



Plant Disease Classification with Vision Transformers and a CNN–Transformer Score Ensemble

Divyesh Shah^{1*}, Arun Warriar¹, Akshay Vhatkar¹,
Pooja Raundale²

¹Department of Computer Engineering, Sardar Patel Institute of Technology, Mumbai, Maharashtra, India.

²Department of Master of Computer Applications, Sardar Patel Institute of Technology, Mumbai, Maharashtra, India.

*Corresponding author(s). E-mail(s): divyesh.shah@spit.ac.in;

Contributing authors: arun.warrier@spit.ac.in;

akshay.vhatkar@spit.ac.in; pooja@spit.ac.in;

Abstract

This paper examines Vision Transformer (ViT) models for classifying leaf diseases in cinnamon, rice, potato, and maize images. We compare a ViT initialized from ImageNet-21k weights with a custom ViT trained from random initialization. The custom model has approximately 399 million parameters, making it a high-capacity experimental architecture rather than a model intended for resource-constrained deployment. We also evaluate an equal-weight score ensemble of the scratch-trained ViT and a pretrained VGG16 network on the two-class cinnamon task. In the detailed cinnamon evaluation output, the scratch-trained ViT obtained about 84% accuracy and a macro F1-score of 0.84. Averaging its class probabilities with those produced by VGG16 yielded 89.57% accuracy and a macro F1-score of 0.90. The ensemble substantially reduced RoughBark false negatives, although StripeCanker recall declined slightly. Its overall accuracy was essentially equal to the separately reported pretrained-ViT result for cinnamon. The evidence therefore supports a narrower conclusion: convolutional predictions improved the recorded scratch-ViT result, but the ensemble did not outperform the strongest transformer baseline.

Keywords: Plant Disease Classification, Vision Transformer, VGG16, Convolutional Neural Network, Ensemble Learning, Score Fusion, Precision Agriculture

1 Introduction

Leaf symptoms are routinely used to identify plant diseases, but visual inspection can be slow and inconsistent when symptoms are subtle, several diseases look alike, or trained personnel are unavailable. Image classifiers can assist this process by mapping a photograph of a leaf to a disease category. Early systems relied on engineered color, texture, and shape descriptors followed by conventional classifiers. Convolutional neural networks (CNNs) later replaced much of this manual feature design by learning task-specific representations directly from training images [8, 11].

Vision Transformers (ViTs) treat an image as a sequence of patches and use self-attention to model relationships among them [12]. This formulation gives the model access to long-range spatial context, but it does not encode locality as directly as a convolution. A ViT trained from random initialization may consequently require more data and optimization effort than a comparable CNN. Pretraining can reduce this burden, particularly for agricultural datasets that are modest in size [13].

The different inductive biases of CNNs and ViTs motivate hybrid models and prediction ensembles. A CNN can respond strongly to small lesions, edges, and texture changes, whereas a transformer can relate symptoms across distant regions of a leaf. Recent plant-disease work has combined these representations through compact hybrid networks, feature fusion, and dual-branch architectures [14, 16–18].

We study this relationship in two stages. First, pretrained and scratch-trained ViTs are compared on cinnamon, rice, potato, and maize datasets. Second, the scratch-trained ViT is paired with VGG16 through probability averaging for the cinnamon task. The paper makes the following contributions:

- a cross-dataset comparison of pretrained and randomly initialized ViTs under the recorded experimental settings;
- a class-level analysis of the cinnamon predictions, including the discrepancy between two stored scratch-ViT evaluation outputs;
- an evaluation of equal-weight probability fusion between the scratch-trained ViT and VGG16; and
- a documented account of dataset and protocol limitations that constrain the interpretation of the reported results.

2 Related Work

Conventional plant-disease classifiers commonly combined handcrafted image descriptors with algorithms such as Support Vector Machines, Random Forests, and Naive Bayes [1, 8]. CNN-based methods later produced stronger representations for crop-specific and multi-crop disease recognition [4, 9, 11]. Researchers have also used ensembles in which deep features or predictions from multiple models are combined before classification [7].

