جامعة حلوان
Helwan University
1975

كلية الحاسبات و الذكاء الصناعي

Plant Village Diseases Image Classification by Transfer learning

Student names:

1- Omar Ayman Assem
2- Omar Ahmed Ayad
3- Youssef Wael Attallah
4- Sara Omar Mohamed
5- Rehap Abdelghany Mohamed
6- Sama Haitham Ezzat
7- Sama Sameh Abdelaal

Contents

1. Introduction

Plant diseases reduce crop quality and yield, creating major challenges for farmers. Early detection is important, but traditional diagnosis requires experts and takes time. Deep learning provides an automated and accurate way to identify diseases from leaf images.

In this project, we use the **Plant Village dataset** to build a plant disease classification system. Four pretrained CNN architectures are evaluated: **VGG19 (Scartch Model) , ResNet, MobileNet, and Inception v1**. The images are resized, augmented, and balanced to improve model performance. The dataset is also split into training, validation, and testing sets.

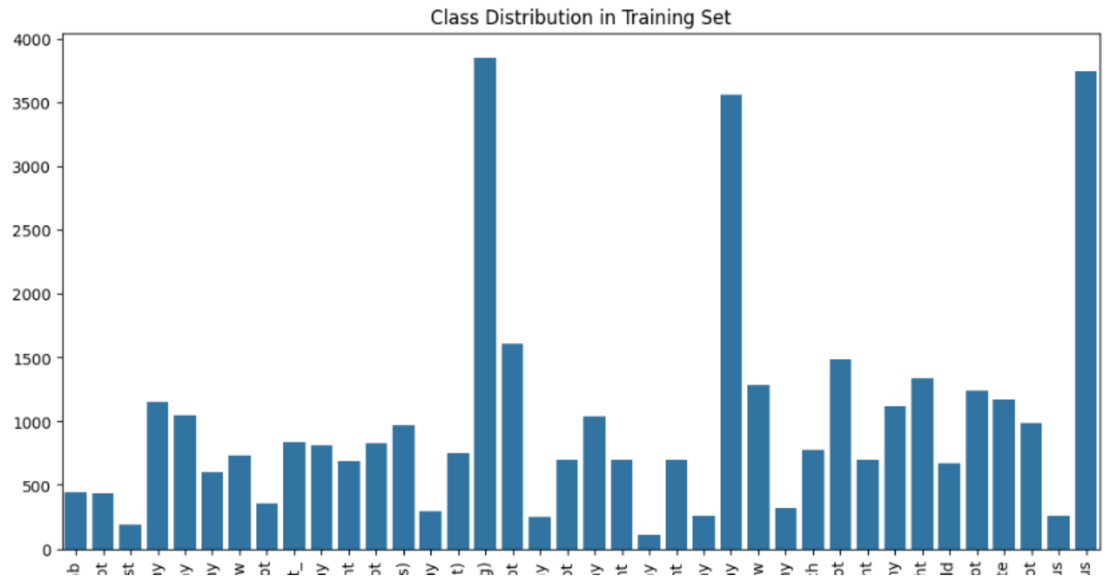
The goal of the project is to compare these architectures and determine which one provides the highest accuracy and best performance for detecting plant diseases. This system can help support modern agriculture by offering fast and reliable disease diagnosis.

2. About dataset

The dataset represents the diseases happened in plant that appeared in leaf images

Key characteristics of Dataset

1. Total number of images : ~ 54,300 images
2. Number of Classes : 38 Classes (Healthy / Not Healthy) of crop species
3. Crop species :14 different species(Apples, Blueberries, Cherries, Corn, Grape, Orange, Peach, Pepper, Potato, Raspberry, Soybean, Strawberry and Tomato)
4. Image type : RGB photos ; All photographed in same background but different lightening and leaf shape
5. Distribution : The Data is not perfectly balanced; some classes are representative (with max count images 3854) and some of them has few images



3. Data preprocessing

3.1. Decreasing Approach (Class Reduction)

To handle severe class imbalance in the PlantVillage dataset, a **decreasing approach** was applied.
Minor crops and under-represented disease categories were removed to prevent extremely skewed class distributions.

After removal, the dataset was reduced from **38 classes to 25–26 classes**, ensuring:

- Better class balance
- More meaningful representation
- Improved model performance and stability

This pruning step helps the models learn consistently without being dominated by rare classes that have only a few samples.

3.2. Image Resizing

All images were resized to **224 × 224 × 3**, which is the standard input size required by the CNN architectures used in this project (VGG19, ResNet, MobileNet, Inception v1).
This ensures:

- Uniformity across all inputs
- Compatibility with pretrained ImageNet-based models
- Reduced computational cost

3.3. Normalization / Rescaling

Pixel values were normalized using ImageNet statistics:

$$\text{mean} = [0.485, 0.456, 0.406], \text{std} = [0.229, 0.224, 0.225]$$

Normalization benefits include:

- Faster and more stable training
- Compatibility with pretrained weights
- Improved numerical performance

3.4. Train - Validation – Test Split

The dataset was split as follows:

- 70% Training
- 15% Validation
- 15% Testing

This split ensures:

- Sufficient data for training
- Reliable model evaluation during training
- Unbiased final testing performance

4. Handling Data imbalance

4.1. The problem

Even after removing under-represented classes, the dataset still contains **unequal samples per class**.
Some disease categories naturally have more images, which can bias the model toward majority classes.

4.2. Solution : Class-Balanced Sampling

a. Compute Class weights.

Which obtained from training labels and Classes with fewer samples receive **higher weights**, ensuring they appear more frequently during training.

b. Weighted Random Sampler

The sampler uses these weights to draw samples so that **all classes appear with similar frequency** in each epoch, without duplicating images.

This approach ensures:

- Balanced batches
- Better minority class learning
- Improved overall accuracy

4.3. Data Augmentation

Used to increase dataset variability and help the model generalize better:

- **RandomHorizontalFlip:** Randomly flips the image left-right to simulate natural variations in leaf orientation.
- **RandomRotation(10°):** Rotates images slightly to make the model robust against positional differences.
- **ToTensor():** Converts the image into a PyTorch tensor and scales pixel values from 0–255 to 0–1.

Outcome : A more balanced and diverse training set that prevents model bias toward majority classes and enhances overall performance.

4.4. Data Loader setup

The dataloaders were configured as follows:

Training Data loader:

- Uses **WeightedRandomSampler**
- Balanced batches
- No shuffle needed

Validation & Test Data Loaders:

- No Sampler

- Shuffle = False
- Reflect real distribution for proper evaluation

5. Model Architectures

5.1. VGG19

i. Architecture:

The custom VGG19 model implemented from scratch follows the core design principles of the original VGG19 paper—using **deep stacks of small 3×3 convolution filters**, increasing the number of channels gradually, and applying **max pooling** to reduce spatial dimensions.

This implementation includes:

1. **Feature Extraction Blocks (Block 1 → Block 5)**

   Each block consists of:

   - **3×3 convolution layers** with padding=1
   - **Batch Normalization** (an improvement over the original VGG19, which stabilizes training)
   - **ReLU activation**
   - A **MaxPool2d** layer to downsample by a factor of 2

   The number of filters grows as follows:

   - Block 1 → 64 channels
   - Block 2 → 128 channels
   - Block 3 → 256 channels
   - Block 4 → 512 channels
   - Block 5 → 512 channels

   Each block deepens the network and increases its capacity to learn complex features.

2. **Adaptive Average Pooling**

   After the 5 convolutional blocks, an **AdaptiveAvgPool2d(7×7)** layer ensures the output is always the same size, regardless of input resolution.

3. **Fully Connected Classifier**

   The classifier is a simplified version of VGG19's original FC layers:

   - Flatten
   - Linear → 1024
   - ReLU + Dropout
   - Linear → 512

- ReLU + Dropout
- Linear → num_classes (26)

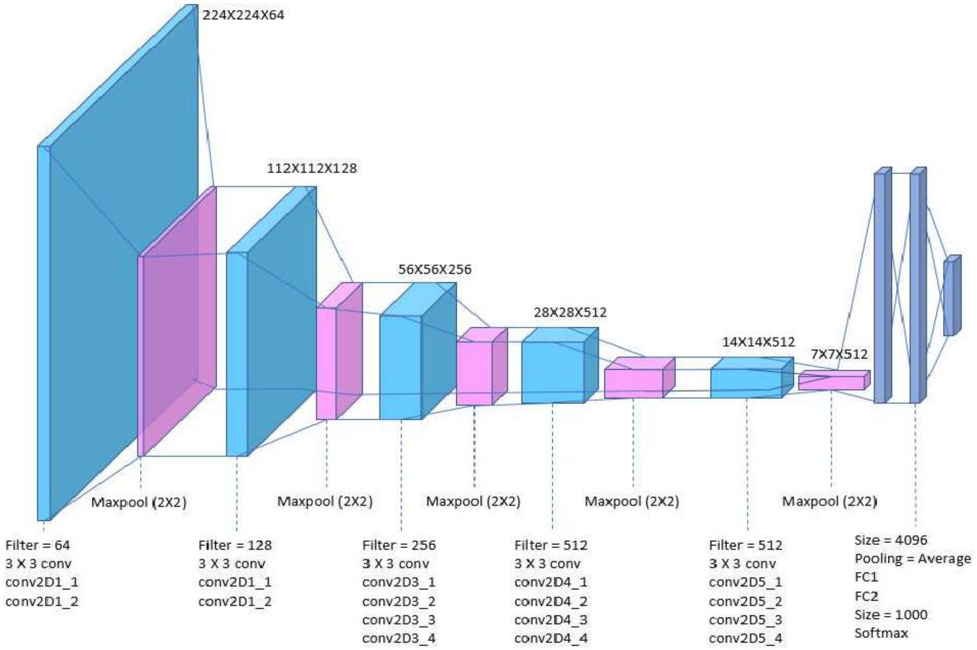
This reduces overfitting and improves generalization.

4. Forward Pass

The input passes through:

Block 1 → Block 2 → Block 3 → Block 4 → Block 5 → AvgPool → Classifier

ii. Diagram:



iii. Proper Referencing of the Original Papers:

Simonyan, K., & Zisserman, A. (2014).
“Very Deep Convolutional Networks for Large-Scale Image Recognition.”
https://arxiv.org/abs/1409.1556

5.2. ResNet50

i. Architecture:

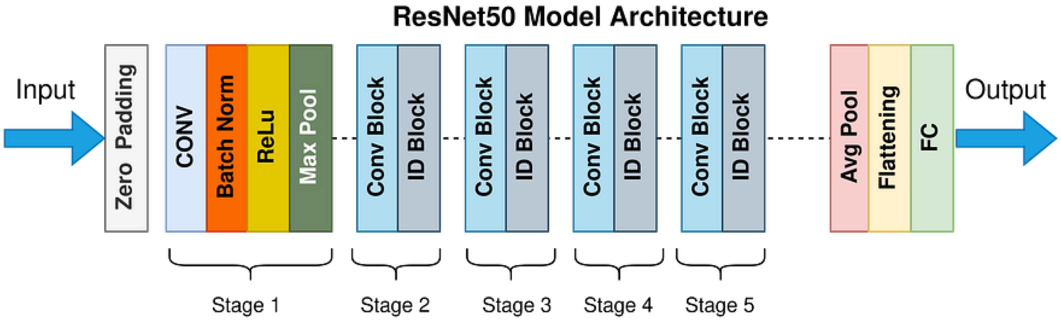
ResNet-50 is a 50-layer deep convolutional network built using **bottleneck residual blocks**.

The main feature of ResNet is the use of **skip connections** (identity shortcuts) that allow the gradient to flow easily through very deep networks, solving the vanishing gradient problem.

