Computers, Materials & Continua

Tech Science Press

https://doi.org/10.32604/cmc.2026.084384

ARTICLE

STHarDNet: A Statistically Validated Swin Transformer–HarDNet Framework for High-Precision Plant Disease Detection and Classification

Amit Pimpalkar[1,*], **Kapil N. Vhatkar**[2], **Rachna K. Somkunwar**[3], **Shweta Koparde**[4], **Dalia H. Elkamchouchi**[5], **Ateeq Ur Rehman**[6,*], **Pooja Verma**[7] **and Salil Bharany**[8]

[1]School of Computer Science and Engineering, Ramdeobaba University, Nagpur, India
[2]School of Computer Science and Information Technology, Symbiosis Skills and Professional University, Pune, India
[3]Department of Computer Engineering, Dr. D. Y. Patil Institute of Technology, Pimpri, Pune, India
[4]School of Engineering and Technology, Dr. D. Y. Patil University, Ambi, Pune, India
[5]Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia
[6]School of Computing, Gachon University, Seongnam-si, Republic of Korea
[7]Uttaranchal Institute of Technology, Uttaranchal University, Dehradun, India
[8]Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, Punjab, India

*Corresponding Authors: Amit Pimpalkar. Email: amit.pimpalkar@gmail.com; Ateeq Ur Rehman. Email: 202411144@gachon.ac.kr

Received: 21 April 2026; Accepted: 02 June 2026; Published: 23 July 2026

ABSTRACT: Plants are fundamental to global food security; however, plant diseases significantly reduce agricultural productivity, making early and accurate detection essential. Traditional inspection approaches rely heavily on manual observation, which is labor-intensive, subjective, difficult to scale, and susceptible to human error. In contrast, artificial intelligence (AI) combined with computer vision (CV) offers an effective solution for early-stage disease detection, minimizing yield losses while overcoming the limitations of manual monitoring systems. In this study, a novel deep learning architecture, the Swin Transformer with Harmonic Densely Connected Network (STHarDNet), is proposed. The framework integrates a Swin Transformer (ST) as the initial skip connection within a HarDNet-based U-Net architecture to precisely localize diseased regions in leaf images. Subsequently, a modified ResNet-152 model is employed for disease classification. The ST component captures long-range dependencies at high resolution, enabling enhanced feature representation and more accurate boundary delineation. To ensure robustness and reliability, extensive statistical validation techniques are applied, including 5-fold cross-validation, bootstrap confidence intervals, Gelman-Rubin convergence diagnostics, Cohen's d effect size, and paired t-tests with Bonferroni correction. These analyses confirm both the statistical stability and practical effectiveness of the proposed model. Experiments conducted on the hybrid PlantVillage dataset, comprising 20,798 images across 17 classes, demonstrate that STHarDNet achieves an outstanding classification accuracy of 99.81%, outperforming existing methods across multiple evaluation metrics. This research establishes a reproducible and statistically validated benchmark for automated plant disease detection (PDD), supporting its scalability in precision agriculture. Furthermore, the proposed system highlights the potential of intelligent, high-accuracy tools to assist non-expert users in identifying plant diseases at early stages, thereby enabling timely intervention and improved crop management.

KEYWORDS: Deep learning; swin transformer; HarDNet; ResNet-152; convolutional neural network

1 Introduction

Agriculture is a major source of food, income, and economic growth worldwide. In India, it contributes around 18% to the national gross domestic product (GDP), while in many low and middle-income countries it employs nearly 42% of the population [1]. The sector's contribution to the economy has increased from 17.6% to 20.2% in recent years [2]. Early detection of plant diseases and pests can improve agricultural production and decision-making [3]. In particular, every medical condition and pest situation has distinct patterns that are diagnostically useful for categorizing anomalies [4,5]. Moreover, manual inspection to identify plant infection is time-consuming, subjective, and may lead to incorrect diagnosis. Recent advances in inexpensive image-acquisition systems have enabled the acquisition of numerous images for image-based plant disease diagnosis. Nevertheless, digital images contain compressed data that is highly challenging for computers to comprehend, necessitating additional procedures such as preprocessing and segmentation to extract properties such as color and shape [6–10]. Precision agriculture might have benefited from integrating advanced AI and CV technologies, providing a platform for more precise recognition of plant diseases [11]. Deep learning further enhances precision agriculture by automatically learning discriminative features without human intervention.

This research proposes a hybrid method for plant disease detection (PDD) and classification, the STHarDNet segmentation model. The model employs a ResNet-152 for Semantic Leaf Segmentation (SLS). The approach effectively discriminates between healthy and diseased districts by emphasizing the foreground and background regions. The high-density maps are encoded by the model, which then divides them into several disease categories. The novel contributions of the proposed STHarDNet framework are listed as follows:

1. The STHarDNet model introduces a novel hybrid segmentation architecture by integrating the Swin Transformer (ST) within the original skip connections of the HarDNet-based U-Net rather than at the bottleneck layer. This design enables the model to capture long-range contextual dependencies on high-resolution feature maps, thereby significantly improving the segmentation accuracy of irregular and complex disease lesions.
2. The study developed a deep learning–based framework for automatically labeling and classifying 17 plant disease categories from image data. Using a strategically enhanced ResNet-152 architecture with residual shortcut connections to alleviate the vanishing gradient problem, the proposed model achieved a classification accuracy of 99.81% on the hybrid PlantVillage dataset. The model outperformed several state-of-the-art approaches, including Capsule Networks, VGG16, DenseNet101, and conventional CNN-based architectures.
3. Unlike other studies that report only accuracy, our model provides a comprehensive measure of context and data imbalance, optimally balancing their effects on segmentation for a (250×250) patch size.
4. The model has been statistically validated using techniques such as 5-fold cross-validation, bootstrap confidence intervals, Gelman–Rubin ($\hat{R}$) convergence diagnostics, Cohen's 'd' effect size, and paired t-tests with Bonferroni correction to verify its stability and practical significance.

The remainder of this paper is organized as follows. Section 2 reviews the existing literature, focusing on conventional deep learning–based approaches and identifying the research gaps that motivate the proposed framework. Section 3 presents the proposed methodology and describes each component in detail. Section 4 discusses the experimental results and provides a comparative analysis with existing studies. Finally, Section 5 concludes the paper and outlines potential directions for future research.

2 Literature Survey

A crucial initial phase in the prevention and treatment of plant diseases is prompt identification. Farmers can implement preventive and therapeutic measures by using accurate methods for disease identification. Authors in [12,13] established the conceptual foundation of self-supervised learning (SSL) in PDD. They proposed ConMamba, a state-space-based SSL framework with dual-level contrastive learning that captures long-range dependencies and improves performance on benchmark plant disease datasets. To more precisely classify multi-crop leaf Disease (MCLD) images, the Visual Geometry Group (VGG) model was initially demonstrated in [14,15]. In [16,17], researchers combined feature extraction with deep learning techniques for plant disease detection (PDD) using architectures such as ResNet-50, GoogleNet, and VGG16, followed by traditional classifiers including KNN and SVM. In [18], the CaffeNetDL framework was proposed to collect and enhance plant leaf image datasets for crop disease analysis. Studies in [19,20] introduced DCNN-based PDD models incorporating data augmentation and hyperparameter optimization. The authors in [21] emphasized the importance of early disease detection for improving agricultural productivity. Lightweight and efficient architectures, including InceptionV3, InceptionResNetV2, MobileNetV2, and EfficientNet, were explored in [22] to reduce computational complexity while optimizing performance through tuning of batch size, dropout, and epochs. In [23], the ResNet-197 model demonstrated the effectiveness of deep residual CNNs for PDD, highlighting the role of hyperparameter optimization and data augmentation. Furthermore, studies in [24,25] emphasized the significance of early disease detection using DCNN-based approaches. Table 1 summarizes the some existing studies considering their key contributions and limitations.

Table 1: Comparative analysis of existing literature with key contributions and their limitations.