Dosovitskiy et al. introduced the ViT architecture for large-scale image recognition by applying transformer encoders to patch embeddings [12]. For plant images, transfer learning is especially useful because labelled datasets are often much smaller than general-purpose image corpora [13]. Hossain et al. evaluated a pretrained Data-efficient

Image Transformer against CNN baselines for mango leaf disease recognition and reported strong transformer performance [15]. Thakur et al. developed a compact CNN–ViT model with 0.85 million trainable parameters and tested it on five public plant-disease datasets [14].

Other studies have used more tightly coupled CNN–Transformer designs. Tunio et al. combined convolutional and Mobile ViT features with domain adaptation to address the shift from controlled images to field photographs [16]. Aboelenin et al. fused features from several pretrained CNNs with a ViT for apple and corn disease classification [17]. Meng et al. proposed a learnable dual-branch CNN–ViT model with 4.9 million parameters and evaluated it on PlantVillage and potato datasets [18].

Our experiments address a different question. Instead of designing a compact hybrid backbone, we compare a very large ViT trained from scratch with a pretrained ViT and then test whether VGG16 predictions can improve the scratch model on cinnamon images. Since the cited studies use different datasets, class definitions, splits, and evaluation procedures, their reported metrics provide context rather than a direct ranking against our results.

3 System Overview

The application design separates image acquisition from inference. A mobile interface captures or uploads a leaf image and sends it to a Python-based web service. The server preprocesses the image, runs the classifier, and returns the predicted category. Disease descriptions and supporting information may be retrieved from a database. The interface is designed to accommodate multiple languages.

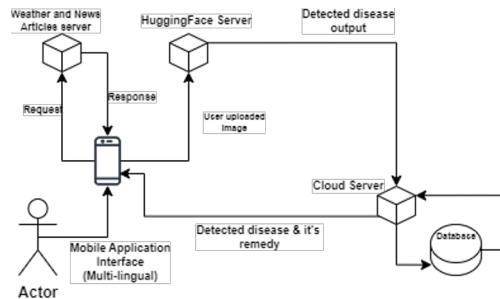


Fig. 1 Application-level design. Images are acquired on the mobile client and classified by the server-side model.

The scratch-trained ViT has roughly 399 million parameters and is not proposed for on-device inference. In the illustrated deployment, the mobile device handles image capture and result presentation; model execution remains on the server.

A ViT divides an image into fixed-size patches, flattens and projects each patch into an embedding, and adds positional information. Transformer encoder blocks process the resulting token sequence. The encoded classification token is then passed to a task-specific classification head.

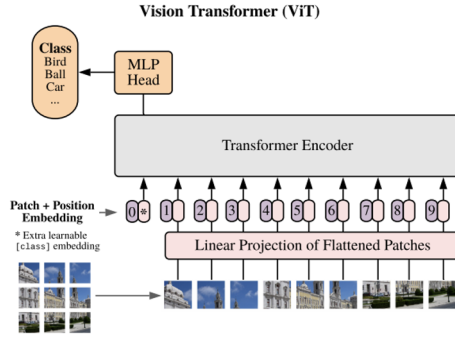


Fig. 2 Image-patch processing pipeline used by a Vision Transformer.

4 Datasets and Experimental Setup

The experiments use four public datasets containing cinnamon, rice, potato, and maize leaf images. Their class counts and recorded image totals are listed in Table 1. The diseases differ in visual scale: some appear as localized texture changes, while others produce large regions of discoloration or necrosis.

Table 1 Plant-disease datasets included in the experiments.

Crop	Dataset source	Classes	Images used/reported
Cinnamon	Plant Disease Dataset	2	326 (held-out set)
Rice	Rice Leaf Disease Image Samples	4	5,932
Potato	PlantVillage potato subset	3	2,152
Maize	PlantVillage maize subset	4	3,852

4.1 Cinnamon Dataset

The cinnamon experiment uses two categories from the Plant Disease Dataset hosted on Kaggle:

<https://www.kaggle.com/datasets/duggudurgesh/plantdataset>

The full public dataset contains 5,106 images across 20 plant-disease categories. Only *RoughBark* and *StripeCanker* were selected for this experiment. The class names and directory structure in the preserved notebook were used to verify the label assignment.

The available records contain a held-out set of 326 images: 158 *RoughBark* images and 168 *StripeCanker* images. The same set was used for the detailed scratch-ViT and ensemble evaluations. The records describe this set as 15% of the data, but they do

not retain the number of training images. We therefore report the observed held-out count without reconstructing an unverified total subset size.