Key components:

1. Initial 7×7 Conv layer → BatchNorm → ReLU → 3×3 MaxPool
2. **4 stages** of bottleneck residual blocks:
   - Conv2_x → 3 blocks
   - Conv3_x → 4 blocks
   - Conv4_x → 6 blocks
   - Conv5_x → 3 blocks
3. Each bottleneck block contains:
   - 1×1 Conv (reduce channels)
   - 3×3 Conv
   - 1×1 Conv (expand channels)
   - Skip connection
4. Global Average Pooling
5. Fully connected (FC) classification layer

ii. Diagram:



iii. Proper Referencing of the Original Papers:

He, K., Zhang, X., Ren, S., & Sun, J. (2015).

"Deep Residual Learning for Image Recognition."

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

https://arxiv.org/abs/1512.03385

5.3. Inception V1

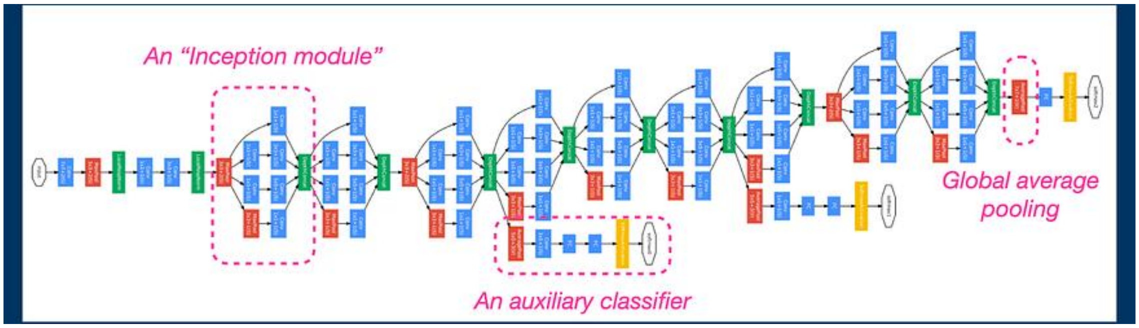
i. Architecture:

Inception v1 is a 22-layer deep CNN designed to improve both accuracy and computational efficiency.
Its main feature is the **Inception module**, which processes the input at multiple filter sizes simultaneously.

Key Components:

1. **Parallel convolutions inside each module:**
   - 1×1
   - 3×3
   - 5×5
   - MaxPool → 1×1
2. **Dimensionality reduction using 1×1 Conv**
3. **Stacking multiple Inception modules**
4. **Auxiliary classifiers at intermediate layers (for training)**
5. **Global Average Pooling**
6. **Fully connected classifier**

ii. Diagram:



iii. Proper Referencing of the Original Papers:

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., et al. (2014).
“Going Deeper with Convolutions.”
In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). https://arxiv.org/abs/1409.4842

5.4. MobileNet V2

i. Architecture:

MobileNet is built for mobile and real-time applications.
The key idea in **MobileNetV2** is the **Inverted Residual Block with Linear Bottleneck**, which improves feature representation while keeping computation low.
Each block is built from three main steps:

1. **Expansion Layer (1×1 convolution) → increases the number of channels**
2. **Depthwise convolution (3×3) → applies one filter per channel**
3. **Projection Layer (1×1 convolution, linear) → reduces the channels back to a compact form**

MobileNetV2 also uses **shortcut (skip) connections** between blocks when the input and output dimensions match.

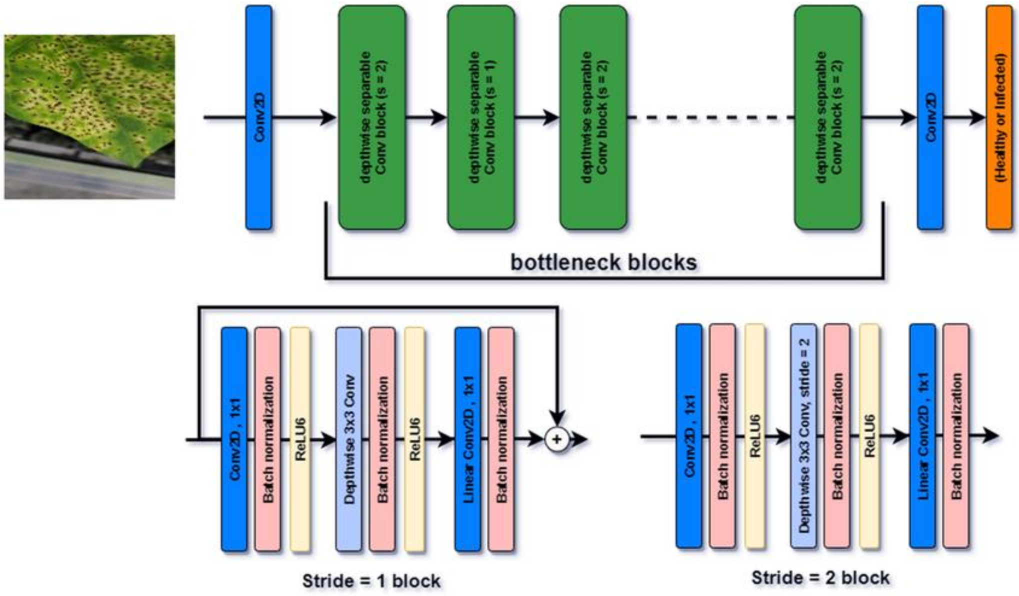
This results in:

- **~30–40% fewer parameters than standard CNNs**
- **Much faster inference** on mobile and embedded devices
- **Improved accuracy** compared to MobileNetV1

Architecture Flow:

1. **Initial 3×3 convolution**
2. **Repeated inverted residual blocks:**
    - 1×1 Expansion Conv
    - 3×3 Depthwise Conv
    - 1×1 Linear Projection Conv
3. **Global Average Pooling**
4. **Fully connected classifier**

ii. Diagram:



iii. Proper Referencing of the Original Papers:

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., et al. (2017).

“MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.”

https://arxiv.org/abs/1704.04861

6. Experimental Setup

6.1. VGG19

6.1.1. Environment Setup

i. GPU:

CUDA GPU

ii. Library:

PyTorch (torch, torch.nn, torch.optim, tqdm)

iii. Training Epochs: **15 Epochs**

iv. Batch Size: **16**

v. Optimizer:

Adam optimizer

vi. Loss Function:

CrossEntropyLoss

vii. Learning Rate: **0.0001 (1e-4)**

6.1.2. Training Method

i. Method:

Training from scratch using custom VGG19 architecture with supervised learning and mini-batch gradient descent.

ii. Frozen layers: **None (all layers are trainable)**

iii. Fine Tuning process:

No fine-tuning applied (model trained entirely from scratch).

Early stopping implemented with:

- Patience = 10
- Min delta = 0.001

6.2. ResNet

6.2.1. Environment Setup

i. GPU:

CUDA GPU

ii. Library:

PyTorch, Torchvision

iii. Training Epochs:

- Initial training: **10 epochs**
- Fine-tuning phase: **5 epochs**

Total: 15 epochs

iv. Batch Size: **32**

v. Optimizer:

Adam optimizer

vi. Loss Function:

CrossEntropyLoss with:

- Class weights (to handle class imbalance)

vii. Learning Rate:

- initial training: **1e-4 (0.0001)**
- Fine-tuning phase: **1e-5 (0.00001)**

6.2.2. Training Method

i. Method:

Transfer learning using **pretrained ResNet50** backbone with custom fully connected classifier.

ii. Frozen layers:

- Initially: **all backbone layers frozen**
- During fine-tuning: **All layers unfrozen (full network trainable)**

iii. Fine Tuning process:

- Train only the new fully connected head with the pretrained backbone frozen.
- Unfreeze all network layers.
- Continue training with a lower learning rate (1e-5).
- Save the best model based on validation accuracy.

6.3. Inception V1

6.3.1. Environment Setup

i. GPU:

CUDA GPU

ii. Library:

PyTorch, Torchvision

iii. Training Epochs:

- Initial training: **6 epochs**
- Fine-tuning phase: **12 epochs**

  Total: 18 epochs

iv. Batch Size: **32**

v. Optimizer:

- Initial training: **AdamW**
- Fine-tuning phase: **SGD with momentum**

vi. Loss Function:

CrossEntropyLoss with:

- Class weights (to handle class imbalance)

vii. Learning Rate:

- initial training: 1e-3 (0.001)
- Fine-tuning phase: 1e-4 (0.0001)

6.3.2. Training Method

i. Method:

Transfer learning using **pretrained GoogLeNet (Inception-V1)** with auxiliary classifiers and two-stage training strategy.

ii. Frozen layers:

- Initially: **all backbone layers frozen** , only :
    - Final fully connected layer (fc)
    - Auxiliary classifiers (aux1, aux2) were trainable
- During fine-tuning: all layers unfrozen

iii. Fine Tuning process:

- Train only the classification head and auxiliary classifiers for 6 epochs (AdamW, LR = 1e-3).
- Load best head weights.
- Unfreeze the entire network.
- Fine-tune all layers for 12 epochs using SGD with momentum and LR = 1e-4.
- Best models saved based on validation accuracy.

6.4. MobileNet

6.4.1. Environment Setup

i. GPU:

CUDA GPU

ii. Library:

PyTorch, Torchvision

iii. Training Epochs:

- Initial training: **10 epochs**
- Fine-tuning phase: **5 epochs**

  Total: 15 epochs

iv. Batch Size: **32**

v. Optimizer:

Adam optimizer

vi. Loss Function:

CrossEntropyLoss with:

- Class weights (to handle class imbalance)
- Label smoothing (0.1)

vii. Learning Rate:

- initial training: 1e-4 (0.0001)
- Fine-tuning phase: 1e-5 (0.00001)

6.4.2. Training Method

i. Method:

Transfer learning using **pretrained MobileNetV2** with:

- Frozen backbone
- Custom classifier head
- Data augmentation
- CosineAnnealingLR learning rate scheduler

ii. Frozen layers:

- Initially: **all backbone layers frozen**
- During fine-tuning: all layers frozen **except blocks "17" and "18"** in model.features

iii. Fine Tuning process:

- Train only the custom classifier with frozen backbone.
- Unfreeze last convolutional blocks (layers 17 and 18).
- Retrain full model using a smaller learning rate (1e-5).
- Save best model based on validation accuracy.