Paper (Year)	Method Used	Evaluation Metrics	Key Contributions	Limitations/Gaps
Alsuhaibani et al. (2026) [26]	DL-based PDD (CNN/Hybrid DL)	Accuracy: 95%–99%	- Uses advanced DL for crop disease detection - Focus on improving classification robustness	- Evaluated on controlled datasets - Limited real-world validation
Ananthajothi et al. (2026) [27]	Infrared imaging + DL for early detection	Accuracy: 70%–92%	- Uses NIR imaging for early disease detection - Works in natural environmental conditions	- Lower accuracy compared to RGB-based models - Sensitive to environmental noise and requires specialized sensors
Sankhe and Ambhaikar (2026) [28]	Ceta-Coyote calibrated DCNN	Accuracy: 98.27%, Sensitivity: 98.68%, Specificity: 97.04%	- Focus on intelligent agriculture systems - Integration of AI techniques	- Lack of reported quantitative metrics - Limited benchmarking against state-of-the-art (SOTA)
Arumugam and Arumugam (2026) [29]	Butterfly Optimization Algorithm with adaptive crossover	Accuracy: 98.97% F1: 97.57%	- Enables early and accurate detection of apple leaf diseases	- Limited DL-based benchmarking - Lack of detailed performance metrics

(Continued)

Table 1 (continued)

Paper (Year)	Method Used	Evaluation Metrics	Key Contributions	Limitations/Gaps
Dey et al. (2026) [30]	EFFINCEP-NET, EfficientNet-B4	Accuracy: 98.65%	- Emphasis on PDD in real crops - Bridges plant science and AI	- Insufficient technical details on model performance - Limited reproducibility
Devarajan et al. (2026) [31]	CNN-based real-time PDD	Accuracy: 98.32% Precision: 97.85% Recall: 98.14% F1: 97.99% Inference: 42.6 ms	- Real-time disease diagnosis - Scalable and retrainable framework - Supports precision agriculture	- Limited dataset diversity - Possible overfitting - No cross-dataset validation
Sahu et al. (2025) [32]	Hybrid dilated CNN + Attention + Optimized Mask R-CNN	Accuracy: 84%–97.49%	- Combines segmentation + classification - Uses attention for improved feature extraction	- Missing explicit metric values - High model complexity - Limited real-world validation
Murugavalli and Gopi (2025) [33]	Vision Transformer (ViT) for PDD	Accuracy: 99.88% Validation accuracy: 99.88% External accuracy: 99.83%	- Introduces transformer-based detection - Better global feature representation	- Computationally expensive
Ibrahim et al. (2025) [34]	AI + IoT-based smart agriculture system with ResNet50 model	Accuracy: 99.8% F1-score: 99.91% Recall: 99.92% Precision: 100%	- Integrates IoT, UAV, and AI - Enables real-time monitoring and treatment	- System complexity - Hardware is expensive
Bijlwan et al. (2025) [35]	ANN, Random Forest, Regression models	R^2: 0.96–0.98 Validation R^2: 0.93–0.95	- Uses weather data for disease prediction - Strong predictive modeling	- Not image-based detection - Limited generalization
Zhang and Xu (2025) [36]	DL-based PDD	Accuracy: >95%	- Efficient DL-based classification - Focus on practical deployment	- Dataset dependency - Limited robustness in field conditions
El Hanafy et al. (2025) [37]	AI-driven detection with optimization	Accuracy: 96%–99%	- Combines optimization with DL	- High computational cost - Limited interpretability
Hammam et al. (2025) [38]	CNN-based classification model	Accuracy: 94%–98%	- Lightweight CNN design - Focus on efficiency	- Lower accuracy vs. SOTA - Limited dataset scope
Krishna et al. (2025) [39]	ML + image processing hybrid	Accuracy: 90%–95%	- Hybrid classical + ML approach - Reduced complexity	- Lower performance than DL - Less scalability

(Continued)

Table 1 (continued)

Paper (Year)	Method Used	Evaluation Metrics	Key Contributions	Limitations/Gaps
Babu et al. (2025) [40]	DCNN with optimization	Accuracy: 99%	- High accuracy on benchmark datasets - Optimized training strategy	- Risk of overfitting - Lack of real-world validation

Despite substantial advances in the PDD, significant gaps remain in current research. Computational complexity, inference time, and hardware requirements are often not reported, and the deployment feasibility is unclear. Other studies do not specify assessment measures or statistical significance tests, which lowers reliability. Moreover, in very complex hybrid and optimization-based models, the high training cost reduces reproducibility. Thus, there is a strong need for a robust, effective, and generalizable model tested in the agricultural environment with a detailed performance analysis.

3 Methodology

This research presents a novel DL-based segmentation for PDD, aiming to enhance accuracy, efficiency, and scalability. Fig. 1 is an architectural illustration of our proposed approach, where segmentation operates on a (256 × 256) image input. In contrast, classification operates on (224 × 224) images and regions of interest (ROIs) extracted from the segmentation mask. Initially, images are split into training and test sets (80:20), and augmentation and normalization are applied. The most fundamental novelty is a segmentation component that embeds an ST into the HarDNet encoder-decoder's initial skip connection, enabling global attention at high resolution. By inserting it at the first skip, the model can propagate global contextual information across the entire high-resolution feature map, which is critical for segmenting irregular disease margins. The segmented mask enables ROI extraction for classification with a modified ResNet-152 and a custom dense head. The model is trained using a hybrid loss function that combines Dice, cross-entropy, and classification, with SGD. Convergence is tracked iteratively, followed by statistical validation to assess robustness, reliability, and the model's generalization. A convergence point of the decision assesses the $\hat{R}$ statistic ($\hat{R} \leq 1.1$) which guides the further optimization of the statistical validation. The last step calculates overall performance measures, and it includes a feedback loop that allows the model to be refined.

3.1 Dataset Description

A PlantVillage dataset is employed to classify plant leaf diseases, a widely used database for plant disease recognition [41,42]. To enhance the diversity and robustness of the plant disease classification framework, we used a hybrid data fusion approach. Here, we used two complementary datasets, one with 15 classes (potato, tomato, pepper bell) and another with 19 classes (apple, corn, grape, citrus), rather than adhering to a single benchmark dataset with 39 classes. Strict class reconciliation was performed to avoid class overlap, using semantic alignment and perceptual hashing to identify class overlaps between images. It is a multi-source strategy that improves generalization across acquisition protocols and lighting conditions, enabling STHarDNet to learn dataset-avoidant feature representations that can be utilized in a real environment. The harmonized PlantVillage dataset collection includes 20,798 color leaf images with a uniform foreground, with 16,638 images in the training sets and 4160 in the test set. For cross-validation experiments, 5-fold stratified cross-validation was engaged, confirming each fold preserved the original class proportions.

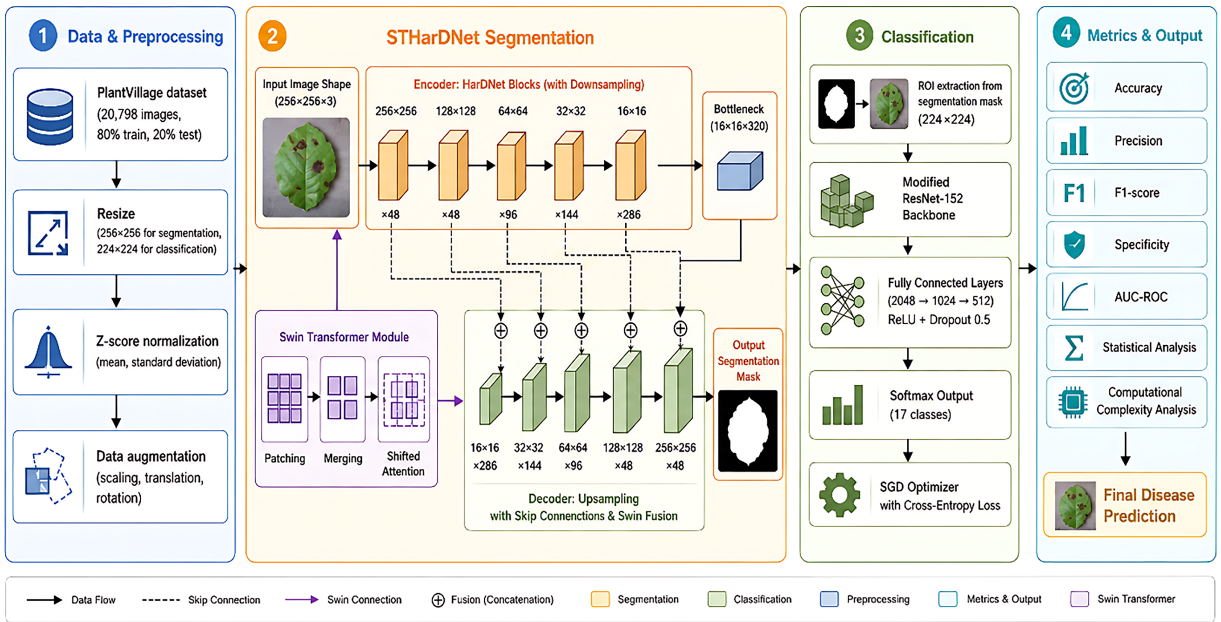


Figure 1: Architecture illustration of our proposed approach.

3.2 *Pre-Processing*

The input image is normalized using z-score normalization, which accounts for the training set mean. Z-score normalization was applied using the dataset's mean (μ = [0.485, 0.456, 0.406]) and standard deviation (σ = [0.229, 0.224, 0.225]) computed from the training split. The training images in this study were balanced using three augmentation strategies: translation, scaling, and rotation. To enhance generalization, augmentation was applied only to training images using the augmentations library: random rotation (15 ± 5°), horizontal/vertical flips (0.5 probability), random scaling (1 ± 0.1), translation (7.5 ± 2.5°), Gaussian noise additions (σ = 0.1), and contrast of the brightness (1 ± 0.2).

3.3 *Segmentation*

3.3.1 HarDNet

The HarDNet block serves as the foundation for the model's encoder-decoder structure, a U-Net representation, as depicted in Fig. 2. The term "encoder" refers to a procedure that uses downsampling to reduce image size while extracting a feature map. The encoder is composed of four HarDNet blocks, four down-sampling blocks, and one convolution block.

