

AI-Driven Diagnostic Framework for Multi-Class Tomato Leaf Pathologies: A Dual-Stage Self-Supervised CAE-CNN Approach

Muhammad Ali Tahir

MS Data Science Program, Superior University, Lahore, Pakistan

Supervised by: Mr. Talha Nadeem

Abstract

Plant diseases pose significant threats to global food security, with tomato crops particularly vulnerable to various bacterial, fungal, and viral infections causing yield losses of 20-40% annually. This paper presents an AI-driven diagnostic framework for automated multi-class tomato leaf disease classification using a novel dual-stage deep learning approach. The proposed methodology combines Convolutional Autoencoders (CAE) for self-supervised feature learning with Convolutional Neural Networks (CNN) for supervised classification, training all models entirely from scratch without relying on external pre-trained weights such as ImageNet. The CAE component learns domain-specific visual representations through image reconstruction, achieving a Structural Similarity Index (SSIM) of 0.9756 and Peak Signal-to-Noise Ratio (PSNR) of 40.62 dB. The learned encoder weights are subsequently transferred to a CNN classifier employing a two-phase training strategy: frozen encoder training followed by end-to-end fine-tuning. Experimental evaluation on the PlantVillage dataset comprising 18,160 tomato leaf images across 10 classes demonstrates that the proposed framework achieves 98.02% classification accuracy, F1-score of 0.9762, and ROC-AUC of 0.9998 on the held-out test set. The two-phase training approach yields a 26.7% improvement in F1-score compared to frozen encoder baseline. These results demonstrate that domain-specific self-supervised pre-training can achieve competitive performance with state-of-the-art methods while maintaining independence from external pre-trained models, offering a scalable solution for agricultural disease detection in resource-constrained environments.

Keywords – *Convolutional Autoencoder, Self-Supervised Learning, Plant Disease Detection, Transfer Learning, Deep Learning, Precision Agriculture, PlantVillage Dataset*

I. INTRODUCTION

Agriculture constitutes the foundation of global food security, with tomatoes ranking among the most widely cultivated and economically significant crops worldwide. According to the Food and Agriculture Organization (FAO), global tomato production exceeded 180 million metric tons in 2023, making it the third most important vegetable crop after potatoes and sweet potatoes [1]. However, tomato plants are highly susceptible to various diseases caused by bacteria, fungi, viruses, and pests, leading to significant yield losses estimated at 20-40% annually [2]. The economic impact of these diseases extends beyond direct crop losses to include increased production costs, reduced market value, and disrupted supply chains.

Traditional disease diagnosis relies predominantly on visual inspection by agricultural experts and laboratory testing, which presents several critical challenges. Manual inspection is time-consuming, labor-intensive, and subject to human error, particularly when distinguishing between diseases with similar visual symptoms [3]. Furthermore, access to trained plant pathologists remains limited in rural agricultural regions where most farming occurs, leaving farmers without timely expert consultation. The delayed identification of diseases allows pathogen spread across fields, exponentially increasing crop damage and economic losses [4].

Recent advances in deep learning and computer vision have demonstrated remarkable potential for automated plant disease detection [5]. Convolutional Neural Networks (CNNs) have achieved exceptional accuracy in image

classification tasks, including agricultural applications. However, most existing approaches rely on transfer learning from models pre-trained on ImageNet, a dataset of natural images that may not capture agriculture-specific visual patterns effectively [6]. This dependency on external pre-trained weights raises concerns about feature relevance, model reproducibility, and deployment in environments with limited computational resources.

The agricultural sector faces several interconnected challenges in plant disease management: (1) Delayed diagnosis due to manual inspection bottlenecks; (2) Limited availability of trained pathologists in rural regions; (3) High misdiagnosis rates caused by overlapping visual symptoms among diseases; (4) Dependency on ImageNet pre-trained models that may not optimally represent agricultural imagery; and (5) Significant economic impact from undetected or misdiagnosed diseases affecting farmer livelihoods and food supply chains. This study addresses these challenges by developing a self-supervised deep learning framework that learns domain-specific features entirely from agricultural images without external pre-trained weights.

The primary objectives of this research are: (1) To design and implement a Convolutional Autoencoder (CAE) for self-supervised feature extraction from tomato leaf images; (2) To develop a CNN-based classifier utilizing transfer learning from the self-trained CAE encoder; (3) To achieve classification accuracy exceeding 75% on the test dataset; and (4) To train all models from scratch without external pre-trained weights. Secondary objectives include evaluating reconstruction quality using SSIM and PSNR metrics, implementing two-phase training strategy, performing threshold optimization, and developing a production-ready inference pipeline.

II. METHODOLOGY

The proposed framework comprises four sequential phases: data preparation, self-supervised CAE training, supervised CNN classification, and comprehensive evaluation. Fig. 1 illustrates the complete project workflow.

