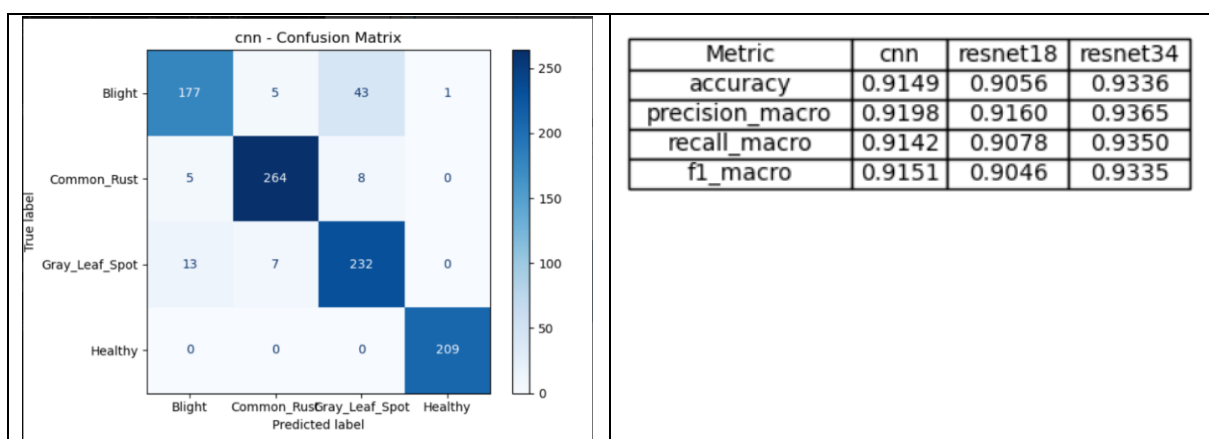
BS=128

參數為

- batch_size=128,
- IR=1e-3
- 優化器: adam
- 權重衰減 weight_decay=1e-4



測試結果(附帶 ResNet18、32 的指標比較)



比較結果 BS=64 較好

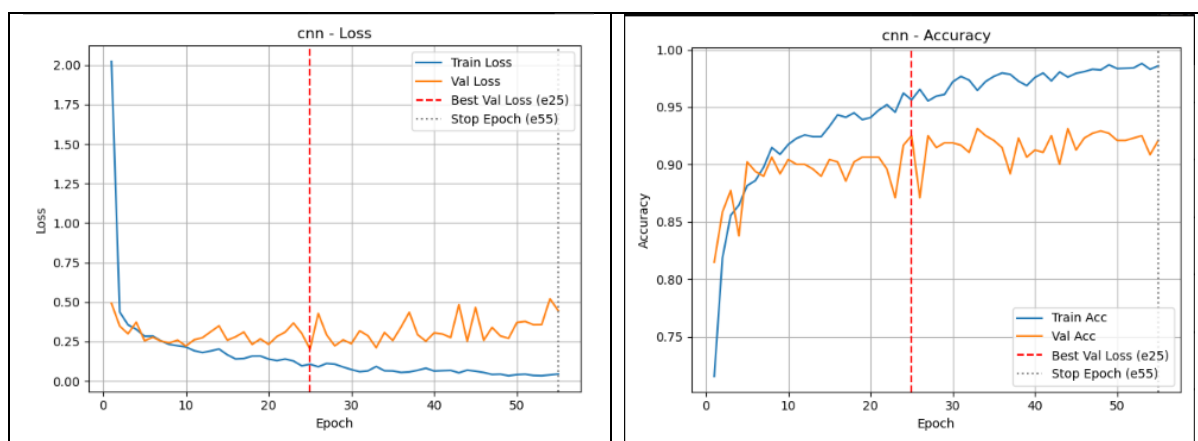
實驗 學習率

LR=1e-3

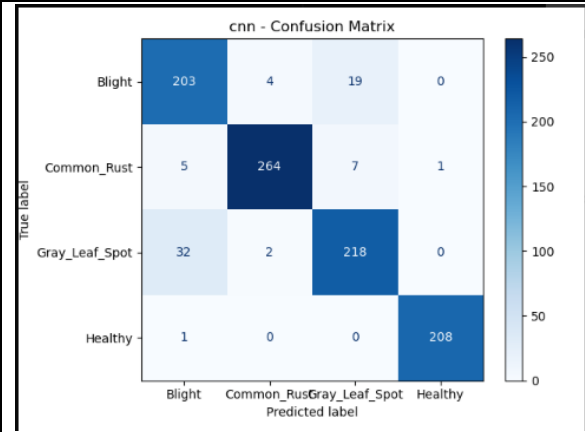
參數為

- batch_size=64,
- LR=1e-3
- 優化器: adam
- 權重衰減 weight_decay=1e-4

訓練驗證



測試結果(附帶 ResNet18、32 的指標比較)