A convolutional layer with kernel size = 3, stride = 1, and filter size = 48 makes up the convolutional block, and for kernel size = 3 and for stride = 2, the filter size = 24. A downsampling block consists of an average pooling layer and a convolutional layer with a kernel size of 1. A HarDNet block is used in the transition stage (bottleneck) to complete parameter transfer for the encoder and decoder. The decoder processes the final convolutional layer with kernel size = 1, five up-sampling blocks, four HarDNet blocks, and five final convolutional layers before generating the final (W, H) size image. An up-sampling block is made up of a convolution layer with kernel size = 1 and an interpolating function that operates in the "bilinear" mode. When an input image is specified as (W (width), H (height), and C (channels)), the encoder's convolutional block computes a feature map of size (W/2, H/2, 48). By employing four downsampling blocks and four HarDNet blocks, a feature map of (W/32, H/32, 286) is generated. The decoder continuously transforms the feature map of (W, H, and class) from the bottleneck's (W/32, H/32, 320) feature map. The class indicates the number of distinct disease categories in the data.

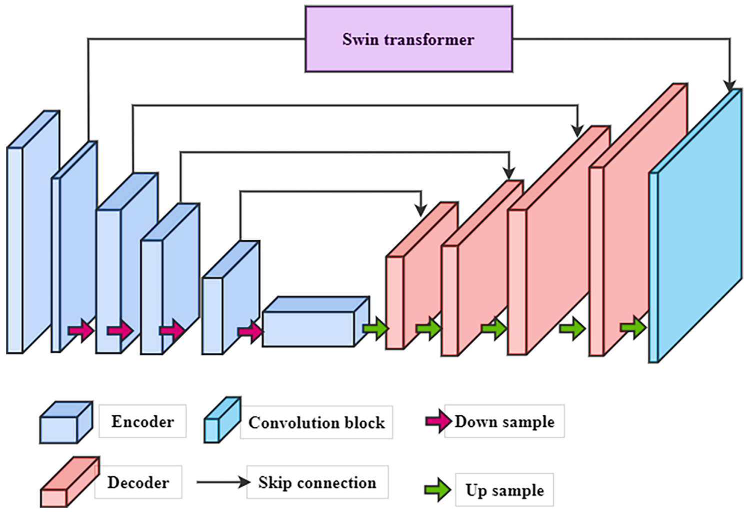


Figure 2: HArDNet model architecture.

3.3.2 Swin Transformer (ST) Block

The ST is used in the first skip connection, which connects the high-resolution feature map from the encoder's initial stage to the corresponding decoder stage. The input image is split into (4 × 4) patches after traversing the patch partition layer, producing patch tokens with shapes W/4, H/4, and (4 × 4) channels. In stage 1, the created patch tokens pass via the linear embedding. After that, they are fed into a junction of associated ST blocks, resulting in tokens of (W/4, H/4, C), where C may represent any dimension. ST blocks and patch merging constitute stages two and three, respectively. Patch merging involves combining neighboring (2 × 2) patches into a single patch, downsampling the tokens to 1/2, and doubling C. In stages two and three, the token shapes are (W/8, H/8, 2C) and (W/16, H/16, 4C), respectively. The feature dimension in the ST's final "Linear Projection" is eight times larger than the input dimension. The output is a feature map with the same shape after performing a (1 × 1) convolution. The first HarDNet block of the encoder transmits an output of (112 × 112 × 48) feature map to the ST model, which is then attached to the last decoder HarDNet block.

3.4 ResNet-152 Neural Network Classification Architecture

To effectively achieve our goal of classifying various categories of plant diseases, we employed the ResNet-152 CNN architecture and made the necessary modifications. The global average pooling layer and the initial fully connected (FC) layer of ResNet-152 are eliminated. It has a new classification head consisting of flattened layers followed by dense layers with fixed numbers of units (2048, 1024, 512) and dropout (0.5) as a regularization parameter, as shown in Fig. 3. Stochastic gradient descent (SGD) was used to train the STHarDNet segmentation network with a Nesterov momentum of 0.9, an initial learning rate of 0.0001, and a cosine annealing scheduler with a minimum learning rate of 1e−6. Segmentation and classification batch sizes of 16 and 32 were selected via grid search. The classification head used categorical cross-entropy loss with L2 regularization (weight decay = 1e−4). Using patience (15 epochs) to stop early prevented overfitting, and model checkpointing saved the best validation weights.

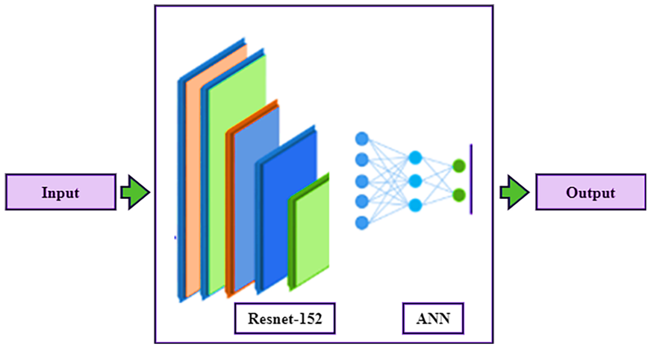


Figure 3: Schematic of the network's architecture.

Furthermore, we used "Softmax" as the activation function for the prediction layer and "ReLU" for the remaining ANN layers. Finally, we employed "Categorical Cross-entropy" as our loss function. The classified regions of the leaves were down-sampled to (224 × 224) pixels before classification. The pre-trained ResNet-152 model we have modified takes images in the format (224, 224, 3), which is in line with the ImageNet format. Subsequently, our algorithm updates its weights and begins identifying patterns in the images.

The STHarDNet proposed framework incorporates segmentation and classification concepts into an all-encompassing PDD Algorithm 1. An ROI is extracted from the resulting segmentation mask and classified using a deep residual network with fully connected layers. The hybrid loss, comprising Dice, cross-entropy, and classification objectives, is used to optimize the model. The standard metrics are used to assess the performance, which is further confirmed by cross-validation, bootstrap confidence interval, $\hat{R}$ convergence, and Cohen 'd' effect size estimation.

Algorithm 1: STHarDNet framework for segmentation–classification with statistical validation

---

Input: *Dataset* $D = \{(I_i, y_i)\}_{i=1}^{N}$ where, I_i is the input image and y_i is an image-level class label

Step 1: *Data Preparation: The dataset D is split into training and testing (80:20) subsets using stratified sampling, and each image I_i is transformed into two representations and M_i is generated mask:*

$I_{seg} = Augment(Normalize(Resize(I_i, 256 \times 256)))$

$I_{cls} = Resize(I_i, 224 \times 224)$

$M_i = MorphologicalRefinement(Threshold(ContrastEnhancement(ColorSpaceTransform(I\_\{i\}))))$

Step 2: *Segmentation using STHarDNet: The segmentation network combines HarDNet-based encoding with transformer-enhanced feature fusion Encoder representation:*

$F_0 = Conv(I_{seg})$ *and* $F_i = Downsample(HarDNet(F_{i-1}))$, *for* $i = 1, 2, 3, 4$

Transformer-enhanced feature: $F_{swin} = SwinTransformer(F_1)$

Decoder with skip connections:

$G_i = HarDNet(Upsample(F_{i+1}) \oplus F_i)$, *for* $i = 4 \rightarrow 2$

$G_1 = HarDNet(Upsample(G_2) \oplus F_1 \oplus F_{swin})$

Final segmentation output: $\hat{M} = Conv_{1\times1}(G_1)$

Step 3: *Region-Based Classification: The predicted mask $\hat{M}$ is used to extract the region of interest from I_{cls} element-wise masking. ROI = ExtractROI($\hat{M}$, $\odot$ I_{cls})*

Feature extraction and classification: $f = ResNet152(ROI)$

$h_1 = ReLU(W_1 f + b_1)$; $h_j = \delta\left(W_j h_{j-1} + b_j\right), j = 2, 3, \ldots, h_4 = Dropout(h_3, p = 0.5)$ *and* $\hat{y} = Softmax(W_4 \ h_4 + b_4)$

(Continued)

Algorithm 1 (continued)

Step 4: *Model Optimization: The model is trained by minimizing a composite loss function:*

$\mathcal{L}_{dice} = 1 - \frac{2\sum(M_i \hat{M}_i)+\epsilon}{\sum M_i + \sum \hat{M}_i + \epsilon}$; $\mathcal{L}_{seg}^{CE} = -\sum_p M_i^{(p)} \log(\hat{M}_i^{(p)})$; $\mathcal{L}_{cls} = -\sum_{c=1}^{K} y_{ic} \log(\hat{y}_{ic})$

$\mathcal{L}_{total} = \mathcal{L}_{dice} + \mathcal{L}_{seg}^{CE} + \lambda \mathcal{L}_{cls}$; *where $\lambda = 0.5$ controls the classification contribution.*

Parameters are updated using SGD: $\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta \mathcal{L}_{total}$ (where η is the learning rate.)