Table 2 Class distribution in the held-out cinnamon set.

Class	Images
RoughBark	158
StripeCanker	168
Total	326

Images were converted to RGB and resized for each model. The pretrained ViT used 224×224 inputs and the normalization parameters supplied by its pretrained image processor. Random horizontal flipping was applied to training images only. Held-out images were resized and normalized without augmentation.

4.2 Rice Dataset

The rice experiment uses the Rice Leaf Disease Image Samples dataset from Mendeley Data:

<https://data.mendeley.com/datasets/fwcj7stb8r/1>

It contains 5,932 field-acquired images from four disease categories: Bacterial Blight, Blast, Brown Spot, and Tungro. Unlike the potato and maize subsets, it does not contain a healthy class.

Table 3 Class distribution in the rice dataset.

Disease class	Images
Bacterial Blight	1,584
Blast	1,440
Brown Spot	1,600
Tungro	1,308
Total	5,932

4.3 Potato Dataset

The potato images form a crop-specific subset of PlantVillage:

<https://github.com/spMohanty/PlantVillage-Dataset>

The subset contains 2,152 images from Early Blight, Late Blight, and Healthy categories. The Healthy class is much smaller than the two disease classes. The preserved experiment used class weighting during training to limit majority-class bias.

Table 4 Class distribution in the PlantVillage potato subset.

Class	Images
Early Blight	1,000
Late Blight	1,000
Healthy	152
Total	2,152

4.4 Maize Dataset

The maize experiment also uses a PlantVillage subset. Its 3,852 images represent Gray Leaf Spot, Common Rust, Northern Leaf Blight, and Healthy leaves.

Table 5 Class distribution in the PlantVillage maize subset.

Class	Images
Gray Leaf Spot	513
Common Rust	1,192
Northern Leaf Blight	985
Healthy	1,162
Total	3,852

PlantVillage images are generally photographed against controlled, visually simple backgrounds. Accuracy on these images should not be assumed to transfer unchanged to field photographs with uneven illumination, occlusion, clutter, or co-occurring symptoms.

4.5 Preprocessing and Evaluation Constraints

All images were converted to RGB. The pretrained ViT received 224×224 inputs normalized by its associated image processor. Its training pipeline included random horizontal flips. The scratch-trained ViT received 200×200 inputs scaled to $[0, 1]$ and divided into non-overlapping 25×25 patches.

The stored records do not establish one identical train-validation-test procedure for every dataset and model. For the cinnamon analysis, the scratch ViT and the

ViT-VGG16 ensemble were evaluated on the same 326 images. In the pretrained-ViT workflow, the held-out set was also passed to the training framework for checkpoint selection and early stopping. These values are therefore described as held-out evaluation results, not results from an untouched test set.

The records also do not provide complete documentation of duplicate detection, corrupt-file screening, augmentation parity, or class weighting for every configuration. We do not assume that undocumented steps were performed. A reproducible follow-up should generate a class-stratified train-validation-test manifest before training, remove duplicates prior to partitioning, restrict augmentation to the training split, and retain results from several random seeds.

4.6 Model Configurations

Three model configurations were evaluated: a pretrained ViT fine-tuned for each dataset, a custom ViT trained from random initialization, and a pretrained VGG16 CNN used in the cinnamon ensemble. The custom ViT has approximately 398.3 million parameters, rounded to 399 million throughout the paper.