Metric	cnn	resnet18	resnet34
accuracy	0.9263	0.9627	0.9606
precision_macro	0.9272	0.9638	0.9609
recall_macro	0.9279	0.9637	0.9613
f1_macro	0.9272	0.9628	0.9607

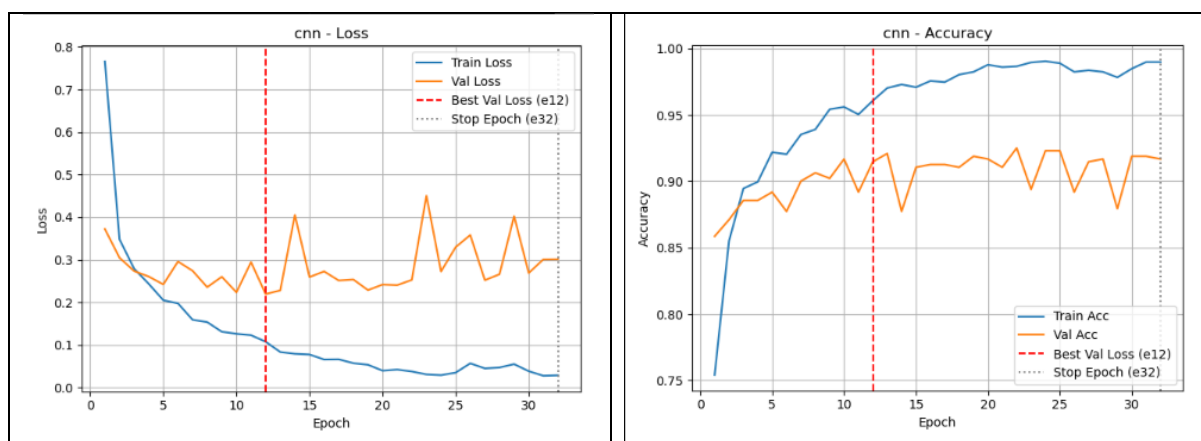
參 $R=3e-4$

BS=64

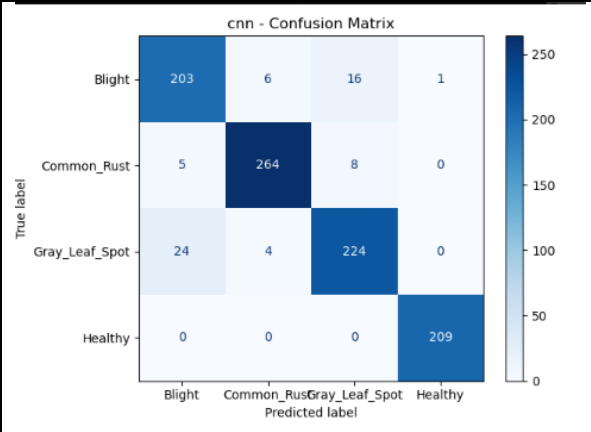
參數為

- batch_size=64,
- $IR=3e-4$
- 優化器: adam
- 權重衰減 $weight_decay=1e-4$

訓練驗證



測試



Metric	cnn	resnet18	resnet34
accuracy	0.9336	0.9637	0.9668
precision_macro	0.9342	0.9636	0.9669
recall_macro	0.9350	0.9642	0.9673
f1_macro	0.9346	0.9639	0.9669