Step 5: *Performance Evaluation: Predictions are evaluated using confusion matrix elements (TP, TN, FP, FN).*

Accuracy, Precision, Recall, F1-score, $Dice = \frac{2|M \cap \hat{M}|}{|M| + |\hat{M}|}$ and $Jaccard = \frac{|M \cap \hat{M}|}{|M \cup \hat{M}|}$

Step 6: *Statistical Validation*

Cross-validation: The dataset is partitioned into k = 5 folds.

Mean and standard deviation are computed as: $\mu = \frac{1}{k}\sum_{i=1}^{k} m_i$; $\sigma = \sqrt{\frac{1}{k}\sum_{i=1}^{k}(m_i - \mu)^2}$

Bootstrap confidence interval: Re-sampling (B = 1000) is performed to estimate:

$CI_{95\%} = [P_{2.5}(\mu^*), P_{97.5}(\mu^*)]$

$\hat{R}$ convergence statistic: Let W denote within-chain variance, and B denote between-chain variance:

$\hat{R} = \sqrt{\frac{(1-\frac{1}{n})W + \frac{1}{n}B}{W}}$

Effect size (Cohen's d): For comparison with a baseline model: $d = \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^2 + \sigma_2^2}{2}}}$

Statistical significance (paired t-test with Bonferroni correction): $t = \frac{\mu_1 - \mu_2}{\sqrt{\frac{\sigma_1^2 + \sigma_2^2}{k}}}$ and $p_{\text{corrected}} = \min(1, p \times m)$

Final Output: *The framework returns learned model parameters θ, predicted outputs ($\hat{M}$, $\hat{y}$), and statistical validation measures (μ, σ, CI, $\hat{R}$, d, p).*

4 Results and Discussion

4.1 Experimental Setup

Each experiment was run on a high-performance computing workstation with a NVIDIA GeForce GTX 3080 GPU (10 GB GDDR6X), an Intel Core i9-12900K (16 cores, 3.2 GHz), and 64 GB of RAM. The system ran Windows 11 Pro with CUDA 11.8 and cuDNN 8.6, which served as the driver for GPU use. Python 3.10, TensorFlow 2.12, Keras 2.12, OpenCV 4.8, NumPy 1.24, and Scikit-learn 1.3 were used as software implementations. Theano was used as the backend for optimized tensor operations.

4.2 Performance Metrics

The definition of the performance criteria is as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{2}$$

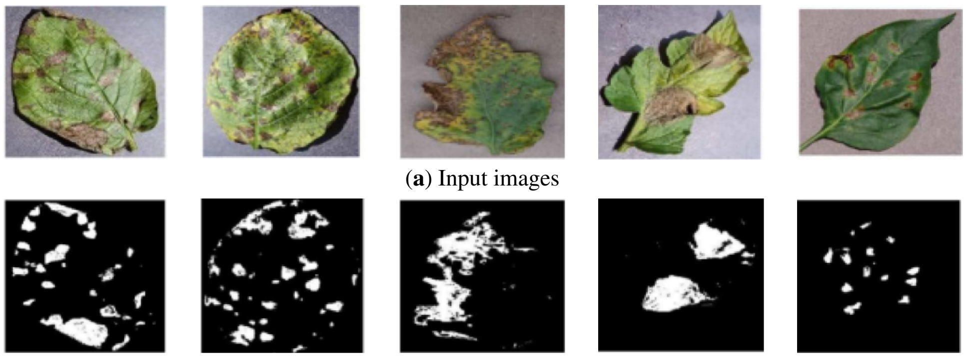
$$\text{Sensitivity} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{4}$$

$$\text{F1} - \text{score} = \frac{2 * Precision * Recall}{Precision + Recall} \tag{5}$$

4.3 Performance Analysis of Segmentation and Classification

The results of the plant disease segmentation procedure for each dataset are presented in Fig. 4. The initial images after preprocessing are shown in the first row. The second row shows the segmentation mask results generated by a sequential preprocessing pipeline comprising color-space transformation, contrast enhancement, threshold-based leaf extraction, and morphological refinement.



(**a**) Input images

(**b**) Corresponding binary segmentation masks

Figure 4: Automatically generated preprocessing-based lesion masks for representative diseased leaf samples.

4.4 Different Patch Sizes Comparison

Usually, the afflicted region has fewer pixels than the entire input image and differs in size from the location. Therefore, by selecting an appropriate patch size that supports the distinctive representation of the affected areas, the data imbalance issue can be resolved. Table 2 and Fig. 5 show the distribution of segmentation performance for the suggested method.

Table 2: Performance of the various patch sizes utilized in the proposed method.

Patch Size	Dice ↑	Jaccard ↑	Precision (%)	F1-Score (%)	Sensitivity (%)	Specificity (%)
125 × 125	0.71	0.6	72.13	64.53	90.65	99.97
150 × 150	0.78	0.7	66.54	54.32	90.21	99.98
175 × 175	0.75	0.72	74.21	62.98	92.54	99.98
200 × 200	0.77	0.66	74.98	64.32	87.68	99.99
225 × 225	0.73	0.68	76.54	72.11	90.32	99.99
250 × 250	**0.77**	**0.74**	**83.63**	**76.54**	**85.48**	**99.99**
275 × 275	0.76	0.72	72.13	74.32	91.2	99.98
300 × 300	0.8	0.69	79.54	72.84	88.9	99.99
325 × 325	0.85	0.76	78.65	75.43	90.01	99.98

Entries in bold represent the best-performing results for each metric across all considered patch sizes.

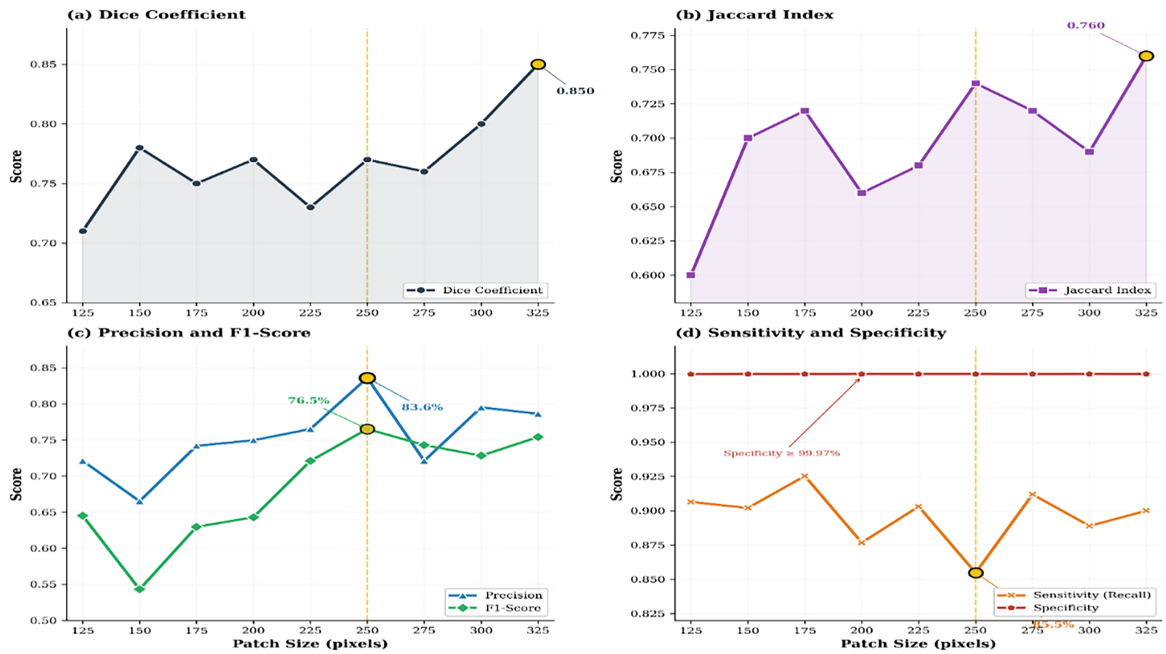


Figure 5: Performance of segmentation and classification metrics across varying patch sizes (125–325 pixels).

A thorough analysis of segmentation performance across nine patch sizes (125 × 125 to 325 × 325) was conducted and tabulated in Table 2, which reports classification metrics and segmentation-specific metrics. The patch size of (250 × 250) performs best, achieving the highest precision (83.63%), F1-score (76.54), and specificity (99.99%). The Dice coefficient (0.77) and Jaccard Index (0.74) at this setup indicate better region overlap, i.e., greater specificity of the disease boundary. It is also important to note that background noise is introduced by larger patch sizes (≥275), which, in turn, leads to lower accuracy despite rising Dice scores. On the other hand, smaller patch sizes (≤200) do not provide sufficient spatial context and therefore yield lower F1-scores. These results confirm that (250 × 250) is the best input resolution for the proposed STHarDNet model, as it balances spatial context and data efficiency.