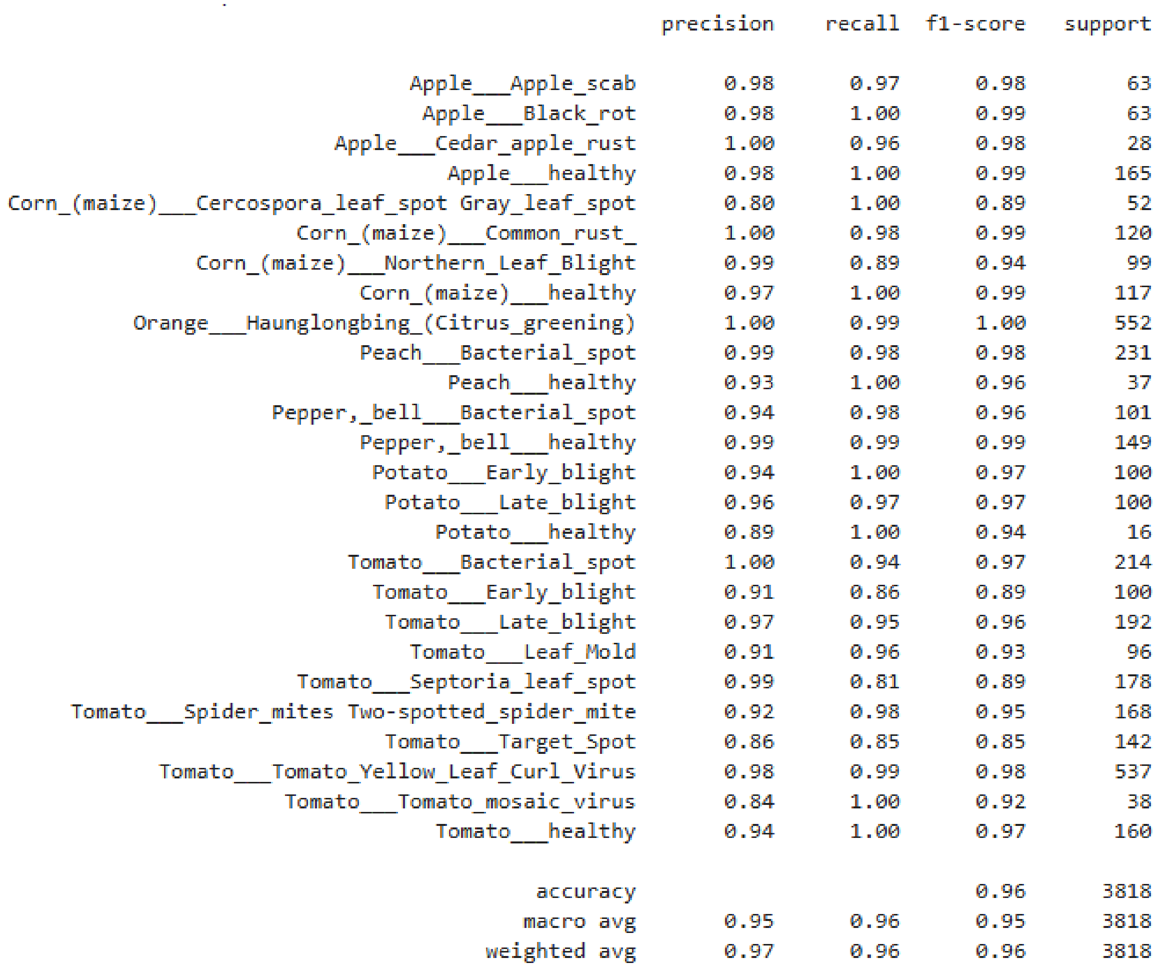
7. Results & Comparative Analysis

7.1. Performance Metrices

7.1.1. VGG 19

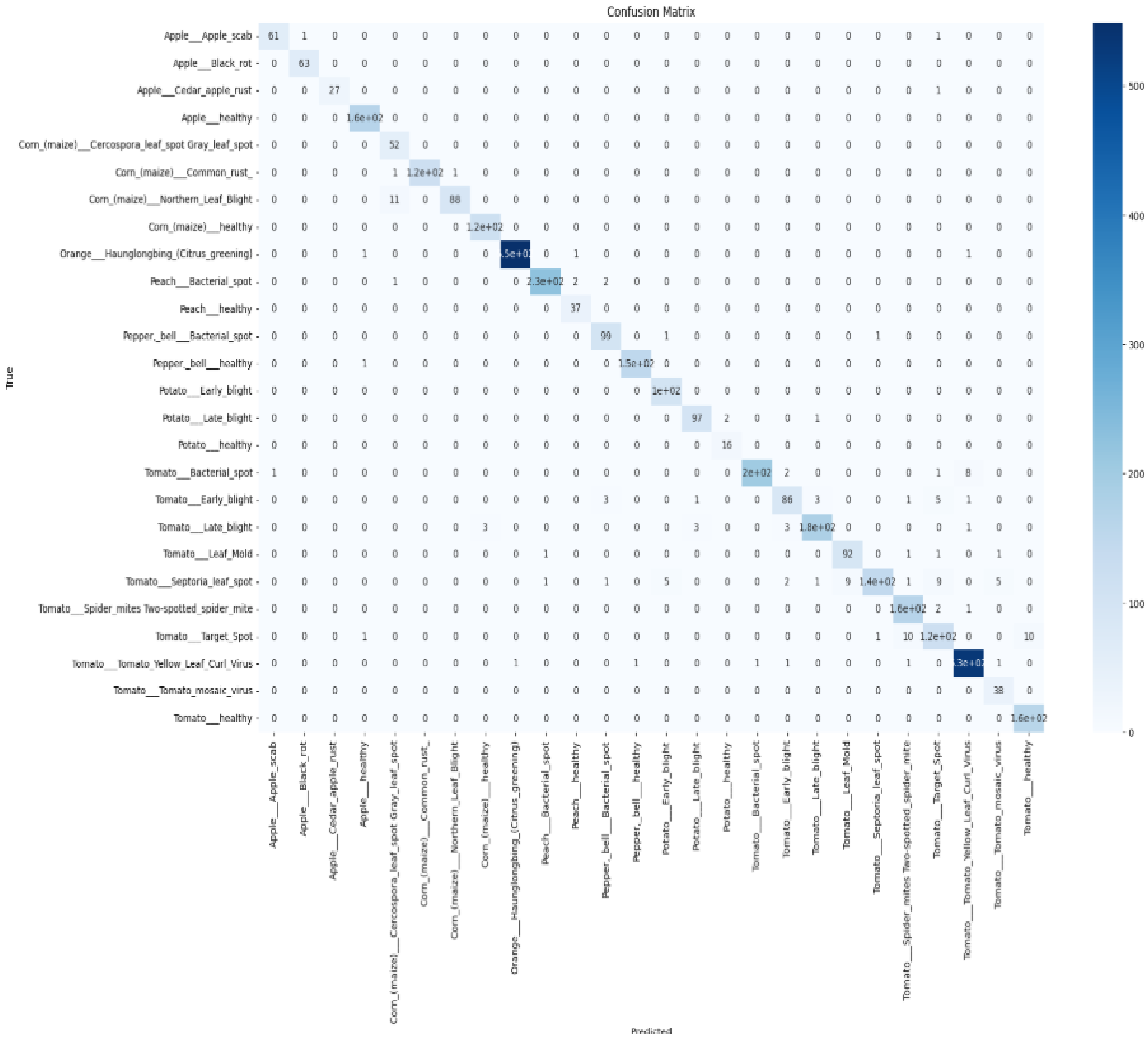
i. Accuracy: 0.964641173389209

ii. Classification report:



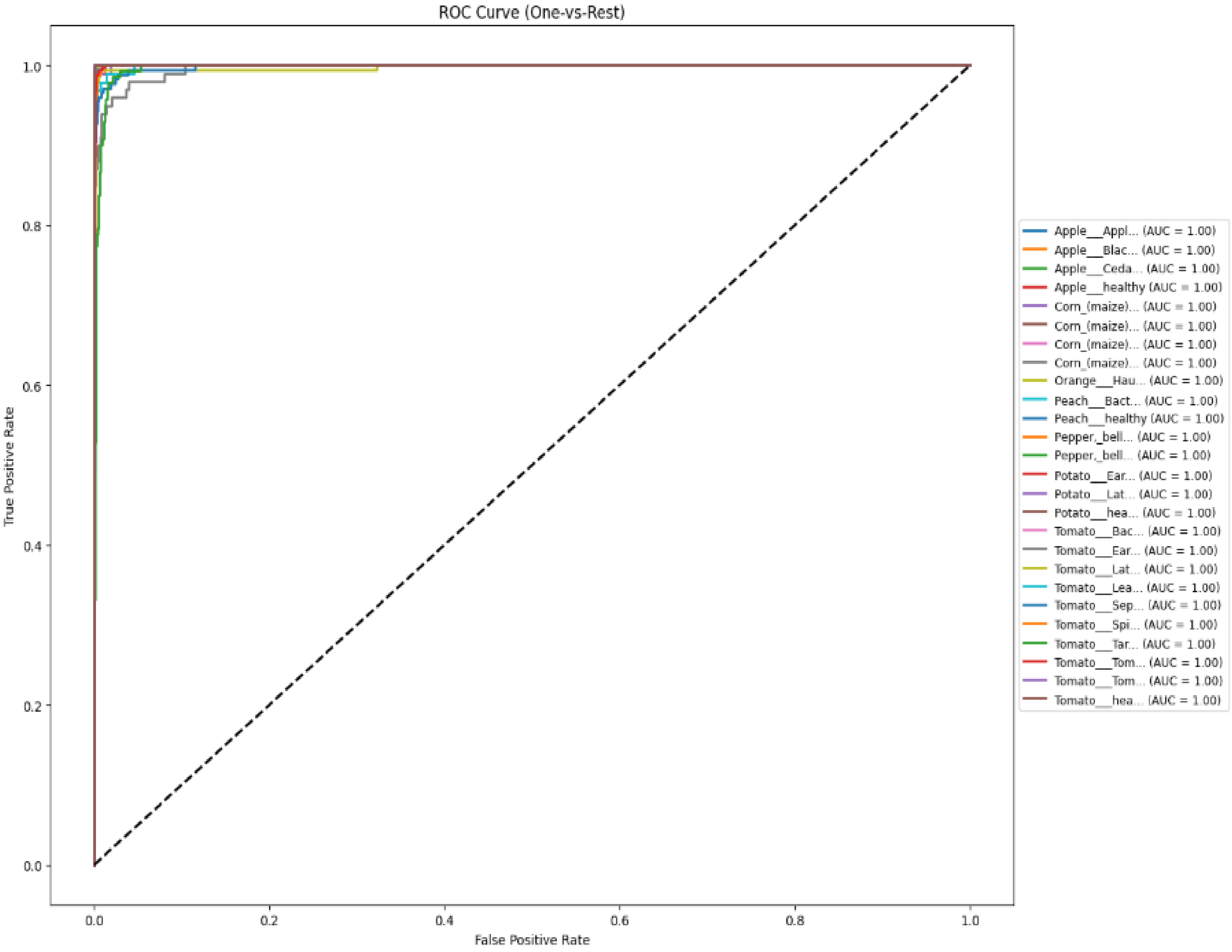
```
                                                    precision    recall  f1-score   support

                              Apple___Apple_scab       0.98      0.97      0.98        63
                               Apple___Black_rot       0.98      1.00      0.99        63
                        Apple___Cedar_apple_rust       1.00      0.96      0.98        28
                                 Apple___healthy       0.98      1.00      0.99       165
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot       0.80      1.00      0.89        52
                      Corn_(maize)___Common_rust_       1.00      0.98      0.99       120
              Corn_(maize)___Northern_Leaf_Blight       0.99      0.89      0.94        99
                           Corn_(maize)___healthy       0.97      1.00      0.99       117
        Orange___Haunglongbing_(Citrus_greening)       1.00      0.99      1.00       552
                           Peach___Bacterial_spot       0.99      0.98      0.98       231
                                  Peach___healthy       0.93      1.00      0.96        37
                    Pepper,_bell___Bacterial_spot       0.94      0.98      0.96       101
                           Pepper,_bell___healthy       0.99      0.99      0.99       149
                            Potato___Early_blight       0.94      1.00      0.97       100
                             Potato___Late_blight       0.96      0.97      0.97       100
                                 Potato___healthy       0.89      1.00      0.94        16
                          Tomato___Bacterial_spot       1.00      0.94      0.97       214
                            Tomato___Early_blight       0.91      0.86      0.89       100
                             Tomato___Late_blight       0.97      0.95      0.96       192
                               Tomato___Leaf_Mold       0.91      0.96      0.93        96
                      Tomato___Septoria_leaf_spot       0.99      0.81      0.89       178
    Tomato___Spider_mites Two-spotted_spider_mite       0.92      0.98      0.95       168
                             Tomato___Target_Spot       0.86      0.85      0.85       142
           Tomato___Tomato_Yellow_Leaf_Curl_Virus       0.98      0.99      0.98       537
                     Tomato___Tomato_mosaic_virus       0.84      1.00      0.92        38
                                 Tomato___healthy       0.94      1.00      0.97       160

                                         accuracy                           0.96      3818
                                        macro avg       0.95      0.96      0.95      3818
                                     weighted avg       0.97      0.96      0.96      3818
```

iii. Confusion Matrix:



	Apple___Apple_scab	Apple___Black_rot	Apple___Cedar_apple_rust	Apple___healthy	Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot	Corn_(maize)___Common_rust_	Corn_(maize)___Northern_Leaf_Blight	Corn_(maize)___healthy	Orange___Haunglongbing_(Citrus_greening)	Peach___Bacterial_spot	Peach___healthy	Pepper,_bell___Bacterial_spot	Pepper,_bell___healthy	Potato___Early_blight	Potato___Late_blight	Potato___healthy	Tomato___Bacterial_spot	Tomato___Early_blight	Tomato___Late_blight	Tomato___Leaf_Mold	Tomato___Septoria_leaf_spot	Tomato___Spider_mites Two-spotted_spider_mite	Tomato___Target_Spot	Tomato___Tomato_Yellow_Leaf_Curl_Virus	Tomato___Tomato_mosaic_virus	Tomato___healthy
Apple___Apple_scab	61	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Apple___Black_rot	0	63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple___Cedar_apple_rust	0	0	27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Apple___healthy	0	0	0	1.6e+02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot	0	0	0	0	52	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Corn_(maize)___Common_rust_	0	0	0	0	1	1.2e+02	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Corn_(maize)___Northern_Leaf_Blight	0	0	0	0	11	0	88	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Corn_(maize)___healthy	0	0	0	0	0	0	0	1.2e+02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Orange___Haunglongbing_(Citrus_greening)	0	0	0	1	0	0	0	0	.5e+02	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
Peach___Bacterial_spot	0	0	0	0	1	0	0	0	0	2.3e+02	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Peach___healthy	0	0	0	0	0	0	0	0	0	0	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pepper,_bell___Bacterial_spot	0	0	0	0	0	0	0	0	0	0	0	99	0	1	0	0	0	0	0	0	1	0	0	0	0	0
Pepper,_bell___healthy	0	0	0	1	0	0	0	0	0	0	0	0	1.5e+02	0	0	0	0	0	0	0	0	0	0	0	0	0
Potato___Early_blight	0	0	0	0	0	0	0	0	0	0	0	0	0	1e+02	0	0	0	0	0	0	0	0	0	0	0	0
Potato___Late_blight	0	0	0	0	0	0	0	0	0	0	0	0	0	0	97	2	0	0	1	0	0	0	0	0	0	0
Potato___healthy	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0
Tomato___Bacterial_spot	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2e+02	2	0	0	0	0	1	8	0	0
Tomato___Early_blight	0	0	0	0	0	0	0	0	0	0	0	3	0	0	1	0	0	86	3	0	0	1	5	1	0	0
Tomato___Late_blight	0	0	0	0	0	0	0	3	0	0	0	0	0	0	3	0	0	3	1.8e+02	0	0	0	0	1	0	0
Tomato___Leaf_Mold	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	92	0	1	1	0	1	0
Tomato___Septoria_leaf_spot	0	0	0	0	0	0	0	0	0	1	0	1	0	5	0	0	0	2	1	9	1.4e+02	1	9	0	5	0
Tomato___Spider_mites Two-spotted_spider_mite	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.6e+02	2	1	0	0
Tomato___Target_Spot	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	10	1.2e+02	0	0	10
Tomato___Tomato_Yellow_Leaf_Curl_Virus	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	1	0	0	0	1	0	.3e+02	1	0
Tomato___Tomato_mosaic_virus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38	0
Tomato___healthy	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1.6e+02

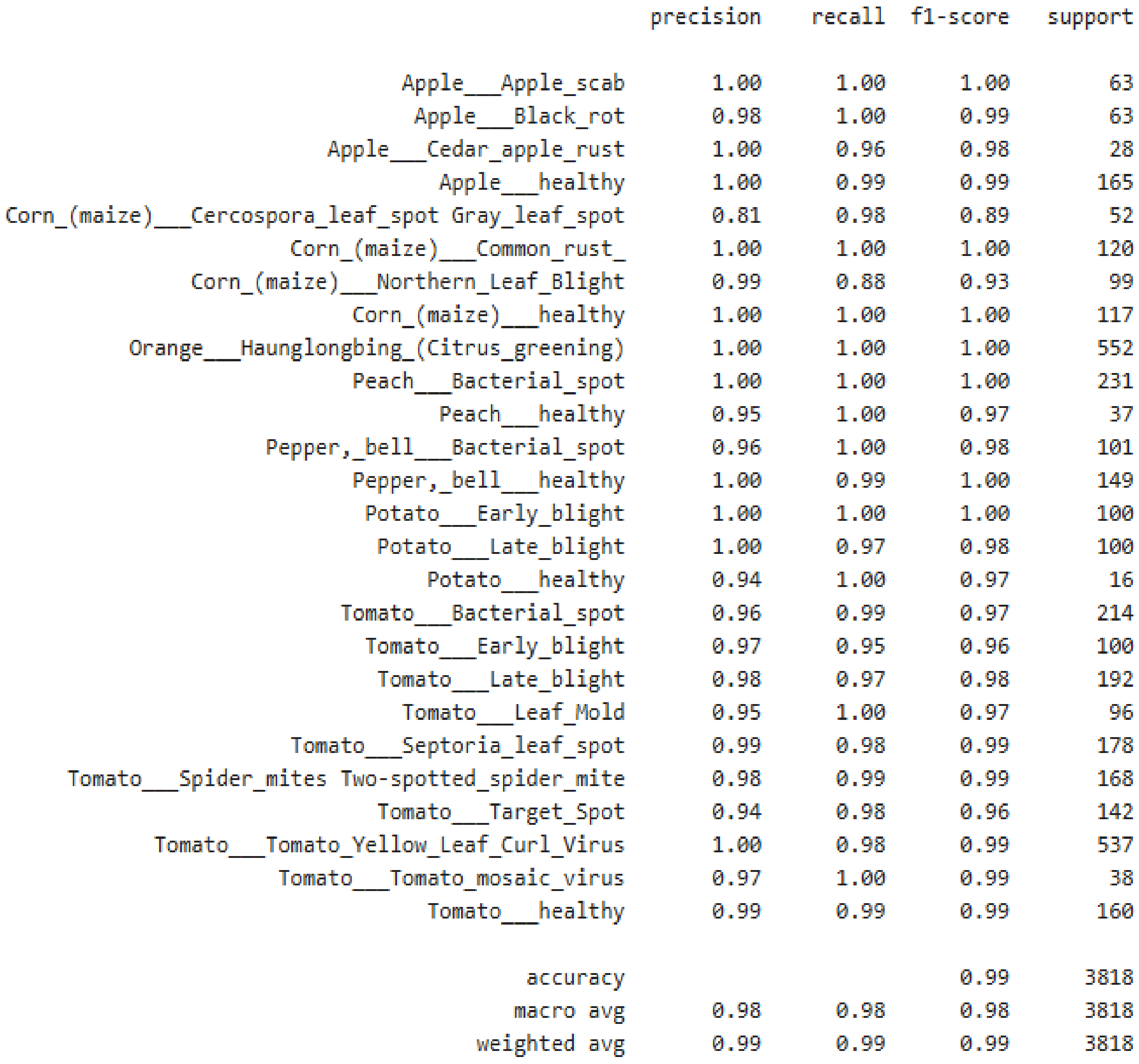
iv. ROC Curve:


ROC Curve (One-vs-Rest)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Apple___Appl... (AUC = 1.00)
Apple___Blac... (AUC = 1.00)
Apple___Ceda... (AUC = 1.00)
Apple___healthy (AUC = 1.00)
Corn_(maize)... (AUC = 1.00)
Corn_(maize)... (AUC = 1.00)
Corn_(maize)... (AUC = 1.00)
Corn_(maize)... (AUC = 1.00)
Orange___Hau... (AUC = 1.00)
Peach___Bact... (AUC = 1.00)
Peach___healthy (AUC = 1.00)
Pepper,_bell... (AUC = 1.00)
Pepper,_bell... (AUC = 1.00)
Potato___Ear... (AUC = 1.00)
Potato___Lat... (AUC = 1.00)
Potato___hea... (AUC = 1.00)
Tomato___Bac... (AUC = 1.00)
Tomato___Ear... (AUC = 1.00)
Tomato___Lat... (AUC = 1.00)
Tomato___Lea... (AUC = 1.00)
Tomato___Sep... (AUC = 1.00)
Tomato___Spi... (AUC = 1.00)
Tomato___Tar... (AUC = 1.00)
Tomato___Tom... (AUC = 1.00)
Tomato___Tom... (AUC = 1.00)
Tomato___hea... (AUC = 1.00)

7.1.2. ResNet 50

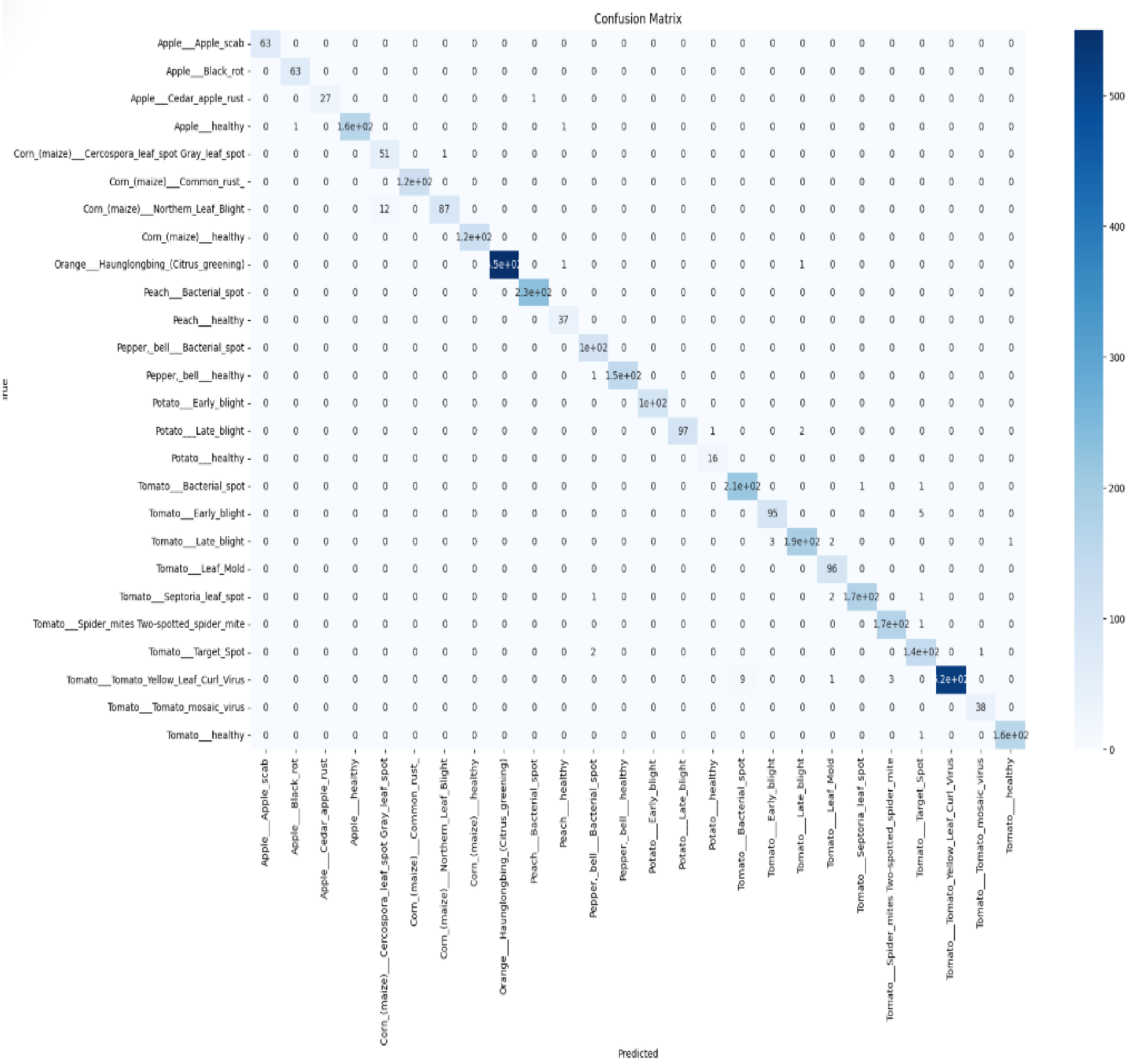
i. Accuracy: **0.9850707176532216**

ii. Classification report:

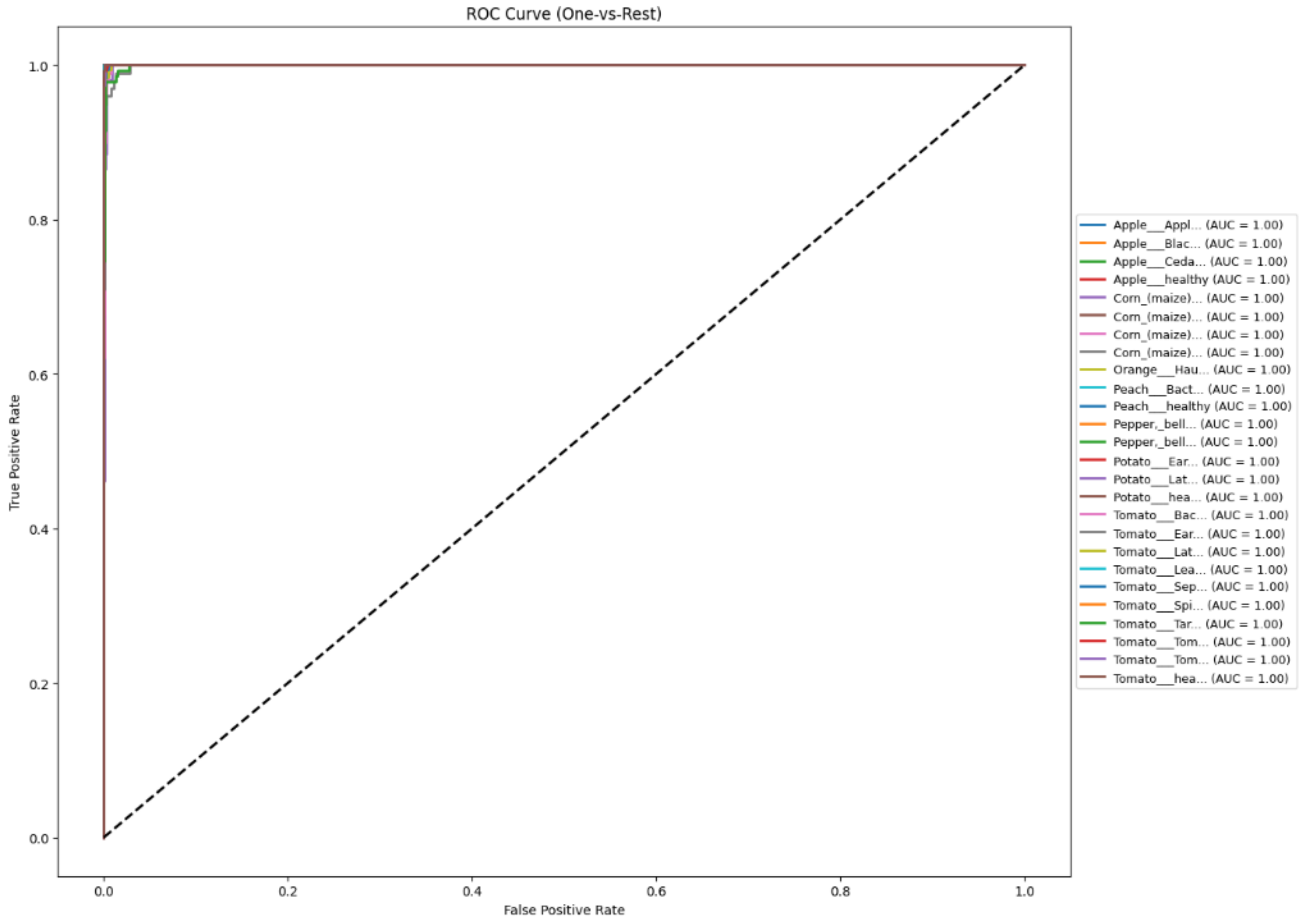


```
                                                    precision    recall  f1-score   support

                                Apple___Apple_scab       1.00      1.00      1.00        63
                                 Apple___Black_rot       0.98      1.00      0.99        63
                          Apple___Cedar_apple_rust       1.00      0.96      0.98        28
                                   Apple___healthy       1.00      0.99      0.99       165
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot       0.81      0.98      0.89        52
                       Corn_(maize)___Common_rust_       1.00      1.00      1.00       120
               Corn_(maize)___Northern_Leaf_Blight       0.99      0.88      0.93        99
                            Corn_(maize)___healthy       1.00      1.00      1.00       117
          Orange___Haunglongbing_(Citrus_greening)       1.00      1.00      1.00       552
                            Peach___Bacterial_spot       1.00      1.00      1.00       231
                                   Peach___healthy       0.95      1.00      0.97        37
                     Pepper,_bell___Bacterial_spot       0.96      1.00      0.98       101
                            Pepper,_bell___healthy       1.00      0.99      1.00       149
                             Potato___Early_blight       1.00      1.00      1.00       100
                              Potato___Late_blight       1.00      0.97      0.98       100
                                  Potato___healthy       0.94      1.00      0.97        16
                           Tomato___Bacterial_spot       0.96      0.99      0.97       214
                             Tomato___Early_blight       0.97      0.95      0.96       100
                              Tomato___Late_blight       0.98      0.97      0.98       192
                                Tomato___Leaf_Mold       0.95      1.00      0.97        96
                       Tomato___Septoria_leaf_spot       0.99      0.98      0.99       178
    Tomato___Spider_mites Two-spotted_spider_mite       0.98      0.99      0.99       168
                              Tomato___Target_Spot       0.94      0.98      0.96       142
            Tomato___Tomato_Yellow_Leaf_Curl_Virus       1.00      0.98      0.99       537
                      Tomato___Tomato_mosaic_virus       0.97      1.00      0.99        38
                                  Tomato___healthy       0.99      0.99      0.99       160

                                          accuracy                           0.99      3818
                                         macro avg       0.98      0.98      0.98      3818
                                      weighted avg       0.99      0.99      0.99      3818
```

iii. Confusion Matrix:



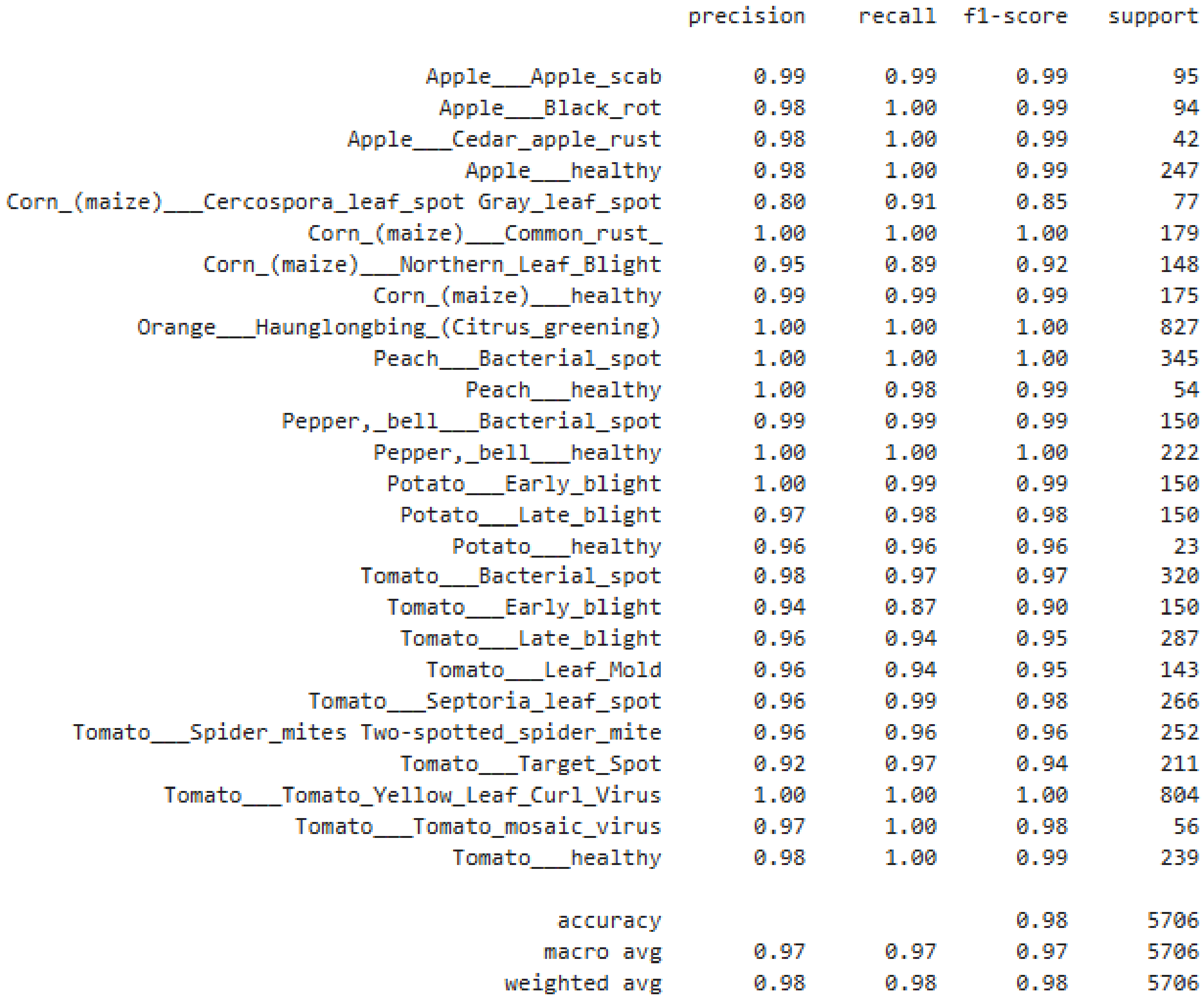
iv. ROC Curve:


ROC Curve (One-vs-Rest)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Apple___Appl... (AUC = 1.00)
Apple___Blac... (AUC = 1.00)
Apple___Ceda... (AUC = 1.00)
Apple___healthy (AUC = 1.00)
Corn_(maize)... (AUC = 1.00)
Corn_(maize)... (AUC = 1.00)
Corn_(maize)... (AUC = 1.00)
Corn_(maize)... (AUC = 1.00)
Orange___Hau... (AUC = 1.00)
Peach___Bact... (AUC = 1.00)
Peach___healthy (AUC = 1.00)
Pepper,_bell... (AUC = 1.00)
Pepper,_bell... (AUC = 1.00)
Potato___Ear... (AUC = 1.00)
Potato___Lat... (AUC = 1.00)
Potato___hea... (AUC = 1.00)
Tomato___Bac... (AUC = 1.00)
Tomato___Ear... (AUC = 1.00)
Tomato___Lat... (AUC = 1.00)
Tomato___Lea... (AUC = 1.00)
Tomato___Sep... (AUC = 1.00)
Tomato___Spi... (AUC = 1.00)
Tomato___Tar... (AUC = 1.00)
Tomato___Tom... (AUC = 1.00)
Tomato___Tom... (AUC = 1.00)
Tomato___hea... (AUC = 1.00)