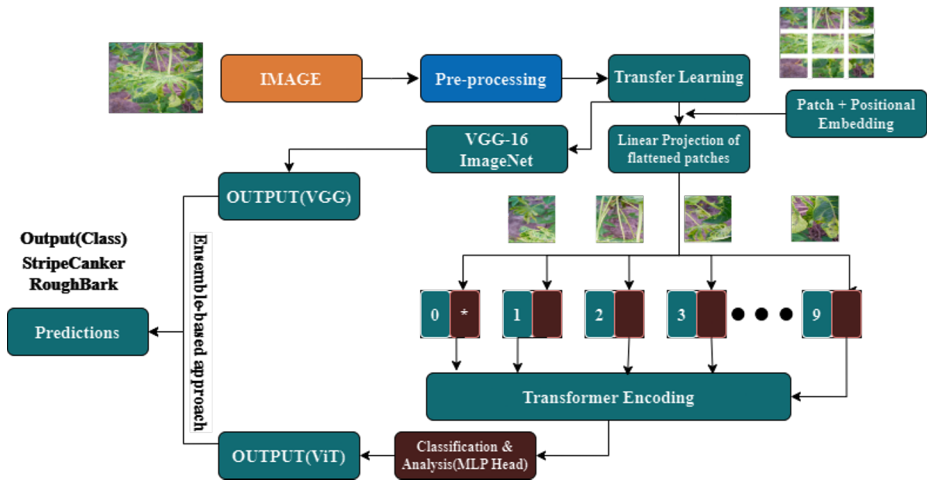


Fig. 3 Score ensemble formed from independently generated scratch-ViT and VGG16 probabilities.

The ensemble was chosen to test whether a convolutional classifier could offset errors made by the scratch-trained transformer on local cinnamon symptoms. We do not claim that the ensemble is smaller or faster than the pretrained ViT because the records contain no complete VGG16 configuration or measured latency.

Table 6 Architecture and training settings preserved in the experimental records.

Model	Architecture	Training configuration
Pretrained ViT	ViT-Base with 16×16 patches and ImageNet-21k weights (<code>google/vit-base-patch16-224-in21k</code>); 224×224 RGB input; 16×16 patches; 196 image patches plus a classification token; 12 encoder blocks; embedding dimension 768; 12 attention heads; MLP dimension 3072; approximately 85.8 million parameters; all layers fine-tuned.	Maximum 100 epochs; learning rate 2×10^{-5} ; training batch size 32; evaluation batch size 4; weight decay 0.01; evaluation and checkpointing after each epoch; best checkpoint selected by accuracy; early-stopping patience 10 and threshold 0.005; default Hugging Face optimizer and scheduler.
Scratch ViT	200×200 RGB input; 25×25 non-overlapping patches; 64 patches; flattened patch dimension 1875; embedding dimension 768; 12 encoder blocks; 12 attention heads; <code>key_dim</code> =768 per head; MLP dimension 3072; GELU activation; dropout 0.1; softmax classification head; approximately 398.3 million parameters.	Maximum 30 epochs; batch size 32; Adam optimizer; learning rate 1×10^{-4} ; gradient-value clipping at 1.0; categorical cross-entropy loss; learning-rate reduction factor 0.1; scheduler patience 10; minimum learning rate 1×10^{-10} ; checkpoint selected by validation loss; random seed 42.
VGG16	Pretrained CNN used as the convolutional member of the cinnamon ensemble. The stored records do not specify the input resolution, weight source, frozen layers, classifier head, or parameter count.	The stored records do not specify the optimizer, learning rate, batch size, number of epochs, scheduler, early-stopping rule, or training time.
Ensemble	Equal-weight averaging of the scratch-ViT and VGG16 softmax outputs.	No joint optimization was recorded. The class with the largest averaged probability was selected.

4.7 Score Fusion

For an image x , let $p_{\text{ViT}}(c \mid x)$ and $p_{\text{VGG}}(c \mid x)$ be the softmax probabilities for class c . The fused probability is

$$p_{\text{ens}}(c \mid x) = \alpha p_{\text{ViT}}(c \mid x) + (1 - \alpha) p_{\text{VGG}}(c \mid x), \quad (1)$$

with $\alpha = 0.5$. The predicted label is

$$\hat{y} = \arg \max_c p_{\text{ens}}(c \mid x). \quad (2)$$

4.8 Recorded Training Protocol

The cinnamon records describe a training subset and a 15% held-out subset. The held-out portion contains 326 images and is the common evaluation set for the detailed scratch-ViT and ensemble reports. The pretrained ViT used 224×224 images, model-specific normalization, and horizontal flips during training. The scratch ViT used 200×200 images normalized to $[0, 1]$ and split into 25×25 patches.

A separate test set was not consistently retained across all configurations. The pretrained-ViT training code used the held-out data for epoch-level evaluation, checkpoint selection, and early stopping. The reported metrics may therefore reflect model-selection decisions made on the same images. They should not be interpreted as an unbiased estimate from a fully independent test set.