Fig. 5 shows the distributional assessment results obtained with different patch sizes. Every point on a curve represents the segmentation result for the corresponding patch size, and each curve indicates the measurement score. The subplots provide segmentation and classification measures of patch sizes (125 to 325) pixels. The segmentation quality (5a), as measured by the Dice coefficient, is good, with a peak of 0.85 at (325 × 325). Similarly, (5b) shows the Jaccard index follows the same trend, with 0.76 at (325 × 325). Precision and F1-score (5c) reach peaks at (250 × 250) (83.63% and 76.54%, respectively), indicating a good compromise between false positives and overall classification accuracy. Sensitivity (5d) ranges from 85% to 92%, and specificity is 99.97% to 99.99%. Fig. 6 shows the performance of training and testing over 100 epochs using the hybrid PlantVillage dataset. Fig. 6a demonstrates accuracy curves for both the training and validation sets, which begin at the baseline of random accuracy (23% and 21%, respectively), and the improvement in accuracy occurs in the initial 50 epochs. Accuracy is over 68% (training) and 63% (validation) at epoch 20, which is similar to the benchmarks, and the final test accuracy is 99.81%.

Fig. 6b exhibits an exponential decrease over the range of about 2.8 to almost zero, with an exponentially steep reduction at the early stages of the curve (epochs 1–20) and a constant convergence in the final stages. The confusion matrix for each class is shown in Fig. 7, and the model considered 17 different classes. It is thought that the suggested approach has improved the accuracy of class labels in the PDD confusion matrix while simultaneously lowering the FP rate for individual leaf images. Fig. 8 shows the Receiver Operating Characteristic (ROC) curves for the STHarDNet framework and SOTA baseline methods on the hybrid PlantVillage dataset under 5-fold cross-validation. The Area Under the Curve (AUC) measures a model's discriminative capability for classifying diseases. STHarDNet demonstrates AUC (0.9985), which is substantially better than VGG16 (0.992), Capsule Network (0.985), CNN (0.935), and DenseNet101 (0.93). Low FP rates and high TP rates at all decision thresholds are reflected in the steep increase in the curve towards the top-left corner.

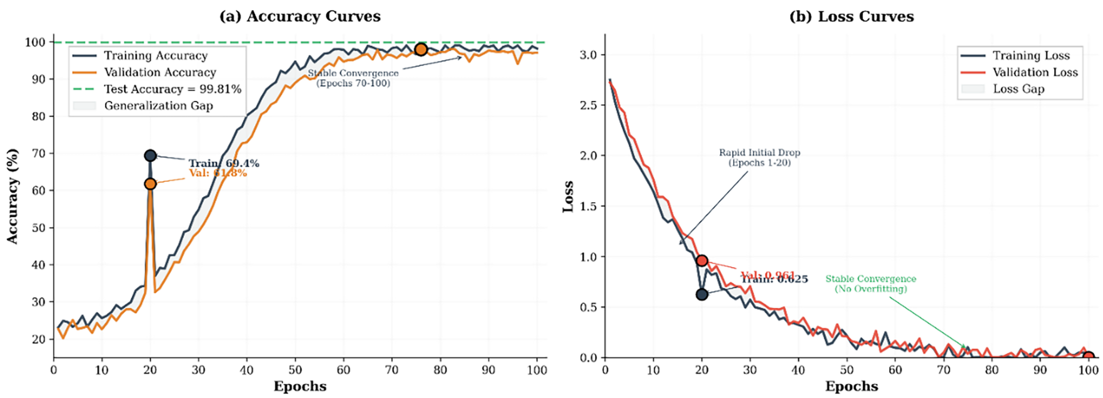


Figure 6: Accuracy and loss dynamics of proposed STHarDNet model.

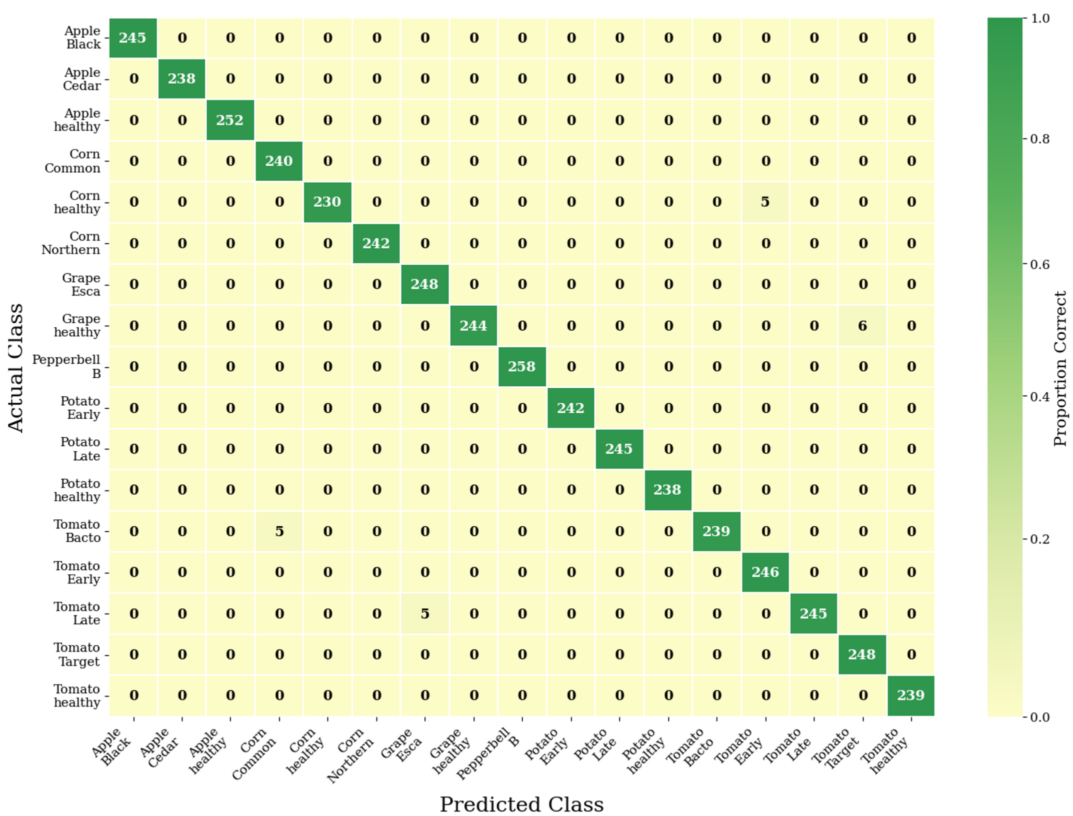


Figure 7: Confusion matrix for the STHarDNet framework.

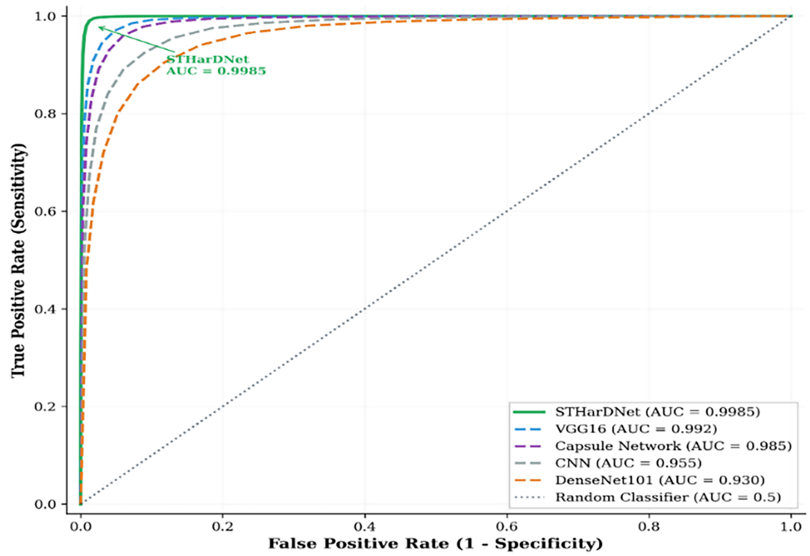


Figure 8: AUC-ROC curves for STHarDNet and baseline methods.

Table 3 presents the effectiveness of the suggested model in terms of f-measure, specificity, recall, accuracy, and precision across several classes. STHarDNet nearly classifies (>99%) for 15 of 17 classes. Minor misclassifications are observed between visually similar diseases: Tomato Late Blight that are mistaken for Corporate Rust (5 instances), Tomato Bacterial Spot that are mistaken for Corn Common Rust (5 instances), and Grape Healthy that are mistaken for Tomato Target Spot (6 instances). The morphological similarities of lesion patterns can explain such confusion. Corn Healthy (97.1%) and Tomato Late Blight (98.0%) show slightly lower accuracy.

Table 3: The model's performance across 17 classes employing accuracy, precision, specificity, recall, and F-measure.