7.1.3. Inception V1

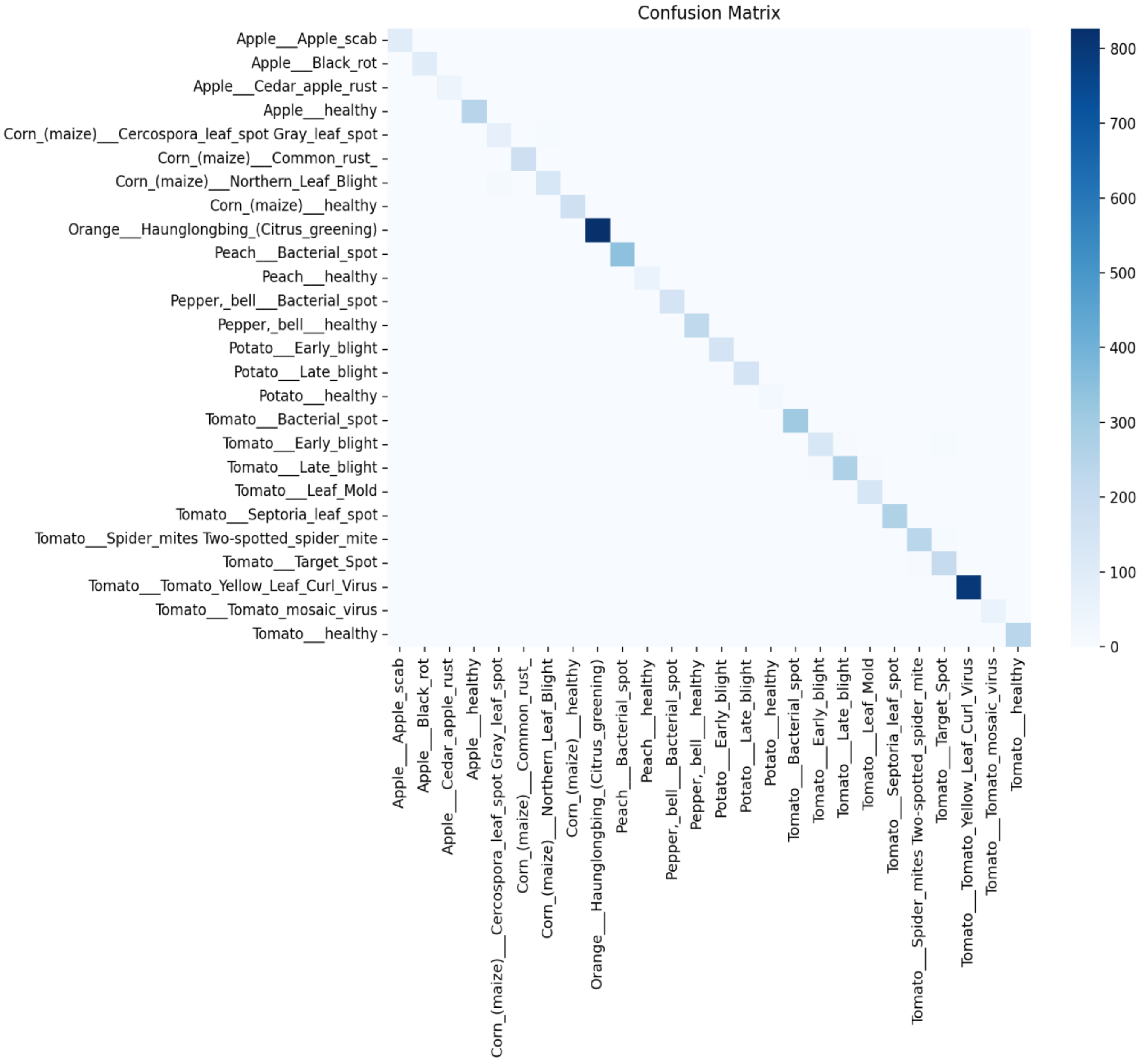
i. Accuracy: 0.9794952681388013

ii. Classification report:

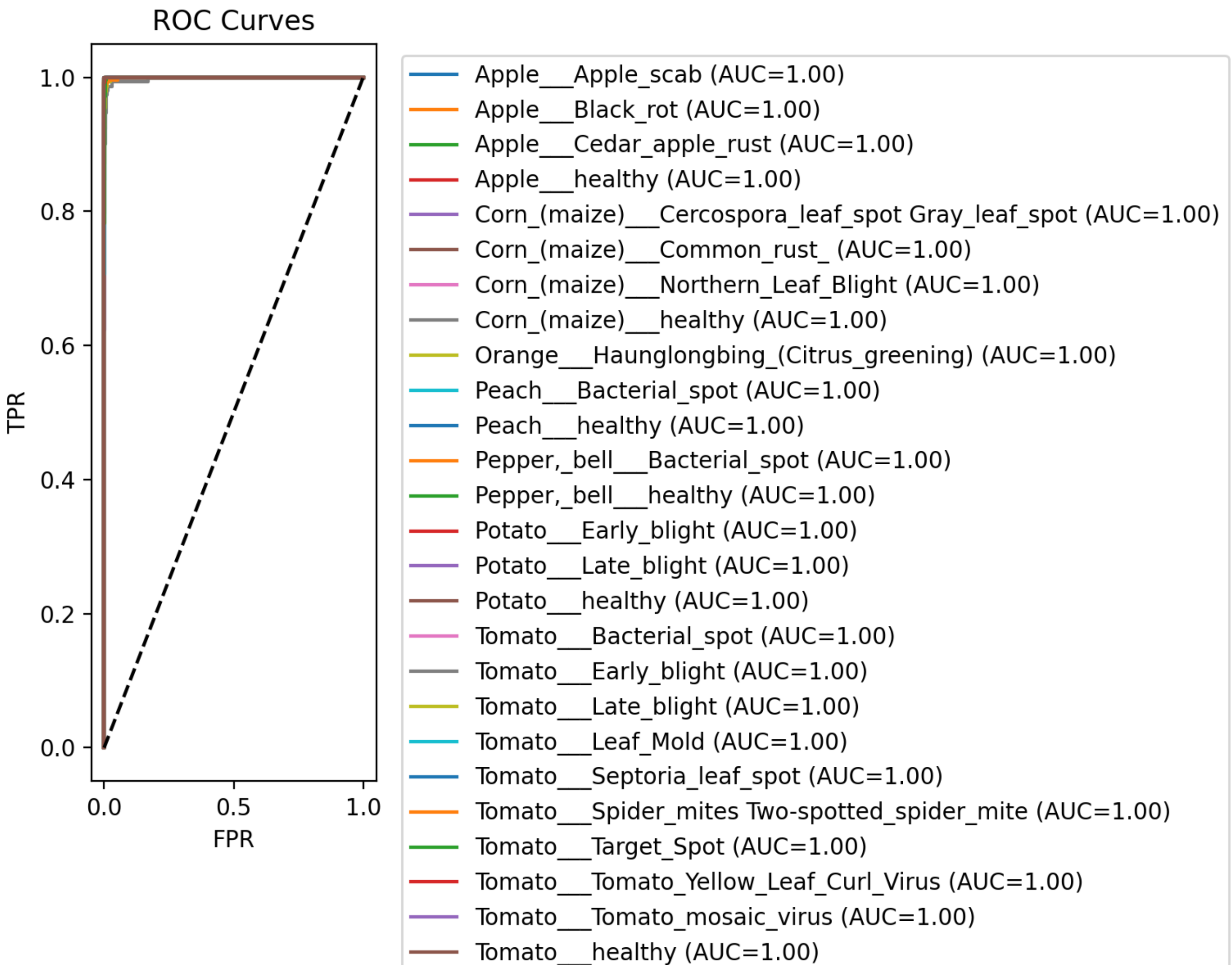


```
                                               precision    recall  f1-score   support

                           Apple___Apple_scab       0.99      0.99      0.99        95
                            Apple___Black_rot       0.98      1.00      0.99        94
                     Apple___Cedar_apple_rust       0.98      1.00      0.99        42
                              Apple___healthy       0.98      1.00      0.99       247
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot       0.80      0.91      0.85        77
                  Corn_(maize)___Common_rust_       1.00      1.00      1.00       179
          Corn_(maize)___Northern_Leaf_Blight       0.95      0.89      0.92       148
                       Corn_(maize)___healthy       0.99      0.99      0.99       175
    Orange___Haunglongbing_(Citrus_greening)       1.00      1.00      1.00       827
                      Peach___Bacterial_spot       1.00      1.00      1.00       345
                              Peach___healthy       1.00      0.98      0.99        54
               Pepper,_bell___Bacterial_spot       0.99      0.99      0.99       150
                      Pepper,_bell___healthy       1.00      1.00      1.00       222
                       Potato___Early_blight       1.00      0.99      0.99       150
                        Potato___Late_blight       0.97      0.98      0.98       150
                            Potato___healthy       0.96      0.96      0.96        23
                     Tomato___Bacterial_spot       0.98      0.97      0.97       320
                       Tomato___Early_blight       0.94      0.87      0.90       150
                        Tomato___Late_blight       0.96      0.94      0.95       287
                          Tomato___Leaf_Mold       0.96      0.94      0.95       143
                 Tomato___Septoria_leaf_spot       0.96      0.99      0.98       266
Tomato___Spider_mites Two-spotted_spider_mite       0.96      0.96      0.96       252
                        Tomato___Target_Spot       0.92      0.97      0.94       211
      Tomato___Tomato_Yellow_Leaf_Curl_Virus       1.00      1.00      1.00       804
                Tomato___Tomato_mosaic_virus       0.97      1.00      0.98        56
                            Tomato___healthy       0.98      1.00      0.99       239

                                    accuracy                           0.98      5706
                                   macro avg       0.97      0.97      0.97      5706
                                weighted avg       0.98      0.98      0.98      5706
```

iii. Confusion Matrix:



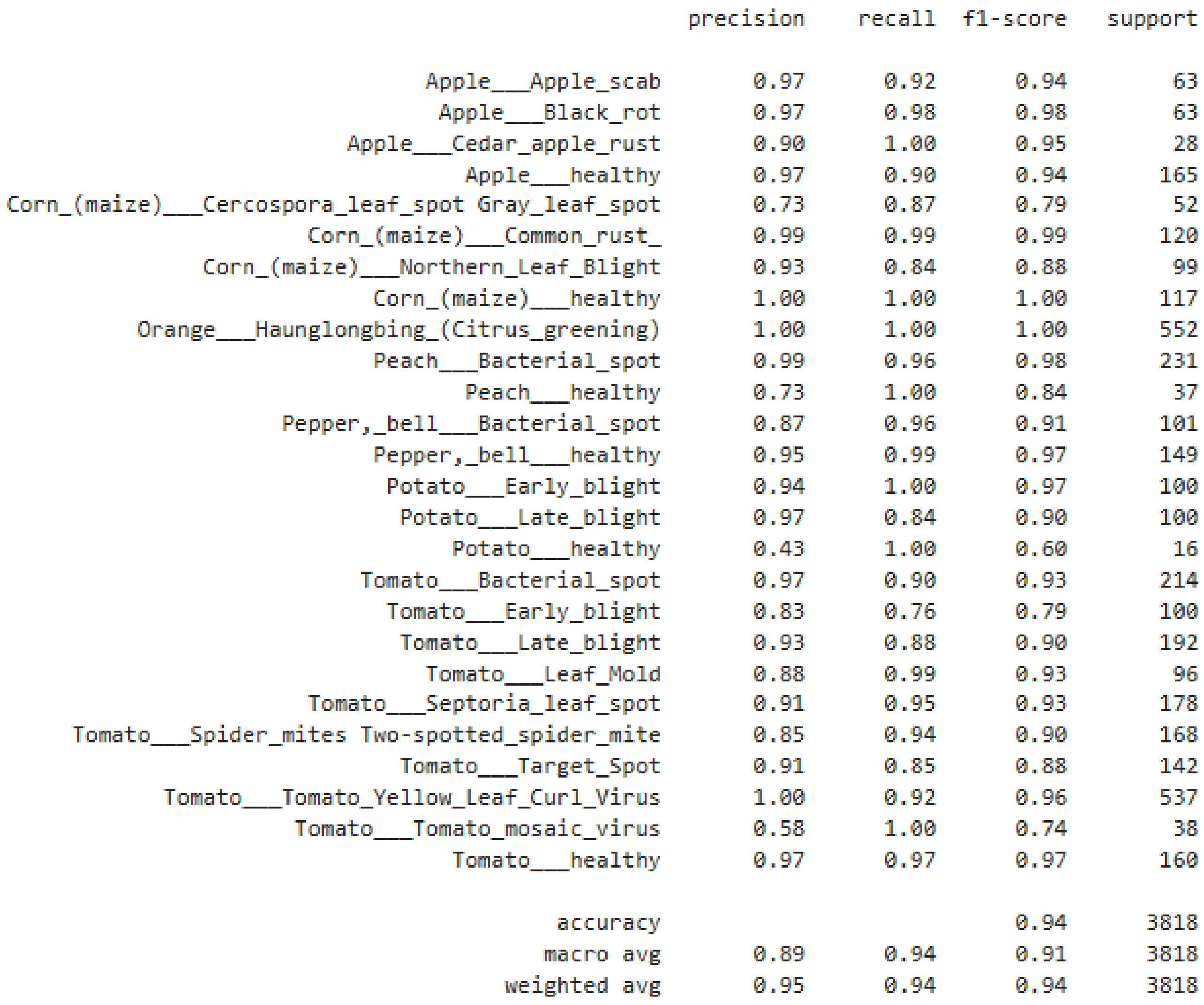
iv. ROC Curve:


ROC Curves
TPR
FPR
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.5
1.0
Apple___Apple_scab (AUC=1.00)
Apple___Black_rot (AUC=1.00)
Apple___Cedar_apple_rust (AUC=1.00)
Apple___healthy (AUC=1.00)
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot (AUC=1.00)
Corn_(maize)___Common_rust_ (AUC=1.00)
Corn_(maize)___Northern_Leaf_Blight (AUC=1.00)
Corn_(maize)___healthy (AUC=1.00)
Orange___Haunglongbing_(Citrus_greening) (AUC=1.00)
Peach___Bacterial_spot (AUC=1.00)
Peach___healthy (AUC=1.00)
Pepper,_bell___Bacterial_spot (AUC=1.00)
Pepper,_bell___healthy (AUC=1.00)
Potato___Early_blight (AUC=1.00)
Potato___Late_blight (AUC=1.00)
Potato___healthy (AUC=1.00)
Tomato___Bacterial_spot (AUC=1.00)
Tomato___Early_blight (AUC=1.00)
Tomato___Late_blight (AUC=1.00)
Tomato___Leaf_Mold (AUC=1.00)
Tomato___Septoria_leaf_spot (AUC=1.00)
Tomato___Spider_mites Two-spotted_spider_mite (AUC=1.00)
Tomato___Target_Spot (AUC=1.00)
Tomato___Tomato_Yellow_Leaf_Curl_Virus (AUC=1.00)
Tomato___Tomato_mosaic_virus (AUC=1.00)
Tomato___healthy (AUC=1.00)

7.1.4. MobileNet

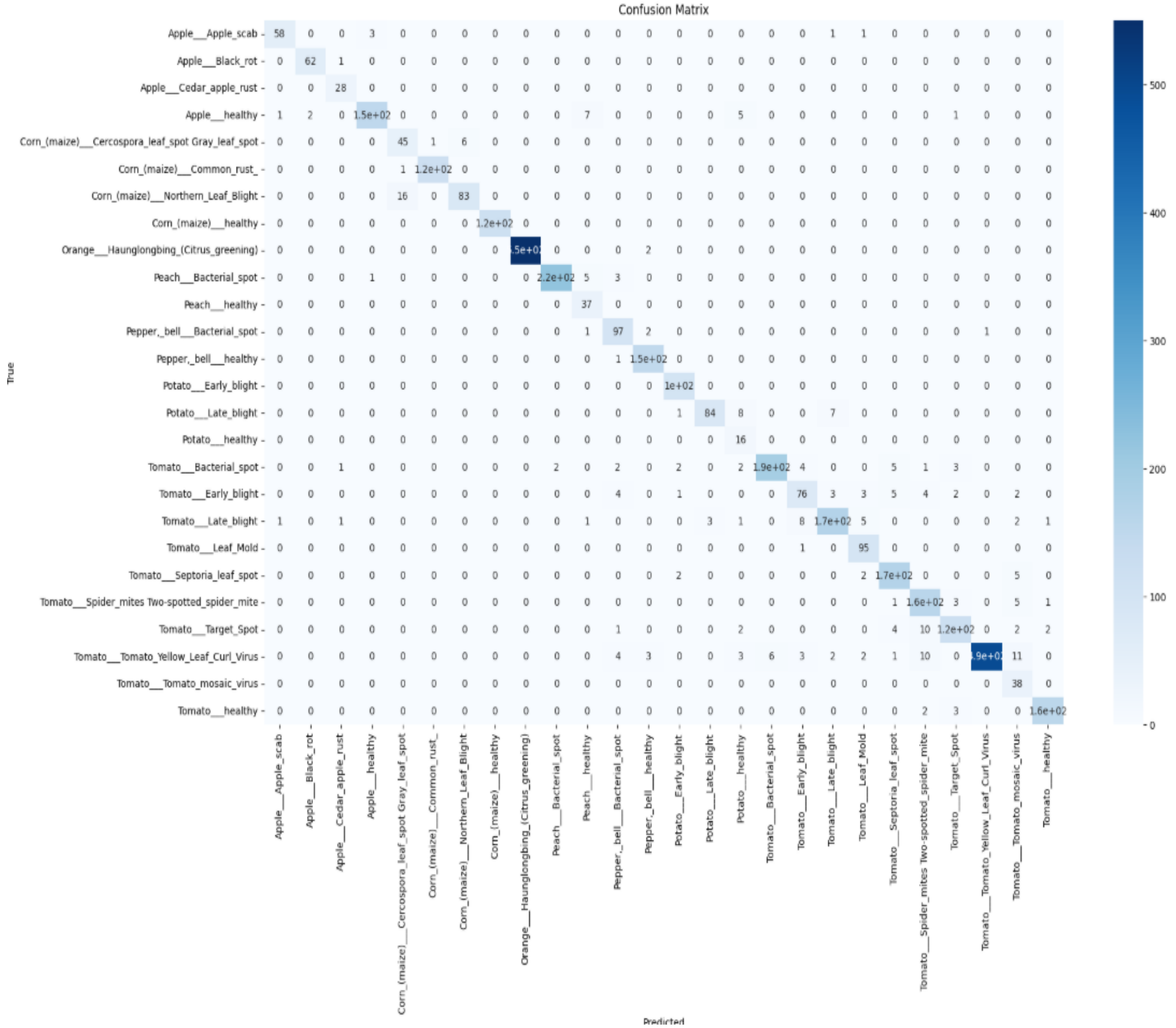
i. Accuracy: **0.9376636982713462**

ii. Classification report:



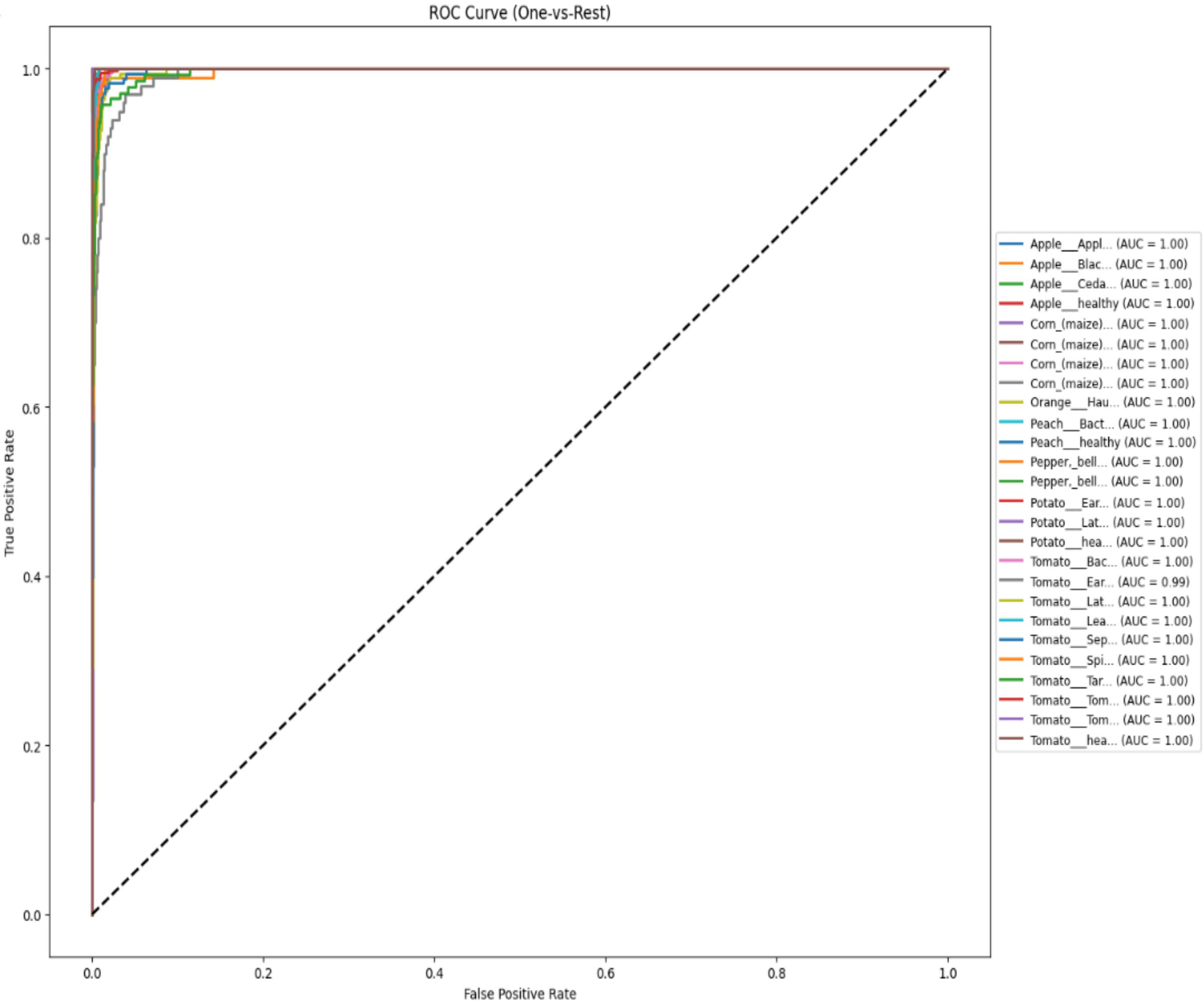
```
                                            precision    recall  f1-score   support

                     Apple___Apple_scab          0.97      0.92      0.94        63
                      Apple___Black_rot          0.97      0.98      0.98        63
               Apple___Cedar_apple_rust          0.90      1.00      0.95        28
                        Apple___healthy          0.97      0.90      0.94       165
 Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot          0.73      0.87      0.79        52
               Corn_(maize)___Common_rust_          0.99      0.99      0.99       120
       Corn_(maize)___Northern_Leaf_Blight          0.93      0.84      0.88        99
                 Corn_(maize)___healthy          1.00      1.00      1.00       117
    Orange___Haunglongbing_(Citrus_greening)          1.00      1.00      1.00       552
                 Peach___Bacterial_spot          0.99      0.96      0.98       231
                        Peach___healthy          0.73      1.00      0.84        37
          Pepper,_bell___Bacterial_spot          0.87      0.96      0.91       101
                 Pepper,_bell___healthy          0.95      0.99      0.97       149
                  Potato___Early_blight          0.94      1.00      0.97       100
                   Potato___Late_blight          0.97      0.84      0.90       100
                       Potato___healthy          0.43      1.00      0.60        16
                Tomato___Bacterial_spot          0.97      0.90      0.93       214
                  Tomato___Early_blight          0.83      0.76      0.79       100
                   Tomato___Late_blight          0.93      0.88      0.90       192
                     Tomato___Leaf_Mold          0.88      0.99      0.93        96
            Tomato___Septoria_leaf_spot          0.91      0.95      0.93       178
  Tomato___Spider_mites Two-spotted_spider_mite          0.85      0.94      0.90       168
                   Tomato___Target_Spot          0.91      0.85      0.88       142
         Tomato___Tomato_Yellow_Leaf_Curl_Virus          1.00      0.92      0.96       537
            Tomato___Tomato_mosaic_virus          0.58      1.00      0.74        38
                       Tomato___healthy          0.97      0.97      0.97       160

                               accuracy                              0.94      3818
                              macro avg          0.89      0.94      0.91      3818
                           weighted avg          0.95      0.94      0.94      3818
```