Each reported result comes from a single recorded run. The experiments do not include confidence intervals, repeated seeds, or significance tests. These omissions prevent a statistical claim that one model would reliably outperform another under repeated training.

5 Results and Discussion

5.1 Pretrained and Scratch-Trained ViTs

Tables 7 and 8 reproduce the final metrics stored for the four datasets. Figures 4-7 show the corresponding validation curves.

Table 7 Final recorded metrics for the pretrained ViT.

Dataset	Final epoch	Accuracy	Loss	F1-score
Cinnamon	20	0.8958	0.0035	0.8960
Rice	20	0.9783	0.0213	0.9784
Potato	50	0.9951	0.0193	0.9951
Maize	20	0.9518	0.0199	0.9544

Table 8 Final recorded metrics for the scratch-trained ViT.

Dataset	Final epoch	Accuracy	Loss	F1-score
Cinnamon	20	0.8567	0.1838	0.8672
Rice	20	0.8370	0.3840	0.8464
Potato	50	0.9186	0.2317	0.9187
Maize	20	0.7975	0.4381	0.7990

The pretrained model records higher accuracy and F1-score on every dataset. The margin is 3.91 percentage points on cinnamon, 14.13 points on rice, 7.65 points on

potato, and 15.43 points on maize. These results are consistent with the use of pre-trained visual representations on datasets that are small relative to the capacity of the scratch model. The difference cannot be attributed to initialization alone, however, because the recorded training pipelines are not identical in all respects.

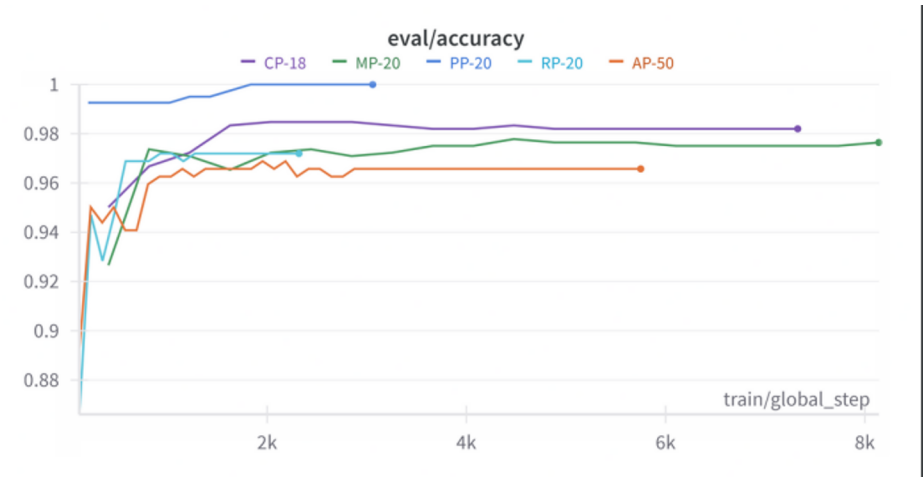


Fig. 4 Recorded validation accuracy for the pretrained-ViT experiments.

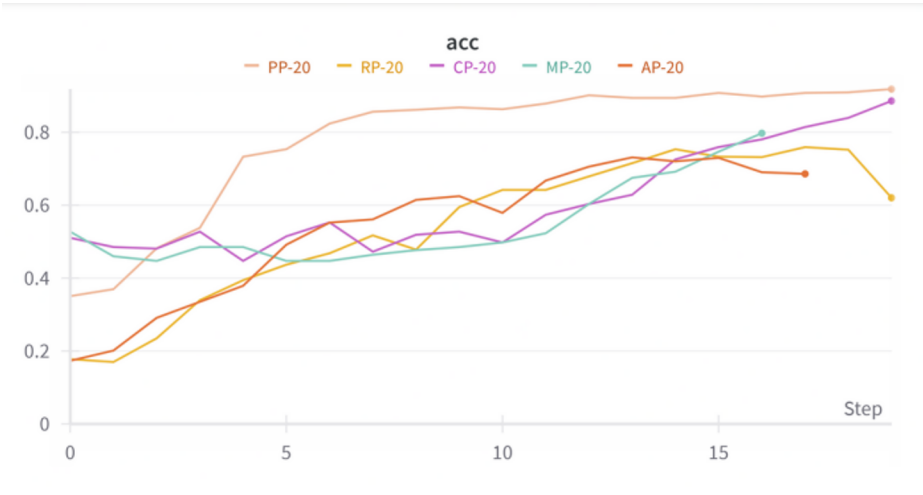


Fig. 5 Recorded validation accuracy for the scratch-ViT experiments.