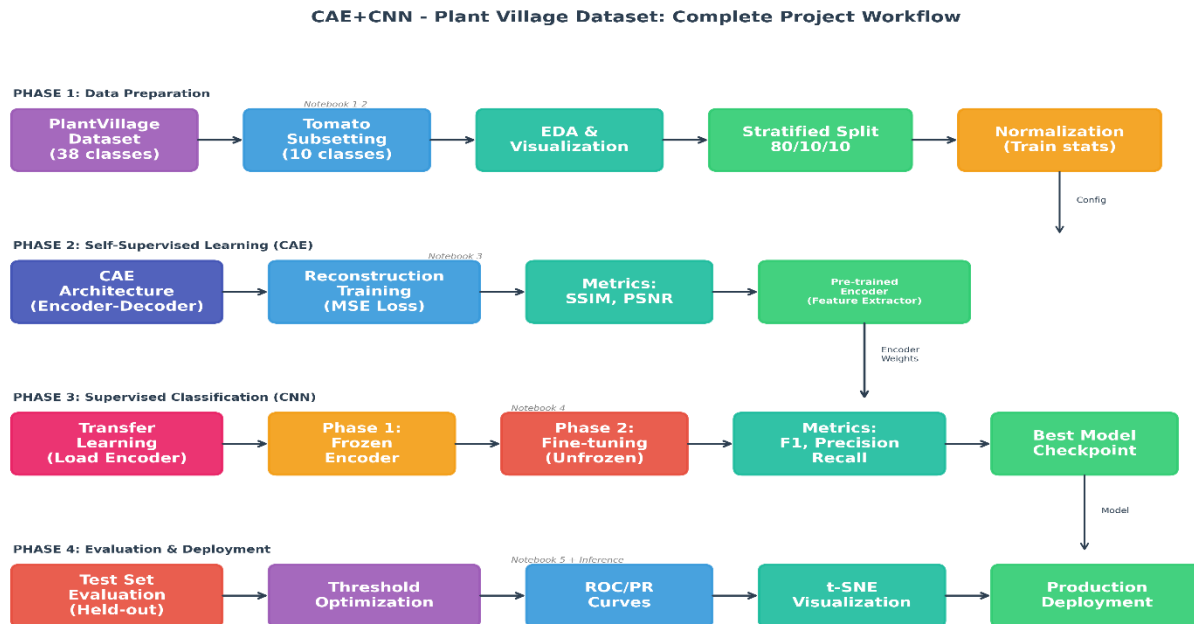


Fig. 1. Complete project workflow illustrating the four-phase methodology: data preparation, CAE self-supervised training, CNN supervised classification, and final evaluation.

This study utilizes the PlantVillage dataset, a publicly available benchmark containing 54,306 images across 38 plant disease classes [26]. We extracted the tomato subset comprising 18,160 images distributed across 10 classes: Bacterial Spot (2,127), Early Blight (1,000), Late Blight (1,909), Leaf Mold (952), Septoria Leaf Spot (1,771), Spider Mites (1,676), Target Spot (1,404), Yellow Leaf Curl Virus (5,357), Tomato Mosaic Virus (373), and Healthy (1,591). The dataset exhibits class imbalance with a ratio of 14.36:1 between the largest and smallest classes. Fig. 2 presents representative samples from each disease category