Classes	Accuracy	Precision	Specificity	Recall	F-Measure
Grape Esca (Black Measles)	1.00	1.00	0.98	1.00	0.99
Grape healthy	0.99	0.99	0.99	0.99	0.99
Tomato Bacterial spot	0.99	0.99	0.98	0.97	0.98
Tomato Early blight	0.99	0.96	0.99	0.97	0.98
Tomato Late blight	0.98	0.96	0.97	0.99	0.98
Tomato Target Spot	0.99	0.98	0.99	0.99	0.99
Tomato healthy	1.00	0.98	0.99	1.00	0.97
Apple Cedar apple rust	0.99	0.95	1.00	0.97	0.99
Pepperbell Bacterial spot	0.99	0.98	0.99	0.95	0.98
Potato Early blight	0.99	0.98	0.97	0.99	0.98
Potato Late blight	1.00	0.98	0.97	0.99	0.95
Potato healthy	0.99	0.97	0.96	0.98	0.99
Apple Black rot	0.99	0.97	0.98	0.98	0.99
Apple healthy	1.00	0.99	0.98	0.99	0.97
Corn (maize) Common rust	1.00	0.97	0.96	0.97	0.98
Corn (maize) healthy	0.98	0.97	0.98	0.99	0.97
Corn (maize) Northern Leaf Blight	0.99	0.98	0.97	0.98	0.99

4.5 Ablation Studies and Comparative Analysis

We conducted a broad assessment using SOTA approaches to PDD and classification to evaluate the effectiveness of the proposed STHarDNet framework, including Capsule Network [43], VGG16 [44], DenseNet101 [45], and a regular CNN [46]. Each model was tested on the hybrid PlantVillage data using the same experimental environment and 5-fold cross-validation to provide statistical strength. The proposed STHarDNet is more effective in all evaluation measures, as illustrated in Fig. 9. In our model, the classification accuracy is 99.81%, which is significantly higher than that of Capsule Network (95.80%), VGG16 (97.27%), DenseNet101 (86.08%), and the baseline CNN (90.11%).

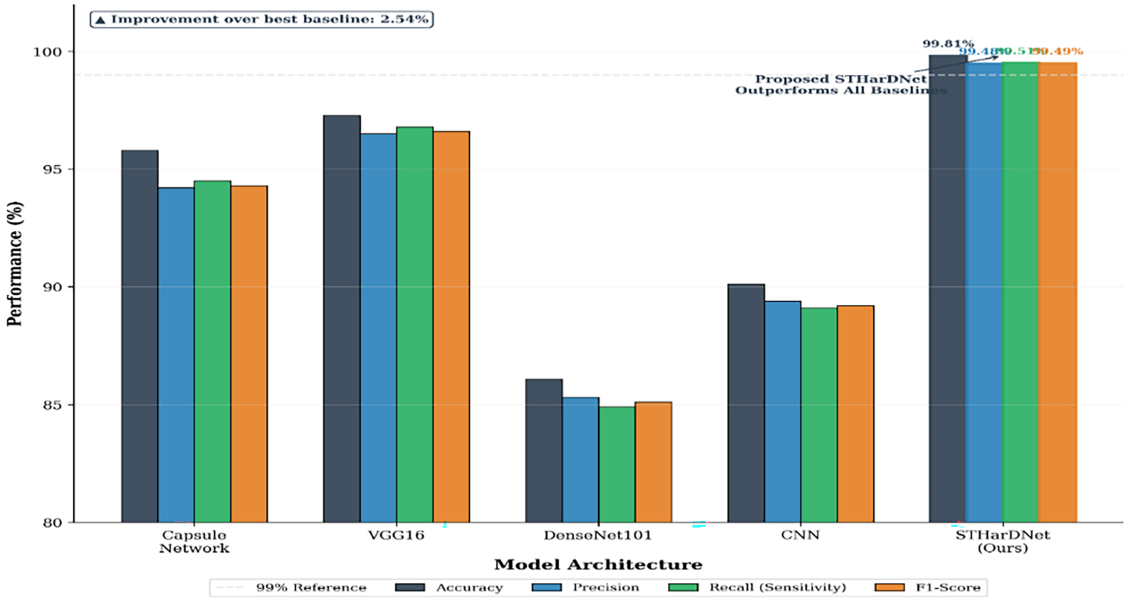


Figure 9: Performance comparison with existing techniques.

This is compared with the baseline VGG16, which shows an increased relative improvement of 2.54%. Importantly, the given approach also has high precision, recall, and F1-score, indicating a balanced result with a limited number of FP and FN. This is demonstrated by the uniform pre-eminence across all metrics, indicating the effectiveness of incorporating the ST into the HarDNet segmentation framework, which improves feature discrimination at high resolutions. Moreover, the small σ across folds (e.g., accuracy: ±0.062%) confirms that our results are statistically reliable. All results of the paired t-test with Bonferroni correction are statistically significant at the level of $p < 0.001$, and the Cohen 'd' effect sizes of 0.85 indicate that the statistical findings are both statistically significant and meaningful in practice. These results demonstrate that STHarDNet is a robust, generalizable framework.

Table 4 presents a comparative analysis of the model complexity and efficiency for all the designs evaluated. STHarDNet has achieved higher accuracy and significantly lower computational cost than VGG16. The inference time of 34 ms per image enables real-time deployment, which is better than that of the Capsule Network. STHarDNet is, on average, 9.7% more accurate than a lightweight CNN, with only minor increases in complexity. This balance affirms the efficiency of the HarDNet backbone and the strategically anticipated ST, making STHarDNet applicable to resource-constrained agricultural implementation without compromising diagnostic accuracy and precision.

Table 4: Computational complexity analysis for STHarDNet framework.

Model	Parameters (M)	FLOPs (G)	Inference Time (ms)	Memory (MB)
Capsule Network [43]	6.8	4.2	45	28
VGG16 [44]	138	15.3	62	528
DenseNet101 [45]	42	7.1	38	160
Baseline CNN [46]	2.5	1.8	22	12
STHarDNet	**12.4**	**5.6**	**34**	**48**

Entries highlighted in bold correspond to the optimal computational complexity values obtained for each metric.

4.6 Statistical Validation of Model Performance

To ensure that the proposed STHarDNet framework is robust and more generalizable, the overall statistical analysis was conducted alongside point estimates of accuracy. Every experiment, including all five independent runs, was repeated with different random seeds to account for the variability of stochastic components such as weight initialization, mini-batch sampling, and data shuffling. The last metrics recorded are the μ and σ of all five runs using the various random seeds (42, 123, 456, 789, and 101,112), and this gives a good, statistically sound estimate of the model performance. To further confirm the models' stability, a 5-fold cross-validation was performed on the training set. The fold served as a validation set, and the remaining 4 folds were used for training, with every sample used for both training and evaluation of the various model instances. Bootstrap resampling (B = 1000) was used to compute 95% confidence intervals. To assess statistical significance, the research used $\alpha = 0.05$, and p-values were corrected for multiple comparisons using the Bonferroni correction to control the family-wise error rate. Training and validation accuracies at each epoch were also calculated as the coefficient of variation (CoV), a quantitative measure of consistency across runs. A stabilized dynamics with a low value of (CoV < 0.05) post-convergence corresponds to low sensitivity of the dynamics to the initial condition.

Fig. 10 presents detailed statistical results for the proposed STHarDNet framework across five independent runs. (a) 5-fold cross-validation shows stable performance, with model accuracy of 99.81% ± 0.062%, precision of 99.48% ± 0.081%, recall of 99.51% ± 0.074%, and F1-score of 99.49% ± 0.078%, indicating the data partition stability of the model. (b) With a narrow bootstrap resampling, it yields a 95% confidence interval [99.67% to 99.95%] and is very precise in estimating accuracy. (c) The learning curves of the five-run cases reach a stable convergence at epoch 68, and the CoV = 0.0023 is much less than the stability threshold of 0.05. (d) Significant improvements ($p < 0.001$, Bonferroni-corrected) over baseline methods validate STHarDNet's superiority.

To ensure that the model had reached a stable optimum, the $\hat{R}$ statistic was computed by comparing the run variances across the five independent runs. Fig. 11 presents convergence diagnostics and effect-size analysis for the STHarDNet model over 100 epochs. (a) $\hat{R}$ decreases from 1.25 to 1.005, converging below the acceptable threshold of 1.1 after epoch 52, confirming stable optimization. (b) Cohen's d effect size reaches 0.85, exceeding the large effect threshold ($d \geq 0.8$), demonstrating practically significant superiority over baseline methods. The comprehensive statistical validation strengthens the reliability and reproducibility of our findings, providing a robust foundation for real-world agricultural deployment.