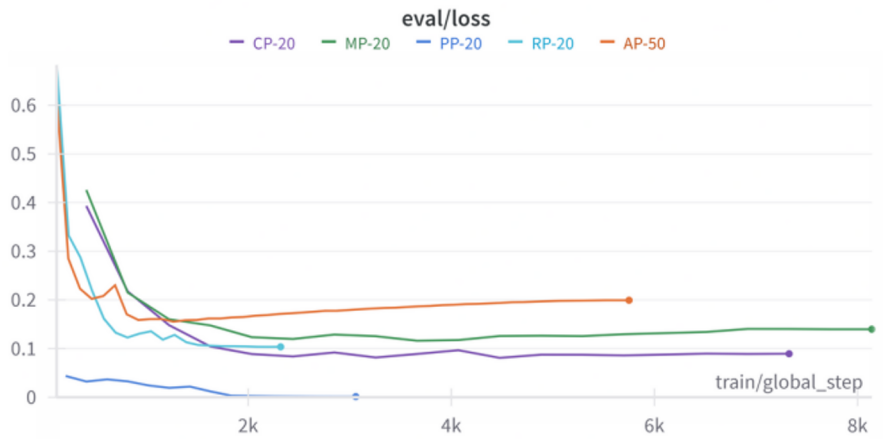


Fig. 6 Recorded validation loss for the pretrained-ViT experiments.



Fig. 7 Recorded validation loss for the scratch-ViT experiments.

5.2 Cinnamon Ensemble

Tables 9 and 10 give the detailed class reports preserved for the cinnamon set. Under this evaluation, probability averaging raises macro F1 from 0.84 to 0.90. The largest change occurs in RoughBark recall, which rises from 0.86 to 0.99. StripeCanker recall moves in the opposite direction, from 0.83 to 0.81. The ensemble therefore redistributes errors between the two classes rather than improving every class-specific measure.

Table 9 Detailed scratch-ViT report on the held-out cinnamon set.

Class / average	Precision	Recall	F1-score	Support
RoughBark	0.80	0.86	0.83	158
StripeCanker	0.88	0.83	0.86	168
Macro average	0.84	0.84	0.84	326
Weighted average	0.85	0.85	0.84	326
Accuracy		0.84		326

Table 10 Detailed ViT-VGG16 ensemble report on the held-out cinnamon set.

Class / average	Precision	Recall	F1-score	Support
RoughBark	0.83	0.99	0.90	158
StripeCanker	0.99	0.81	0.89	168
Macro average	0.91	0.90	0.90	326
Weighted average	0.91	0.90	0.90	326
Accuracy		0.90		326

The confusion matrix contains 292 correct predictions among 326 images, giving 89.57% accuracy. This is higher than the rounded 84% accuracy in the detailed scratch-ViT report. Table 8, however, lists a separate scratch-ViT accuracy of 85.67%. The available files do not identify whether the two values came from different checkpoints, code paths, or runs. We retain both outputs and use the detailed report only for the class-by-class ensemble comparison.

The pretrained-ViT cinnamon result is 89.58%, a difference of 0.01 percentage point from the ensemble. At the reported precision, the two methods are effectively tied. The ensemble evidence is therefore specific to the scratch model: adding VGG16 probabilities improves that recorded result, but does not surpass the pretrained ViT.

5.3 Context from Recent Studies

Table 11 places the experiment alongside recent CNN-Transformer studies. These values cannot be compared as a common benchmark because the datasets, class counts, image conditions, and validation procedures differ.

The present scratch ViT is far larger than the compact hybrids listed in the table. Parameter efficiency is therefore not an advantage of this implementation. Its result is useful as an ablation-style observation: a CNN prediction stream corrected enough scratch-ViT errors to match the recorded pretrained-ViT accuracy on the cinnamon set.