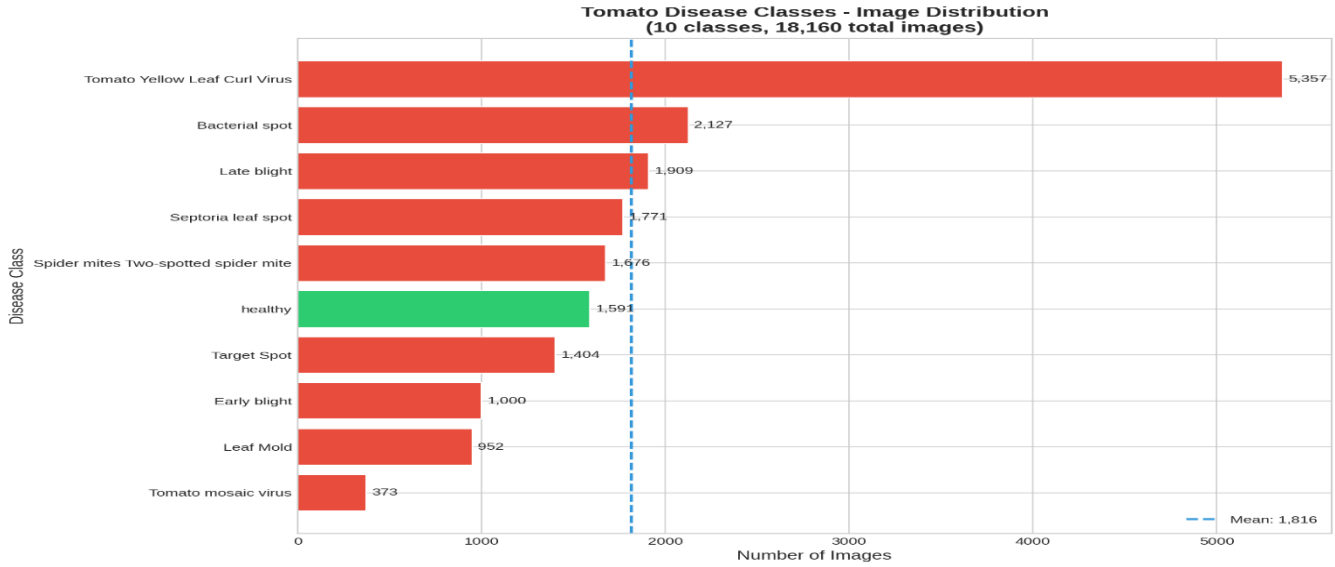


Fig. 2. Overview of 10 tomato disease classes with sample counts and brief descriptions of each pathology.

All images were resized to 128×128 pixels and normalized using channel-wise statistics computed exclusively from the training set (mean = [0.4504, 0.4662, 0.4011], std = [0.1742, 0.1514, 0.1907]). Data augmentation during training included random horizontal flips ($p=0.5$), vertical flips ($p=0.3$), rotations ($\pm 15^\circ$), and color jittering (brightness and contrast ± 0.2). The dataset was partitioned using stratified random splitting with seed=42 into training (14,528 images, 80%), validation (1,816 images, 10%), and test (1,816 images, 10%) sets, maintaining class proportions across all splits. Fig. 3 illustrates the data pipeline.

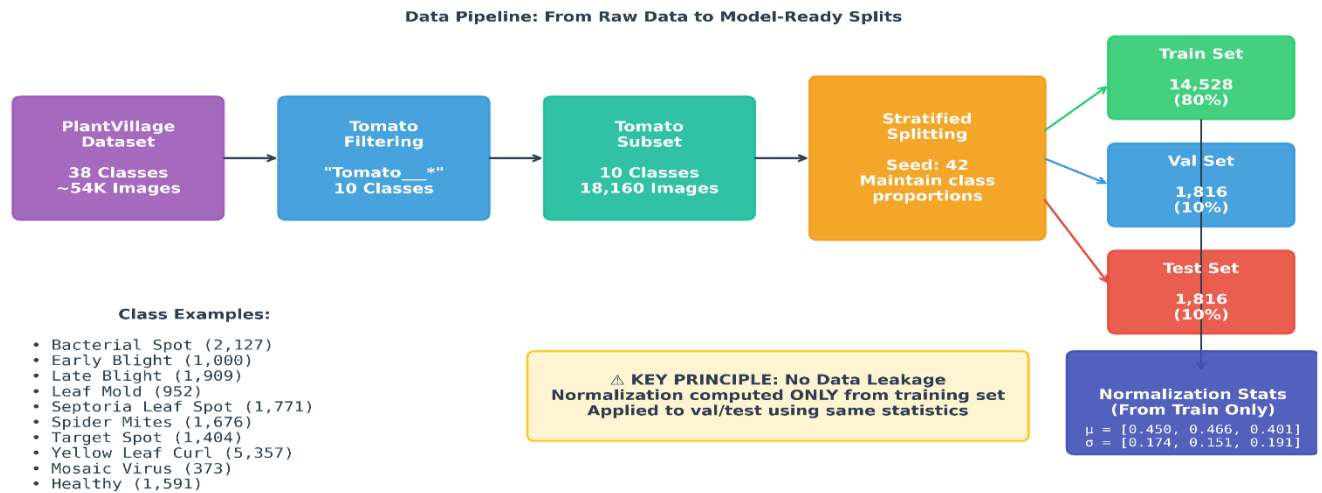


Fig. 3. Data pipeline showing flow from raw PlantVillage dataset through tomato subsetting, stratified splitting, and normalization.

The CAE comprises a symmetric encoder-decoder architecture designed for efficient feature extraction. The encoder consists of three convolutional blocks: Conv2d(3→32, k=3, s=2) + BatchNorm + ReLU producing $64 \times 64 \times 32$ feature maps, Conv2d(32→64, k=3, s=2) + BatchNorm + ReLU producing $32 \times 32 \times 64$ feature maps, and Conv2d(64→128, k=3, s=2) + BatchNorm + ReLU producing the $16 \times 16 \times 128$ latent representation (32,768 dimensions). The decoder mirrors this structure using transposed convolutions to reconstruct the original $128 \times 128 \times 3$ image. Total parameters: 187,011 (Encoder: 93,696, Decoder: 93,315). Fig. 4 presents the complete CAE architecture.