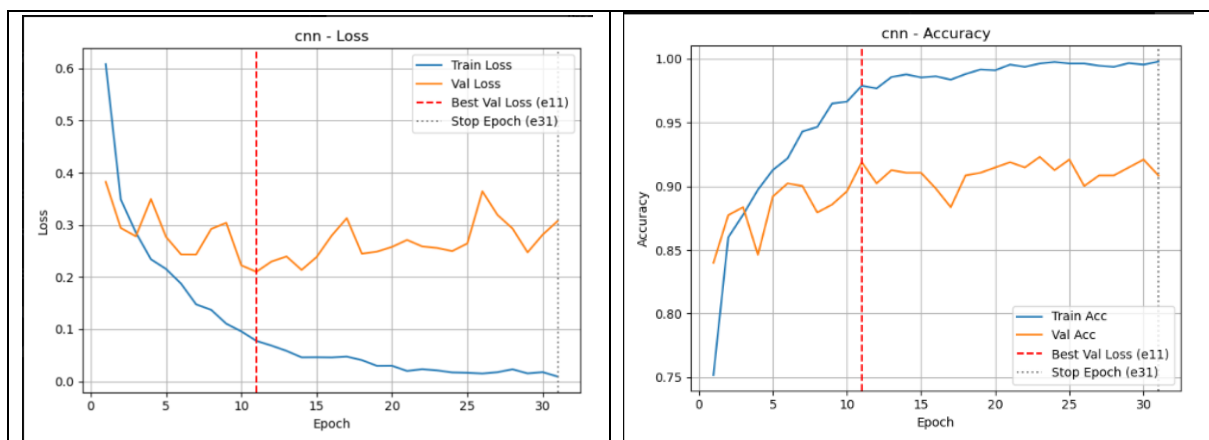
LR=1e-4

batch_size=64,

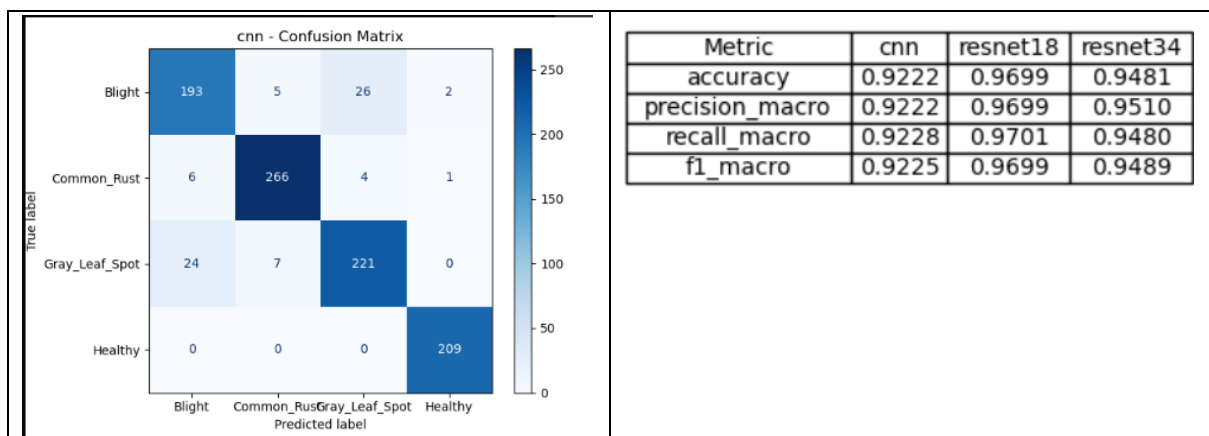
LR=1e-4

優化器: adam

權重衰減 weight_decay=1e-4 -> L2 Regularization



測試