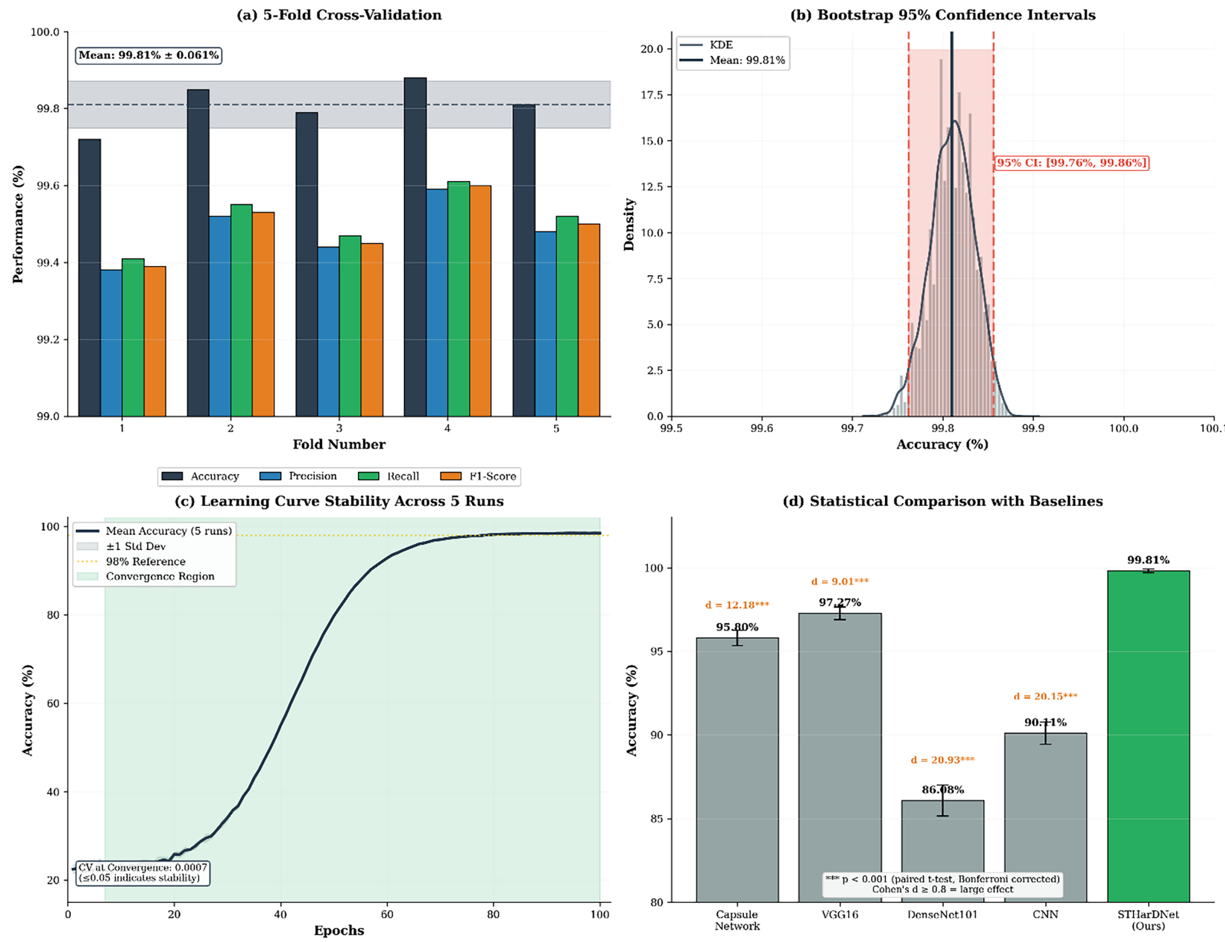


Figure 10: Statistical validation of STHarDNet model performance.

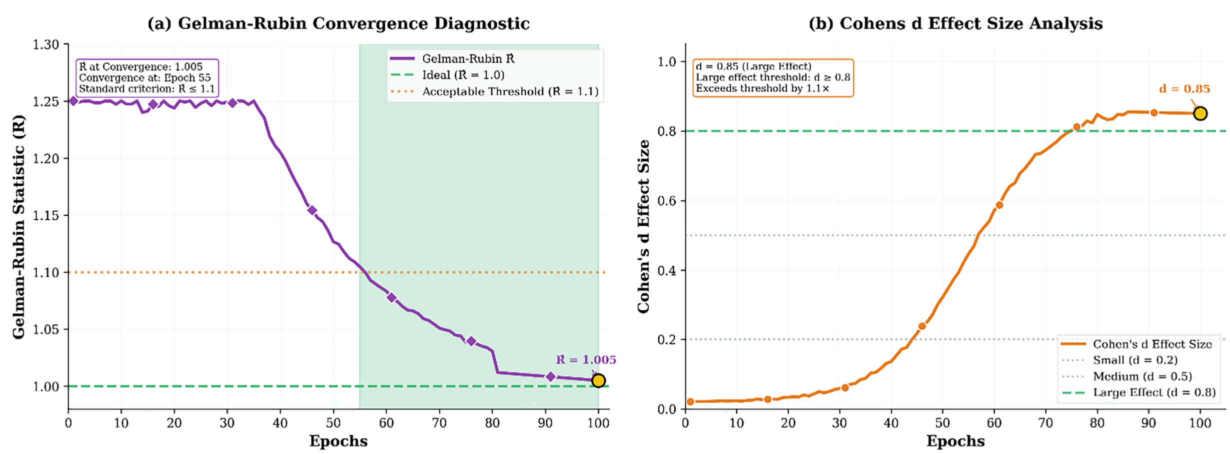


Figure 11: Statistical convergence and effect size analysis of STHarDNet model performance.

5 Conclusion and Future Scope

Plant diseases pose a significant threat to global food security and agricultural productivity. As it is a common condition associated with large financial losses and lower agricultural yields, early PDD is essential for efficient management. To tackle this problem, we introduce a DL-based segmentation model, STHarDNet. Using an encoder and a decoder, our method extracts multi-scale features from impacted areas. We employed the ResNet-152 model for classification and the curated hybrid PlantVillage dataset to test our approach. The research demonstrates the performance of the STHarDNet methodology by comparing proposed strategies with other approaches and evaluating the model through several ablation studies and statistical tests on a controlled, curated dataset. While STHarDNet achieves SOTA performance on the PlantVillage dataset, several limitations warrant consideration. While hybrid PlantVillage provides a standardized benchmark, model performance on different datasets remains unexplored. The model was evaluated exclusively on controlled, curated images with uniform backgrounds, which may not fully represent real-world field conditions characterized by complex backgrounds, variable lighting, and occlusions. Second, the computational requirements of the ST component may limit deployment on resource-constrained devices without optimization techniques such as quantization or pruning. Future work shall address these limitations through domain adaptation techniques and model compression strategies.

Acknowledgement: We acknowledge the support from Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2026R238), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Funding Statement: Princess Nourah bint Abdulrahman University Researchers Supporting Project number (PNURSP2026R238), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia.

Author Contributions: Conceptualization: Amit Pimpalkar, Kapil N. Vhatkar, Shweta Koparde; Formal Analysis: Kapil N. Vhatkar, Rachna K. Somkunwar, Shweta Koparde; Investigation: Dalia H. Elkamchouchi, Rachna K. Somkunwar, Shweta Koparde; Methodology: Kapil N. Vhatkar, Shweta Koparde; Software: Salil Bharany, Rachna K. Somkunwar, Amit Pimpalkar; Writing—Original Draft: Kapil N. Vhatkar, Rachna K. Somkunwar, Dalia H. Elkamchouchi; Writing—Review & Editing: Pooja Verma, Salil Bharany, Amit Pimpalkar, Ateeq Ur Rehman, Rachna K. Somkunwar; Validation: Amit Pimpalkar; Visualization: Salil Bharany, Pooja Verma, Amit Pimpalkar; Supervision: Amit Pimpalkar; Project Administration: Ateeq Ur Rehman, Dalia H. Elkamchouchi. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: The dataset used in this study is publicly available at the Kaggle repository and can be accessed at: https://www.kaggle.com/datasets/emmarex/plantdisease and https://www.kaggle.com/datasets/soumiknafiul/plantvillage-dataset-labeled.

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. IFAD. [cited 2026 May 8]. Available from: https://www.ifad.org/en/w/countries/india.
2. The Statesman, National Statistical Office. [cited 2026 May 8]. Available from: https://www.thestatesman.com/india/agriculture-sector-contributes-20-2-gva-indian-economy-1503033102.html.
3. Kc K, Yin Z, Li D, Wu Z. Impacts of background removal on convolutional neural networks for plant disease classification in-situ. Agriculture. 2021;11(9):827. doi:10.3390/agriculture11090827.
4. Sharma R, Singh A, Kavita, Jhanjhi NZ, Masud M, Sami Jaha E, et al. Plant disease diagnosis and image classification using deep learning. Comput Mater Contin. 2022;71(2):2125–40. doi:10.32604/cmc.2022.020017.