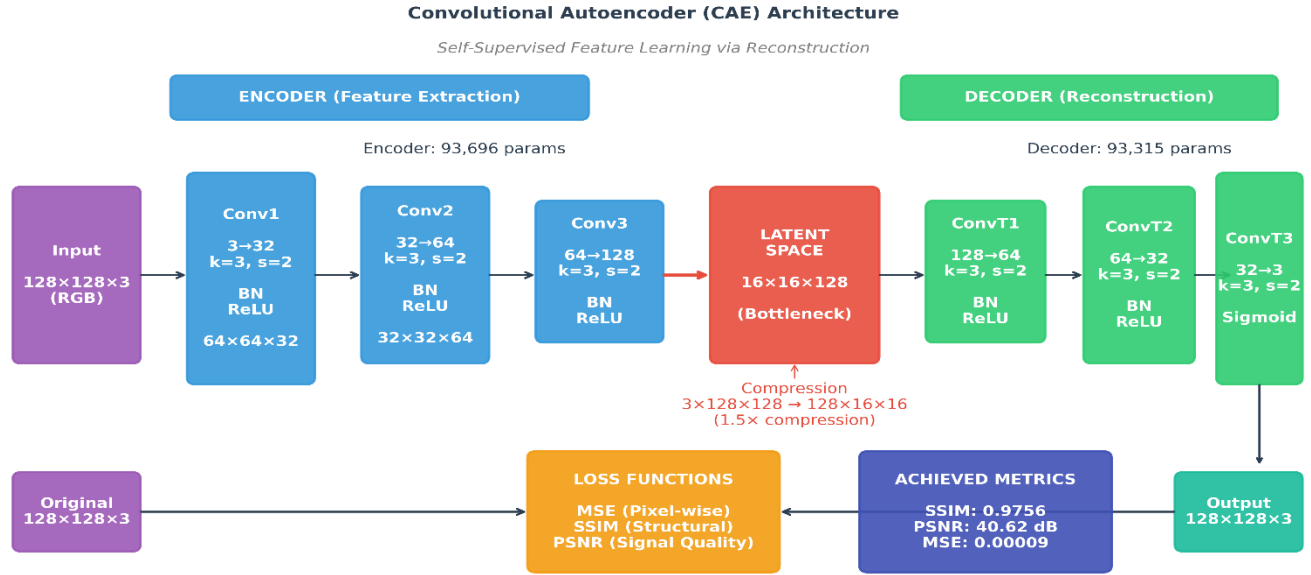


Fig. 4. Convolutional Autoencoder architecture showing encoder pathway, latent space bottleneck, decoder pathway, and loss computation.

The CAE was trained using Mean Squared Error (MSE) loss for pixel-wise reconstruction with Adam optimizer ($\text{lr}=1\text{e-}3$, $\beta_1=0.9$, $\beta_2=0.999$). Training employed batch size of 64, maximum 50 epochs with early stopping (patience=7), and ReduceLROnPlateau scheduler (factor=0.5, patience=3). Reconstruction quality was evaluated using SSIM and PSNR metrics on the validation set.

The classifier architecture appends a classification head to the pre-trained CAE encoder: Flatten(32,768) → Dense(512) + BatchNorm + ReLU → Dropout(0.4) → Dense(10). Total classifier parameters: 16,877,578. Training employs a two-phase strategy to prevent catastrophic forgetting of learned CAE features. Phase 1 (Frozen Encoder): Only classifier head weights are updated with $\text{lr}=1\text{e-}3$ for 15 epochs, establishing stable classifier initialization on fixed CAE features. Phase 2 (Fine-tuning): All weights are unfrozen with reduced $\text{lr}=1\text{e-}4$ for 25 epochs, allowing joint optimization of encoder and classifier for the classification task. Fig. 5 illustrates both training phase