測試下來 3e-4 是最好的

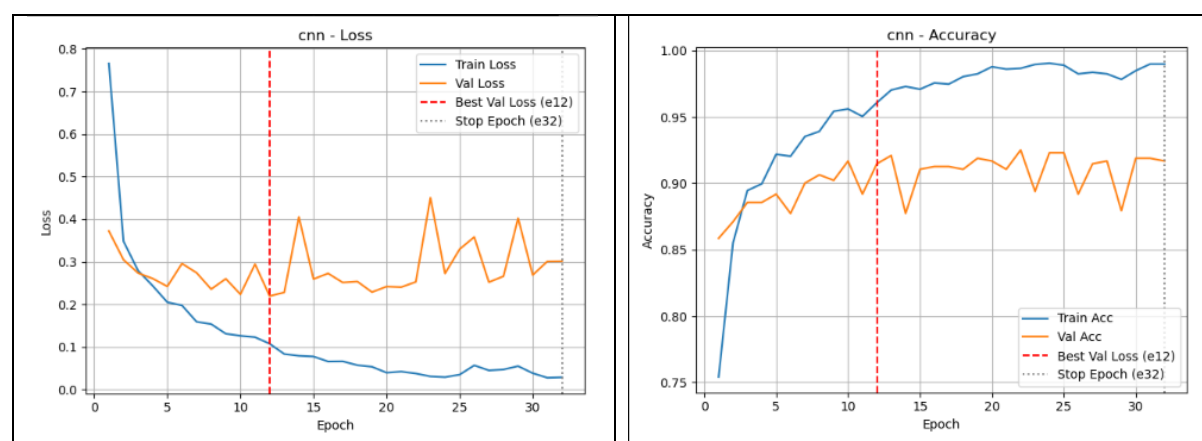
結論

目前測試下來參數為

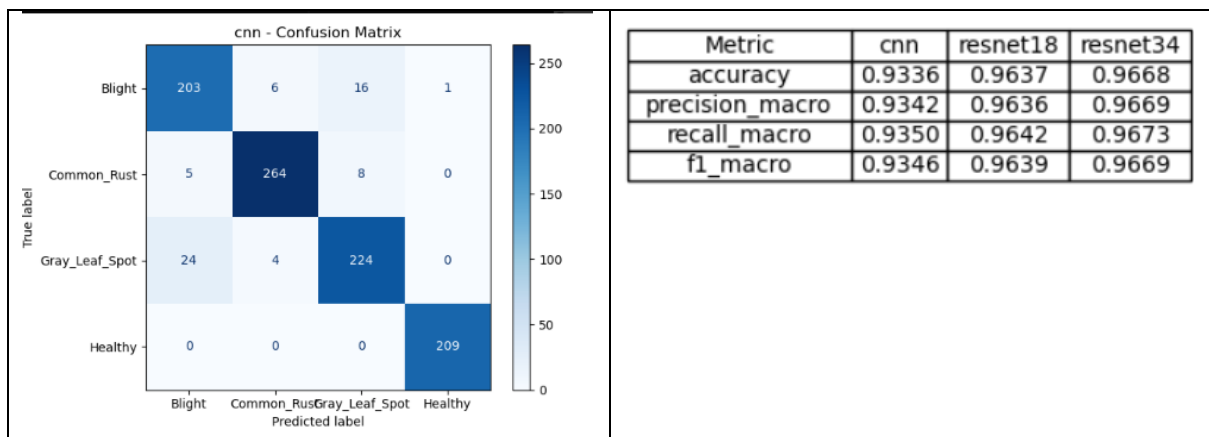
- `batch_size=64`,
- `IR=3e-4`
- 優化器: adam
- 權重衰減 `weight_decay=1e-4`