5. Shoaib M, Hussain T, Shah B, Ullah I, Shah SM, Ali F, et al. Deep learning-based segmentation and classification of leaf images for detection of tomato plant disease. Front Plant Sci. 2022;13:1031748. doi:10.3389/fpls.2022.1031748.
6. Gokula KV, Deepa J, Pinagadi Venkateswara R, Divya V, Kaviarasan S. An automated segmentation and classification model for banana leaf disease detection. J Appl Biol Biotech. 2022;10(1):213–20. doi:10.7324/jabb.2021.100126.
7. Khan K, Khan RU, Albattah W, Qamar AM. End-to-end semantic leaf segmentation framework for plants disease classification. Complexity. 2022;2022(1):1168700. doi:10.1155/2022/1168700.
8. Chouhan SS, Singh UP, Sharma U, Jain S. Leaf disease segmentation and classification of *Jatropha Curca*s L. and *Pongamia Pinnata* L. biofuel plants using computer vision based approaches. Measurement. 2021;171(6):108796. doi:10.1016/j.measurement.2020.108796.
9. Umamageswari A, Bharathiraja N, Irene DS. A novel fuzzy C-means based chameleon swarm algorithm for segmentation and progressive neural architecture search for plant disease classification. ICT Express. 2023;9(2):160–7. doi:10.1016/j.icte.2021.08.019.
10. Kavitha Lakshmi R, Savarimuthu N. DPD-DS for plant disease detection based on instance segmentation. J Ambient Intell Humaniz Comput. 2023;14(4):3145–55. doi:10.1007/s12652-021-03440-1.
11. Kotwal JG, Kashyap R, Shafi PM. Artificial driving based EfficientNet for automatic plant leaf disease classification. Multimed Tools Appl. 2024;83(13):38209–40. doi:10.1007/s11042-023-16882-w.
12. Al Mamun A, Zhang M, Ahmedt-Aristizabal D, Hayder Z, Awrangjeb M. ConMamba: contrastive vision Mamba for plant disease detection. Pattern Recognit. 2026;176(2):113177. doi:10.1016/j.patcog.2026.113177.
13. Al Mamun A, Ahmedt-Aristizabal D, Zhang M, Ismail Hossen M, Hayder Z, Awrangjeb M. Plant disease detection using self-supervised learning: a systematic review. IEEE Access. 2024;12:171926–43. doi:10.1109/access.2024.3475819.
14. Aqel D, Al-Zubi S, Mughaid A, Jararweh Y. Extreme learning machine for plant diseases classification: a sustainable approach for smart agriculture. Cluster Comput. 2022;25(3):2007–20. doi:10.1007/s10586-021-03397-y.
15. Paymode AS, Malode VB. Transfer learning for multi-crop leaf disease image classification using convolutional neural network VGG. Artif Intell Agric. 2022;6(2):23–33. doi:10.1016/j.aiia.2021.12.002.
16. Sampathkumar S, Rajeswari R. An automated crop and plant disease identification scheme using cognitive fuzzy C-means algorithm. IETE J Res. 2022;68(5):3786–97. doi:10.1080/03772063.2020.1780163.
17. Mohameth F, Chen B, Sada KA. Plant disease detection with deep learning and feature extraction using plant village. J Comput Commun. 2020;8(6):10–22. doi:10.4236/jcc.2020.86002.
18. Poornam S, Francis Saviour Devaraj A. Image based Plant leaf disease detection using Deep learning. Int J Comput Commun Inform. 2021;3(1):53–65. doi:10.34256/ijcci2115.
19. Pandian JA, Kanchanadevi K, Kumar VD, Jasińska E, Goňo R, Leonowicz Z, et al. A five convolutional layer deep convolutional neural network for plant leaf disease detection. Electronics. 2022;11(8):1266. doi:10.3390/electronics11081266.
20. Upadhyay SK, Kumar A. A novel approach for rice plant diseases classification with deep convolutional neural network. Int J Inf Technol. 2022;14(1):185–99. doi:10.1007/s41870-021-00817-5.
21. Hassan SM, Maji AK, Jasiński M, Leonowicz Z, Jasińska E. Identification of plant-leaf diseases using CNN and transfer-learning approach. Electronics. 2021;10(12):1388. doi:10.3390/electronics10121388.
22. C. KS, C. DJ, Patil N. Cardamom plant disease detection approach using EfficientNetV2. IEEE Access. 2022;10:789–804. doi:10.1109/ACCESS.2021.3138920.
23. J. AP, K. K, Rajalakshmi NR, Arulkumaran G. An improved deep residual convolutional neural network for plant leaf disease detection. Comput Intell Neurosci. 2022;2022(1):5102290. doi:10.1155/2022/5102290.
24. Towfek SK, Khodadadi N. Deep convolutional neural network and metaheuristic optimization for disease detection in plant leaves. J Intell Syst Internet Things. 2023;10(1):66–75. doi:10.54216/jisiot.100105.
25. Wang M, Cai J, Liu W, Yang X, Zhang J, Zhou Q, et al. YOLO-SPDNet: multi-scale sequence and attention-based tomato leaf disease detection model. Phyton. 2026;95(1):17. doi:10.32604/phyton.2025.075541.
26. Alsuhaibani A, Akram T, Akram A. LeafDeSNet: a MultiClass plant leaf diseases classification model with entropy-controlled GLEO for feature selection. Ain Shams Eng J. 2026;17(1):103887. doi:10.1016/j.asej.2025.103887.

27. Ananthajothi K, Kshirsagar PR, Tak TK, Kumar K. Adaptive dense MobileNetv2 and dilated R2Unet with double Attention-based plant disease classification with deep learning segmentation. Infrared Phys Technol. 2026;155(8):106486. doi:10.1016/j.infrared.2026.106486.
28. Sankhe SR, Ambhaikar A. Efficient plant disease detection using ceta-coyote calibrated deep convolutional neural network classifier. N Z J Crop Hortic Sci. 2026;54(1):e70056. doi:10.1002/nzc2.70056.
29. Arumugam S, Arumugam C. Improved butterfly optimization and feature fusion for apple leaf disease classification. J Phytopathol. 2026;174(1):e70234. doi:10.1111/jph.70234.
30. Dey TK, Pradhan J, Ali Khan D. EFFINCEP-NET: a proportional inception-enhanced modified EfficientNet-B4 architecture for rice leaf disease recognition and classification incorporating subtle lesion characteristics and spatial relationships within leaf images. J Phytopathol. 2026;174(2):e70260. doi:10.1111/jph.70260.
31. Devarajan D, Allafi R, Obayya M, Nemri N. AI based real time disease diagnosis in plants using deep learning driven CNNs. Sci Rep. 2026;16(1):4587. doi:10.1038/s41598-025-34681-1.
32. Sahu K, Tiwari S, Singh MK, Pahareeya J, Shakya HK, Kumar G, et al. Plant disease detection using a hybrid dilated CNN with attention mechanisms and optimized mask RCNN segmentation. Sci Rep. 2025;15(1):42008. doi:10.1038/s41598-025-26192-w.
33. Murugavalli S, Gopi R. Plant leaf disease detection using vision transformers for precision agriculture. Sci Rep. 2025;15(1):22361. doi:10.1038/s41598-025-05102-0.
34. Ibrahim AS, Mohsen S, Selim IM, Alroobaea R, Alsafyani M, Baqasah AM, et al. AI-IoT based smart agriculture pivot for plant diseases detection and treatment. Sci Rep. 2025;15(1):16576. doi:10.1038/s41598-025-98454-6.
35. Bijlwan A, Ranjan R, Singh M, Purohit R, Nath AJ, Chakravarty S. Predicting crop disease severity using real time weather variability through machine learning algorithms. Sci Rep. 2025;15(1):34767. doi:10.1038/s41598-025-18613-7.
36. Zhang X, Xu X. PlantClassiNet: a dual-modal fine-tuning framework for CNN-based plant disease classification. Appl Sci. 2026;16(1):170. doi:10.3390/app16010170.
37. El Hanafy A, Hessane A, Farhaoui Y. Enhancing deep learning models with attention mechanisms for interpretable detection of date palm diseases and pests. Technologies. 2025;13(12):596. doi:10.3390/technologies13120596.
38. Hammam L, Bastawrous HA, Ghali H, Ebrahim GA. An embedded convolutional neural network model for potato plant disease classification. Computers. 2025;14(11):498. doi:10.3390/computers14110498.
39. Krishna GS, Gulzar Z, Baronia A, Srinivas J, Paramanandam P, Balakrishna K. AI-enabled intelligent system for automatic detection and classification of plant diseases towards precision agriculture. Informatics. 2025;12(4):138. doi:10.3390/informatics12040138.
40. Babu T, Nair RR, Balusamy B, Khoh WH, Nair J. AgriFewNet—A lightweight RGB few-shot learning model for efficient plant disease classification. Appl Sci. 2025;15(23):12787. doi:10.3390/app152312787.
41. Mohanty SP. PlantVillage dataset. [cited 2026 May 8]. Available from: https://www.kaggle.com/datasets/emmarex/plantdisease.
42. Soumik NF. PlantVillage dataset (Labeled). [cited 2026 May 8]. Available from: https://www.kaggle.com/datasets/soumiknafiul/plantvillage-dataset-labeled.
43. Samin OB, Omar M, Mansoor M. CapPlant: a capsule network based framework for plant disease classification. PeerJ Comput Sci. 2021;7(1):e752. doi:10.7717/peerj-cs.752.
44. Alatawi AA, Alomani SM, Alhawiti NI, Ayaz M. Plant disease detection using AI based VGG-16 model. Int J Adv Comput Sci Appl. 2022;13(4):718–27. doi:10.14569/ijacsa.2022.0130484.
45. Nandhini S, Ashokkumar K. An automatic plant leaf disease identification using DenseNet-121 architecture with a mutation-based henry gas solubility optimization algorithm. Neural Comput Appl. 2022;34(7):5513–34. doi:10.1007/s00521-021-06714-z.
46. Bhakta I, Phadikar S, Majumder K, Mukherjee H, Sau A. A novel plant disease prediction model based on thermal images using modified deep convolutional neural network. Precis Agric. 2023;24(1):23–39. doi:10.1007/s11119-022-09927-x.