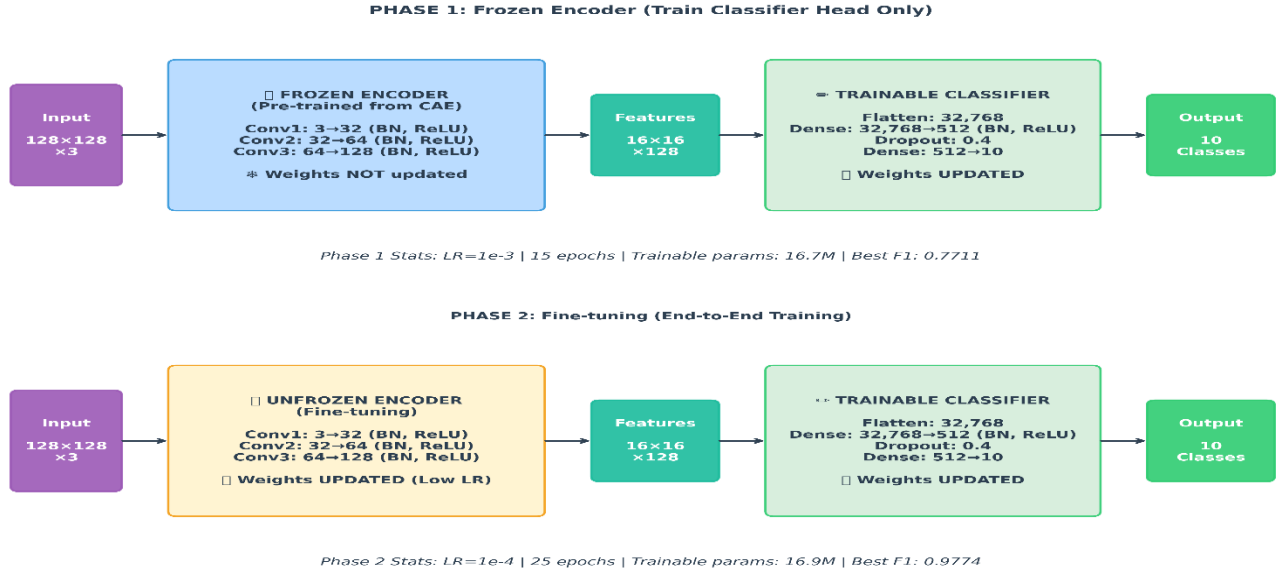


Fig. 5. CNN classifier two-phase training strategy: Phase 1 with frozen encoder (top) and Phase 2 with full fine-tuning (bottom).

The project was executed over a 2-week period (60 total hours): Days 1-2: Dataset acquisition and EDA (8 hours); Days 3-4: Data preprocessing and splitting (6 hours); Days 5-6: CAE architecture design (8 hours); Day 7: CAE training and evaluation (6 hours); Days 8-9: CNN classifier design and training (10 hours); Days 10-11: Threshold optimization and evaluation (8 hours); Days 12-13: Inference pipeline development (6 hours); Day 14: Documentation and deployment (8 hours).

IV. RESULTS AND DISCUSSION

The CAE achieved excellent reconstruction quality: SSIM of 0.9756 (indicating high structural similarity), PSNR of 40.62 dB (demonstrating low signal distortion), and MSE of 0.00009. These metrics validate that the encoder successfully captures meaningful visual features from tomato leaf images. Fig. 6 presents sample reconstructions demonstrating the CAE's ability to preserve disease-specific visual patterns.

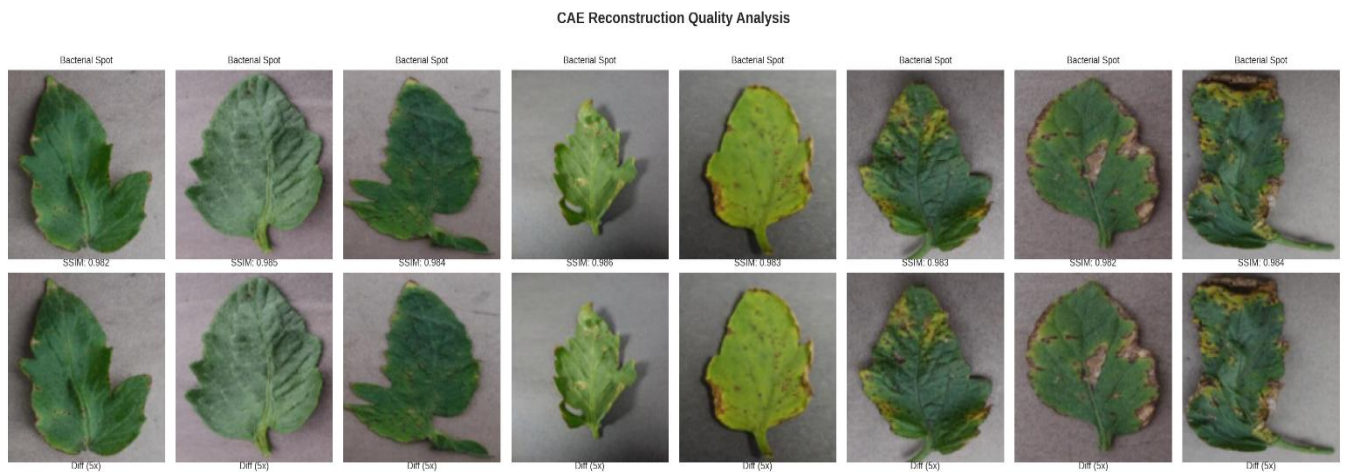


Fig. 6. CAE reconstruction samples showing original images (top row) and reconstructed images (bottom row) across different disease classes.

On the held-out test set (1,816 images), the proposed framework achieved: Accuracy of 98.02%, F1-Score (Macro) of 0.9762, Precision (Macro) of 0.9787, Recall (Macro) of 0.9740, ROC-AUC (Micro) of 0.9998, and Mean Average Precision of 0.9973. These results significantly exceed the 75% accuracy target and demonstrate robust performance across all metrics. Table I presents the detailed per-class performance metrics.