是目前最好的結果，

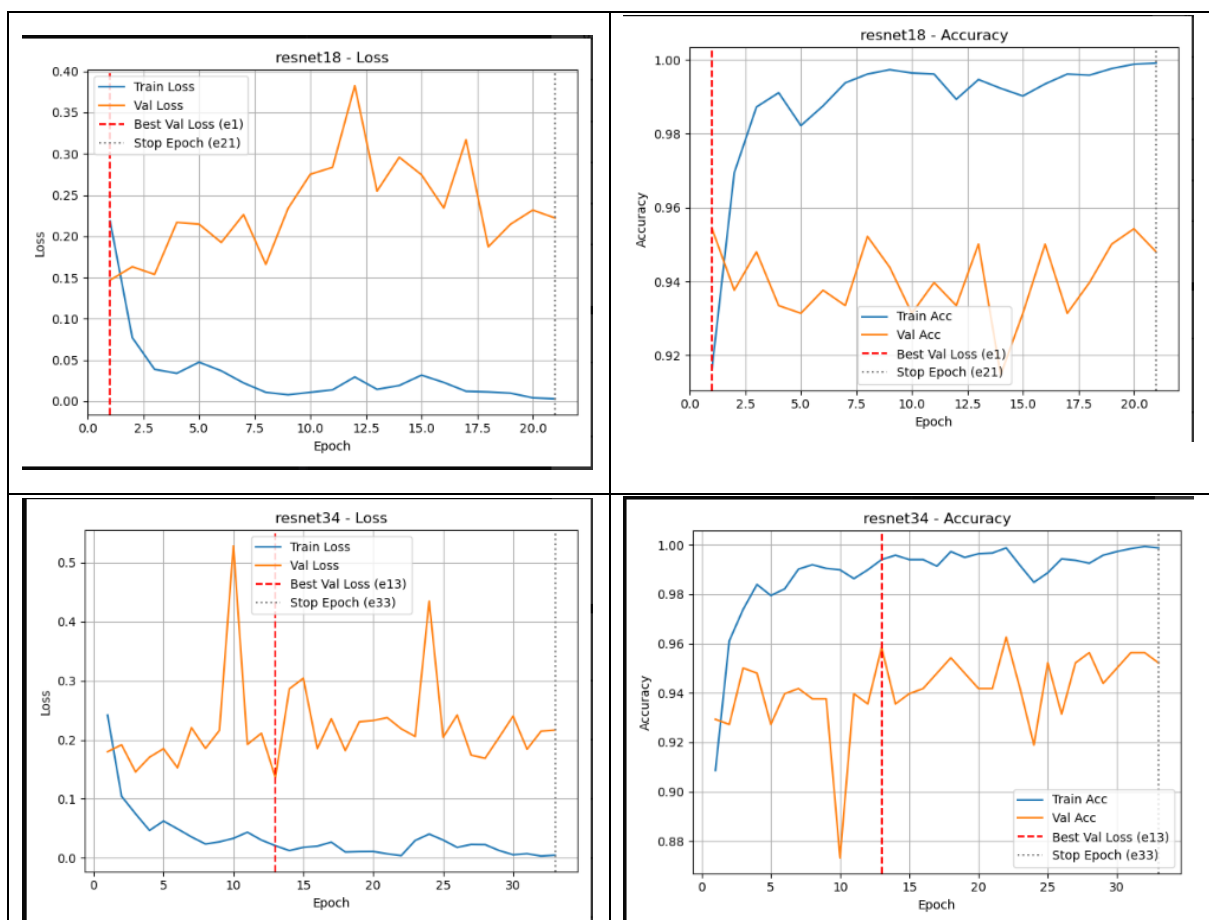
訓練驗證



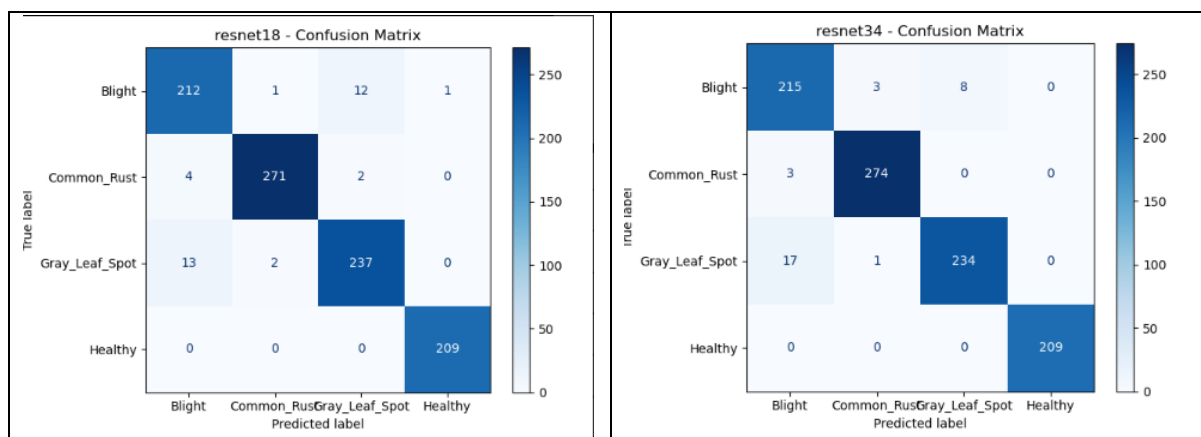
測試



ResNet 18 跟 34 結果是



測試



Metric	cnn	resnet18	resnet34
accuracy	0.9336	0.9637	0.9668
precision_macro	0.9342	0.9636	0.9669
recall_macro	0.9350	0.9642	0.9673
f1_macro	0.9346	0.9639	0.9669

最好的 CNN 目前準確 93.36%