iii. Confusion Matrix:



	Apple___Apple_scab	Apple___Black_rot	Apple___Cedar_apple_rust	Apple___healthy	Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot	Corn_(maize)___Common_rust_	Corn_(maize)___Northern_Leaf_Blight	Corn_(maize)___healthy	Orange___Haunglongbing_(Citrus_greening)	Peach___Bacterial_spot	Peach___healthy	Pepper,_bell___Bacterial_spot	Pepper,_bell___healthy	Potato___Early_blight	Potato___Late_blight	Potato___healthy	Tomato___Bacterial_spot	Tomato___Early_blight	Tomato___Late_blight	Tomato___Leaf_Mold	Tomato___Septoria_leaf_spot	Tomato___Spider_mites Two-spotted_spider_mite	Tomato___Target_Spot	Tomato___Tomato_Yellow_Leaf_Curl_Virus	Tomato___Tomato_mosaic_virus	Tomato___healthy
Apple___Apple_scab	58	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
Apple___Black_rot	0	62	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple___Cedar_apple_rust	0	0	28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Apple___healthy	1	2	0	1.5e+02	0	0	0	0	0	0	7	0	0	0	0	5	0	0	0	0	0	0	1	0	0	0
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot	0	0	0	0	45	1	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Corn_(maize)___Common_rust_	0	0	0	0	1	1.2e+02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Corn_(maize)___Northern_Leaf_Blight	0	0	0	0	16	0	83	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Corn_(maize)___healthy	0	0	0	0	0	0	0	1.2e+02	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Orange___Haunglongbing_(Citrus_greening)	0	0	0	0	0	0	0	0	.5e+0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
Peach___Bacterial_spot	0	0	0	1	0	0	0	0	0	2.2e+02	5	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Peach___healthy	0	0	0	0	0	0	0	0	0	0	37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Pepper,_bell___Bacterial_spot	0	0	0	0	0	0	0	0	0	0	1	97	2	0	0	0	0	0	0	0	0	0	0	1	0	0
Pepper,_bell___healthy	0	0	0	0	0	0	0	0	0	0	0	1	1.5e+02	0	0	0	0	0	0	0	0	0	0	0	0	0
Potato___Early_blight	0	0	0	0	0	0	0	0	0	0	0	0	0	1e+02	0	0	0	0	0	0	0	0	0	0	0	0
Potato___Late_blight	0	0	0	0	0	0	0	0	0	0	0	0	0	1	84	8	0	0	7	0	0	0	0	0	0	0
Potato___healthy	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	16	0	0	0	0	0	0	0	0	0	0
Tomato___Bacterial_spot	0	0	1	0	0	0	0	0	0	2	0	2	0	2	0	2	1.9e+02	4	0	0	5	1	3	0	0	0
Tomato___Early_blight	0	0	0	0	0	0	0	0	0	0	0	4	0	1	0	0	0	76	3	3	5	4	2	0	2	0
Tomato___Late_blight	1	0	1	0	0	0	0	0	0	0	1	0	0	0	3	1	0	8	1.7e+02	5	0	0	0	0	2	1
Tomato___Leaf_Mold	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	95	0	0	0	0	0	0
Tomato___Septoria_leaf_spot	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	2	1.7e+02	0	0	0	5	0
Tomato___Spider_mites Two-spotted_spider_mite	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.6e+02	3	0	5	1
Tomato___Target_Spot	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	2	0	0	0	0	4	10	1.2e+02	0	2	2
Tomato___Tomato_Yellow_Leaf_Curl_Virus	0	0	0	0	0	0	0	0	0	0	0	4	3	0	0	3	6	3	2	2	1	10	0	.9e+0	11	0
Tomato___Tomato_mosaic_virus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	38	0
Tomato___healthy	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	3	0	0	1.6e+02

iv. ROC Curve:


ROC Curve (One-vs-Rest)
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0
Apple___Appl... (AUC = 1.00)
Apple___Blac... (AUC = 1.00)
Apple___Ceda... (AUC = 1.00)
Apple___healthy (AUC = 1.00)
Corn_(maize)... (AUC = 1.00)
Corn_(maize)... (AUC = 1.00)
Corn_(maize)... (AUC = 1.00)
Corn_(maize)... (AUC = 1.00)
Orange___Hau... (AUC = 1.00)
Peach___Bact... (AUC = 1.00)
Peach___healthy (AUC = 1.00)
Pepper,_bell... (AUC = 1.00)
Pepper,_bell... (AUC = 1.00)
Potato___Ear... (AUC = 1.00)
Potato___Lat... (AUC = 1.00)
Potato___hea... (AUC = 1.00)
Tomato___Bac... (AUC = 1.00)
Tomato___Ear... (AUC = 0.99)
Tomato___Lat... (AUC = 1.00)
Tomato___Lea... (AUC = 1.00)
Tomato___Sep... (AUC = 1.00)
Tomato___Spi... (AUC = 1.00)
Tomato___Tar... (AUC = 1.00)
Tomato___Tom... (AUC = 1.00)
Tomato___Tom... (AUC = 1.00)
Tomato___hea... (AUC = 1.00)

7.2. Comparison Table

Model	Train Acc	Val Acc	Test Acc	Parameters (Approx.)	Inference Speed
VGG19	0.9512	0.96	0.9646	~144 million	Slow
ResNet50	0.9766	0.9829	0.985	~25 million	Medium
Inception v1	0.98	0.9794	0.9794	~6.8 million	Medium
MobileNetV2	0.9205	0.9370	0.9377	~3.4 million	Fast

7.3. Analysis

7.3.1. VGG 19

i. Performance

- Train Accuracy: 0.9512
- Validation Accuracy: 0.9600
- Test Accuracy: 0.9646

  VGG19 performed reasonably well but was surpassed by ResNet50 and Inception v1 in overall generalization.

ii. Why it performed this way

- VGG19 uses very deep sequential convolutional layers (19 layers) without shortcut connections.
- It extracts strong spatial features but suffers from:
    - Large parameter count
    - Higher risk of overfitting
    - Slower optimization and poorer gradient flow

iii. Model Size & Speed

- ~144M parameters → the largest model in the comparison.
- Inference speed: Slow, making it less suitable for real-time systems.

iv. Overfitting / Underfitting Behavior

- Slight overfitting: Train acc (0.9512) < Val/Test acc
  This means augmentation helped VGG19 generalize better, but the heavy architecture still tends to memorize training examples.

v. Impact of Augmentation

- Data augmentation helped reduce overfitting.
- Improved validation accuracy by increasing data diversity, which is crucial for such a heavy model.

7.3.2. ResNet

i. Performance

- Train Accuracy: 0.9766
- Validation Accuracy: 0.9829
- Test Accuracy: 0.9850
  ResNet50 achieved the **highest performance** across all models.

ii. Why it performed this way

- Its residual connections solve the vanishing gradient problem, allowing deeper and more effective training.
- Better feature reuse → stronger generalization
- Balanced capacity makes it ideal for medium-sized image datasets..

iii. Model Size & Speed

- **~25M parameters** → significantly smaller than VGG19 but larger than MobileNet/Inception.
- **Inference speed: Medium** — good for both training and deployment.

iv. Overfitting / Underfitting Behavior

- **Minimal overfitting**: train/validation/test accuracies are very close.
- **The model generalizes extremely well due to:**
  - Residual blocks
  - Batch normalization
  - Strong pretraining weights

v. Impact of Augmentation

- Augmentation slightly improved generalization, but ResNet50 performs strongly even without heavy augmentation thanks to its architecture.

7.3.3. Inception v1

i. Performance

- Train Accuracy: 0.98
- Validation Accuracy: 0.9794
- Test Accuracy: 0.9794
  Inception v1 performed very close to ResNet50, showing excellent generalization

ii. Why it performed this way

- Uses multi-scale convolutions (1×1, 3×3, 5×5) inside each Inception module.
- Extracts fine- and coarse-level features simultaneously.
- Efficient architecture for complex pattern variation

iii. Model Size & Speed

- **~6.8M parameters —** smaller than ResNet, much smaller than VGG19.
- **Inference speed: Medium,** faster than ResNet but not as fast as MobileNet.

iv. Overfitting / Underfitting Behavior

- No significant overfitting: train ≈ val ≈ test.
- Auxiliary classifiers improve gradient propagation and stabilize training.

v. Impact of Augmentation

- Augmentation boosted robustness to rotation/scale changes, which aligns well with Inception's multi-scale processing..

7.3.4. MobileNet

i. Performance

- Train Accuracy: 0.9205
- Validation Accuracy: 0. 9370
- Test Accuracy: 0. 0.9377

  MobileNetV2 showed good performance but noticeably lower than the heavier models — expected for a mobile-optimized architecture.

ii. Why it performed this way

- Uses **depthwise separable convolutions and inverted residual blocks**, focusing on efficiency rather than maximum accuracy.
- Lightweight design trades representational power for speed.

iii. Model Size & Speed

- **~3.4M parameters** → the **smallest** model.
- **Inference speed: Fast** — ideal for mobile and real-time use.

vi. Overfitting / Underfitting Behavior

- Slight **underfitting**: train accuracy significantly lower than val/test.
- Model capacity is lower, so it can't learn very complex spatial patterns as deeply as ResNet or Inception.

vii. Impact of Augmentation

- **Augmentation helped significantly, especially:**
  - Random rotation
  - Random resized cropping
  - Color jitter
- These strengthen MobileNet's generalization due to its limited representational power.

8. Conclusion

Among the four models tested, **ResNet50** achieved the best overall performance with the highest validation and test accuracy, thanks to its residual connections that enable deeper learning without overfitting. **Inception v1** offered a good balance between accuracy and size, while **VGG19** was accurate but very large and slow. **MobileNetV2** was the fastest and smallest but had lower accuracy.

Data augmentation improved generalization across all models. For accuracy-critical tasks, **ResNet50** is preferred, while for speed or resource-constrained environments, **MobileNetV2** is the best choice.