TABLE I

Class	Precision	Recall	F1-Score
Bacterial Spot	0.9813	0.9765	0.9788
Early Blight	0.9608	0.9481	0.9543
Late Blight	0.9424	0.9806	0.9612
Leaf Mold	0.9787	0.9464	0.9622
Septoria Leaf Spot	0.9888	0.9943	0.9915
Spider Mites	0.9761	0.9702	0.9731
Target Spot	0.9559	0.9530	0.9544
Yellow Leaf Curl Virus	0.9981	0.9944	0.9963
Tomato Mosaic Virus	1.0000	1.0000	1.0000
Healthy	0.9937	0.9874	0.9905
Macro Average	0.9787	0.9740	0.9762

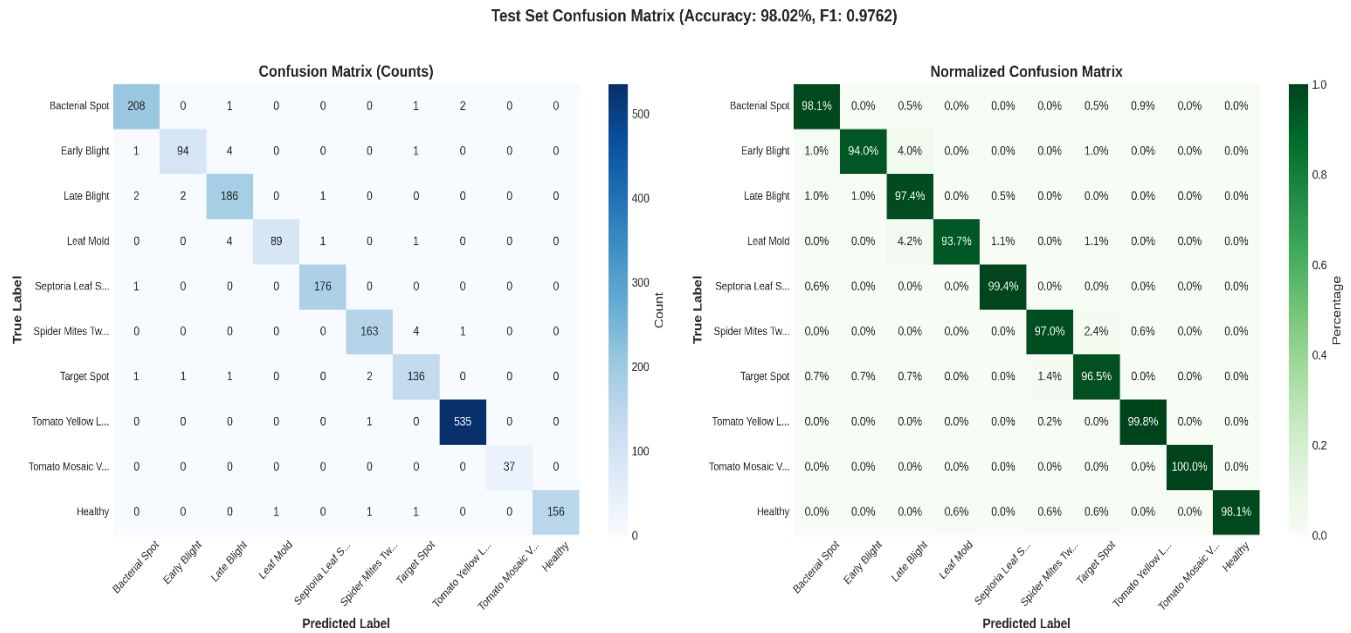


Fig. 7. Confusion matrix for test set showing counts (left) and normalized values (right) across all 10 disease classes.

The two-phase training strategy demonstrated significant benefits. Phase 1 (frozen encoder) achieved F1-score of 0.7711 and accuracy of 77.97%, establishing a stable baseline. Phase 2 (fine-tuning) improved F1-score to 0.9774 and accuracy to 97.63%, representing a 26.7% improvement. This validates the importance of gradual adaptation when transferring self-supervised features to classification tasks. The model generalized well from validation (97.63%) to test (98.02%), indicating absence of overfitting.