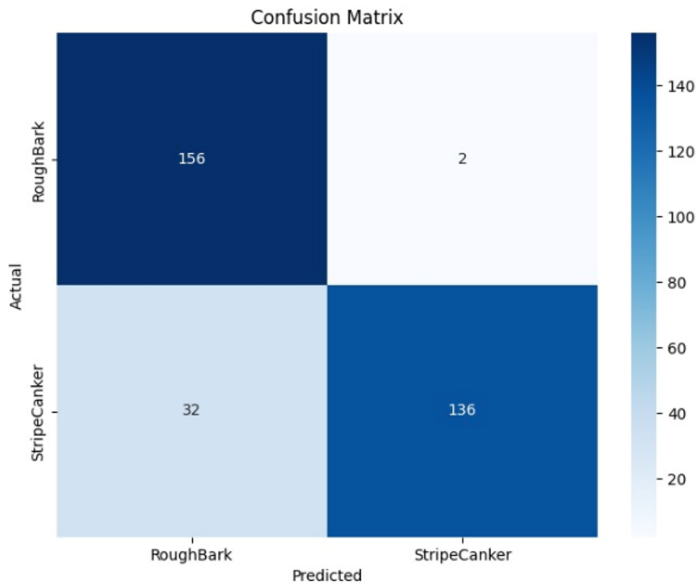


Fig. 8 Confusion matrix for the ViT-VGG16 ensemble on the held-out cinnamon set.

Table 11 Reported results from recent plant-disease studies. Cross-study values are not directly comparable.

Study	Architecture	Dataset(s)	Reported accuracy	Parameters
Thakur et al. [14]	Compact CNN-ViT hybrid	PlantVillage; Embrapa; others	98.86%; 89.24%	0.85 M
Hossain et al. [15]	Pretrained DeiT	Mango leaves	99.75%	Not stated
Tunio et al. [16]	CNN-MViT with domain adaptation	Three disease datasets	Better than reported baselines	Not stated
Aboelenin et al. [17]	Multi-CNN features with ViT	Apple; corn	99.24%; 98.00%	Not stated
Meng et al. [18]	Learnable dual-branch CNN-ViT	PlantVillage; potato	99.71%; 98.78%	4.9 M
Present study	Scratch ViT + VGG16 score averaging	Cinnamon	89.57%	399 M (scratch); ensemble N/A

6 Conclusion

The pretrained ViT records the highest accuracy among the two transformer configurations on cinnamon, rice, potato, and maize. The scratch model has approximately 399 million parameters yet performs worse on all four datasets, which makes it a poor candidate for direct deployment in its current form.

On the held-out cinnamon set, equal averaging of scratch-ViT and VGG16 probabilities produces 292 correct predictions out of 326 images. Macro F1 rises from 0.84 in the detailed scratch-ViT report to 0.90 in the ensemble report. Most of the gain comes from higher RoughBark recall; StripeCanker recall decreases slightly. The ensemble accuracy, 89.57%, is essentially the same as the separately reported pretrained-ViT accuracy of 89.58%.

The experimental record limits stronger conclusions. The held-out data were used for checkpoint selection in the pretrained-ViT workflow, the VGG16 training configuration is incomplete, and the results come from single runs. A follow-up evaluation should use a fixed three-way split, publish the complete VGG16 configuration, report VGG16 as a standalone baseline, repeat training across several seeds, and compare paired predictions on an untouched test set. Deployment work should focus on smaller backbones or compression methods and should include field images rather than controlled-background data alone.

References

- [1] V. Choudhary and A. Thakur, “Comparative Analysis of Machine Learning Techniques for Disease Prediction in Crops,” in *2022 IEEE 11th International Conference on Communication Systems and Network Technologies*, 2022, pp. 190–195, doi: 10.1109/CSNT54456.2022.9787661.
- [2] S. Harika, G. Sandhyarani, D. Sagar, and G. V. S. Reddy, “Image-based Black Gram Crop Disease Detection,” in *2023 International Conference on Inventive Computation Technologies*, 2023, pp. 529–533, doi: 10.1109/ICICT57646.2023.10134027.
- [3] T. S. D. R. Vanukuri et al., “K-Means Clustering Algorithm for Crop Leaf Disease Detection,” in *2023 7th International Conference on Computing Methodologies and Communication*, 2023, pp. 703–709, doi: 10.1109/ICCMC56507.2023.10083881.
- [4] M. Masood et al., “MaizeNet: A Deep Learning Approach for Effective Recognition of Maize Plant Leaf Diseases,” *IEEE Access*, vol. 11, pp. 52862–52876, 2023, doi: 10.1109/ACCESS.2023.3280260.
- [5] A. Dhakal and S. Shakya, “Image-Based Plant Disease Detection with Deep Learning,” *International Journal of Computer Trends and Technology*, vol. 61, 2018.