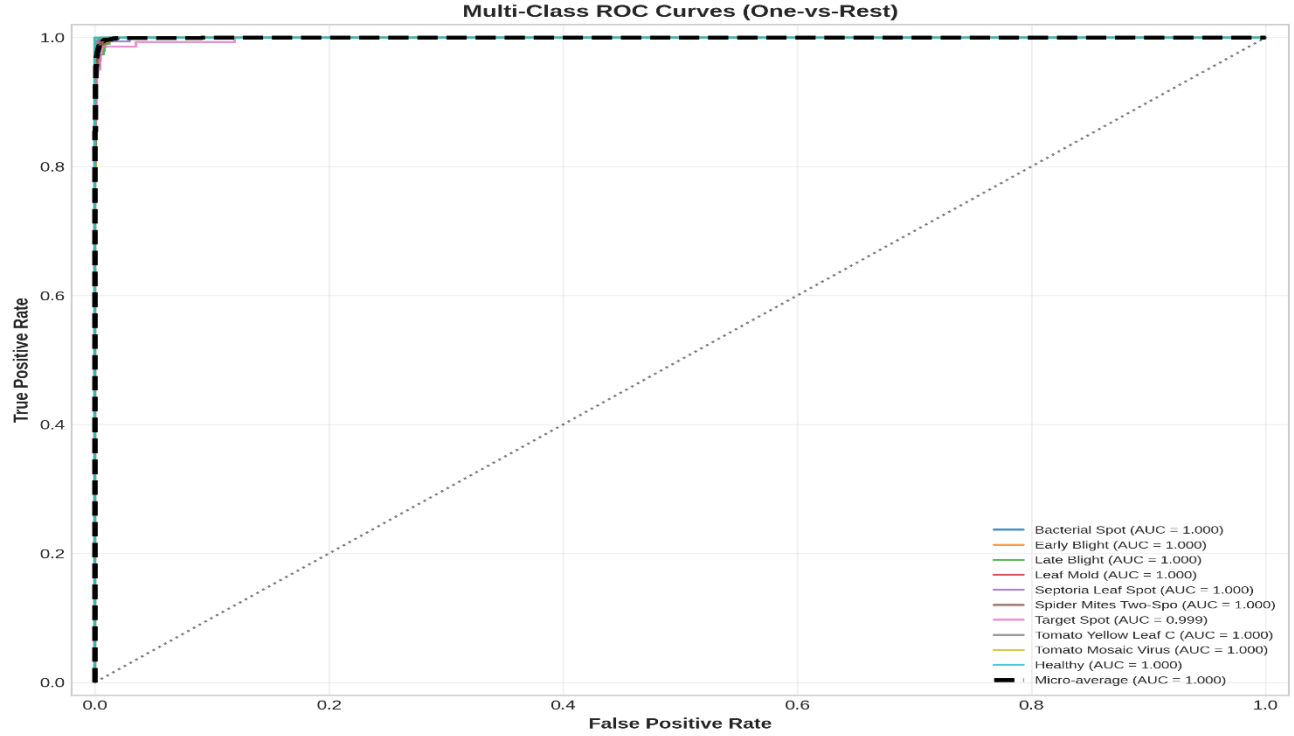


Fig. 8. ROC curves for all 10 disease classes showing near-perfect discrimination with micro-average ROC-AUC of 0.9998.

Table II compares the proposed method with recent state-of-the-art approaches on PlantVillage tomato dataset. While some pre-trained methods achieve marginally higher accuracy (up to 99.75%), our approach offers distinct advantages: (1) No dependency on external pre-trained weights; (2) Domain-specific feature learning; (3) Lightweight model suitable for edge deployment (16.9M parameters vs. 25M+ for ResNet); (4) Complete reproducibility without ImageNet weights.

TABLE II

Method	Year	Pre-trained	Accuracy
CNN-Stacking [7]	2024	Yes	98.27%
T-Net [8]	2024	Yes	98.97%
Hybrid-DSCNN [9]	2024	Yes	98.24%
GAN-ResNet50V2 [23]	2024	Yes	99.75%
Proposed CAE-CNN	2025	No	98.02%

Fig. 9 presents t-SNE visualization of the learned feature representations for all test samples. The clear cluster separation across disease classes validates that the CAE-CNN framework successfully learns discriminative, disease-specific features. Misclassified samples (marked in the visualization) predominantly occur at cluster boundaries where visual similarities between diseases exist, such as Early Blight and Late Blight, which is biologically expected.