- [6] L. S. P. Annabel, T. Annapoorani, and P. Deepalakshmi, "Machine Learning for Plant Leaf Disease Detection and Classification—A Review," in *2019 International Conference on Communication and Signal Processing*, 2019, pp. 538–542, doi: 10.1109/ICCSP.2019.8698004.
- [7] S. Saxena and S. Rathor, "An Ensemble-Based Model of Detecting Plant Disease using CNN and Random Forest," in *2023 6th International Conference on Information Systems and Computer Networks*, 2023, pp. 1–6, doi: 10.1109/ISCON57294.2023.10112023.
- [8] S. Ramesh et al., "Plant Disease Detection Using Machine Learning," in *2018 International Conference on Design Innovations for 3Cs Compute Communicate Control*, 2018, pp. 41–45, doi: 10.1109/ICDI3C.2018.00017.
- [9] Z. Saeed, A. Raza, A. H. Qureshi, and M. H. Yousaf, "A Multi-Crop Disease Detection and Classification Approach using CNN," in *2021 International Conference on Robotics and Automation in Industry*, 2021, pp. 1–6, doi: 10.1109/ICRAI54018.2021.9651409.
- [10] U. Shruthi, V. Nagaveni, and B. K. Raghavendra, "A Review on Machine Learning Classification Techniques for Plant Disease Detection," in *2019 5th International Conference on Advanced Computing and Communication Systems*, 2019, pp. 281–284, doi: 10.1109/ICACCS.2019.8728415.
- [11] L. Li, S. Zhang, and B. Wang, "Plant Disease Detection and Classification by , Deep Learning—A Review," *IEEE Access*, vol. 9, pp. 56683–56698, 2021, doi: 10.1109/ACCESS.2021.3069646.
- [12] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *International Conference on Learning Representations*, 2021.
- [13] M. Xu, S. Yoon, Y. Jeong, and D. S. Park, "Transfer learning for versatile plant disease recognition with limited data," *Frontiers in Plant Science*, vol. 13, 2022, doi: 10.3389/fpls.2022.1010981.
- [14] P. S. Thakur, S. Chaturvedi, P. Khanna, T. Sheorey, and A. Ojha, "Vision transformer meets convolutional neural network for plant disease classification," *Ecological Informatics*, vol. 77, Art. no. 102245, 2023, doi: 10.1016/j.ecoinf.2023.102245.
- [15] M. A. Hossain, S. Sakib, H. M. Abdullah, and S. E. Arman, "Deep learning for mango leaf disease identification: A vision transformer perspective," *Heliyon*, vol. 10, no. 17, Art. no. e36361, 2024, doi: 10.1016/j.heliyon.2024.e36361.
- [16] M. H. Tunio et al., "Advancing plant disease classification: A robust and generalized approach with transformer-fused convolution and Wasserstein domain

- adaptation,” *Computers and Electronics in Agriculture*, vol. 227, Art. no. 109574, 2024, doi: 10.1016/j.compag.2024.109574.
- [17] S. Aboelenin, F. A. Elbasheer, M. M. Eltoukhy, W. M. El-Hady, and K. M. Hosny, “A hybrid framework for plant leaf disease detection and classification using convolutional neural networks and vision transformer,” *Complex & Intelligent Systems*, vol. 11, Art. no. 142, 2025, doi: 10.1007/s40747-024-01764-x.
- [18] Q. Meng, J. Guo, H. Zhang, Y. Zhou, and X. Zhang, “A dual-branch model combining convolution and vision transformer for crop disease classification,” *PLOS ONE*, vol. 20, no. 4, Art. no. e0321753, 2025, doi: 10.1371/journal.pone.0321753.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (<http://creativecommons.org/licenses/by-nc/4.0/>), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