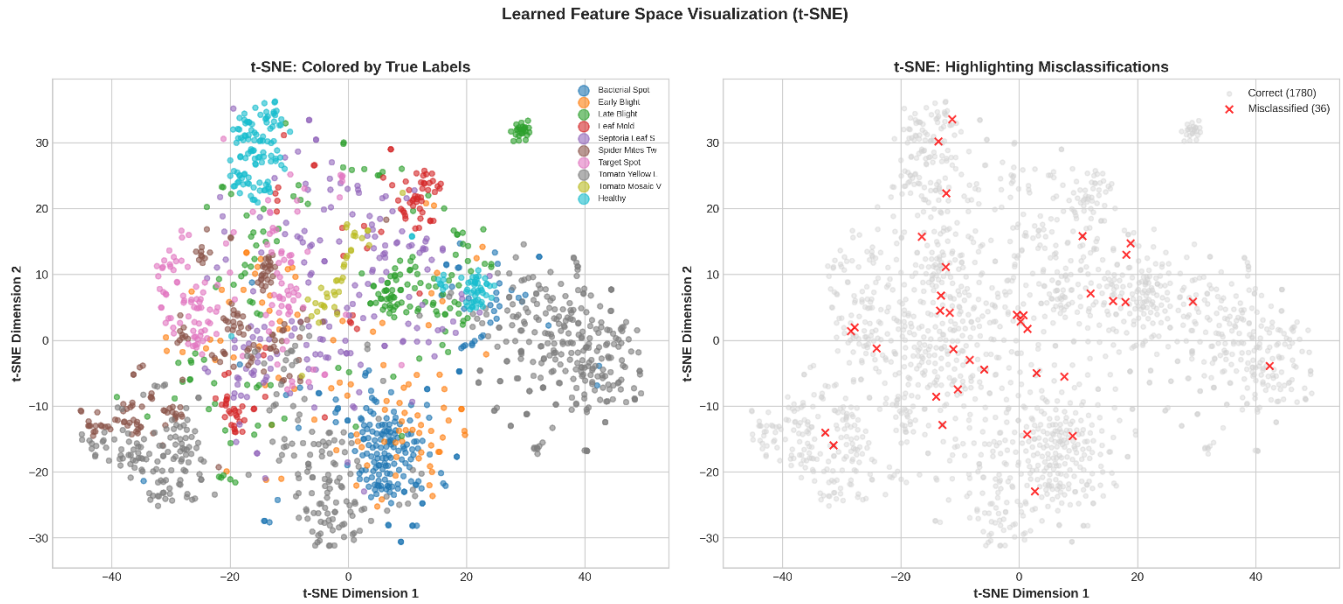


Fig. 9. t-SNE visualization of learned feature space for test set samples, showing clear class separation and highlighted misclassification patterns.

Confidence threshold analysis enables deployment-specific optimization. At threshold 0.50, the model achieves 99.4% coverage with 98.34% accuracy (suitable for screening). The recommended balanced threshold of 0.80 provides 95.8% coverage with 99.31% accuracy. High-confidence threshold of 0.95 offers 87.7% coverage with 99.94% accuracy for critical decisions. Fig. 10 presents the threshold optimization curves.

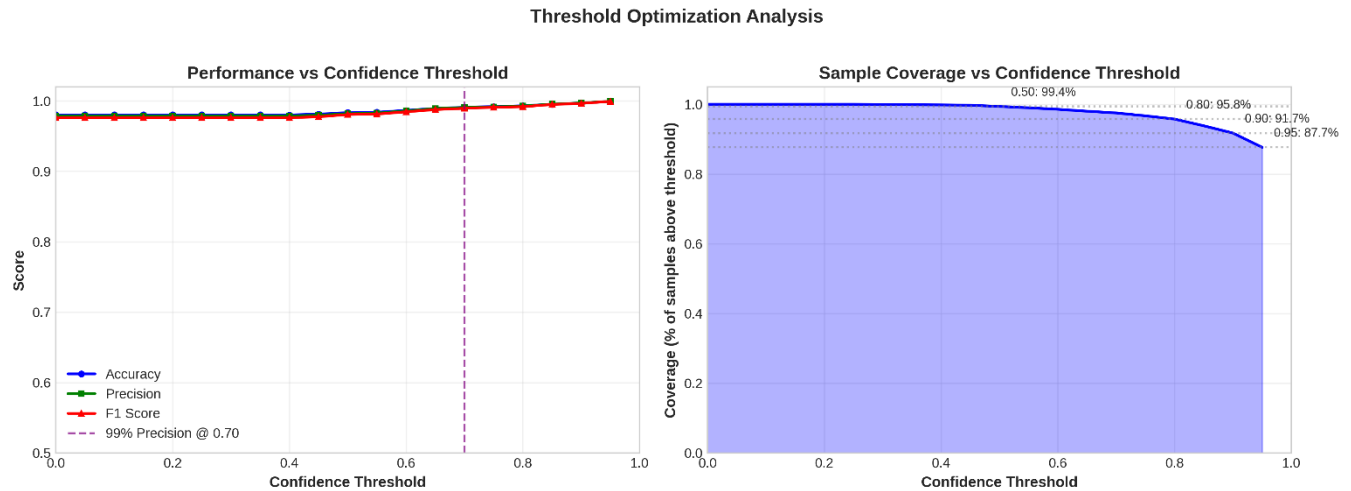


Fig. 10. Threshold optimization analysis showing coverage vs. accuracy trade-off across confidence thresholds from 0.0 to 0.95.

V. CONCLUSION

This paper presented an AI-driven diagnostic framework for multi-class tomato leaf disease classification using a novel dual-stage CAE-CNN approach. The key contributions include: (1) Demonstration that self-supervised CAE pre-training achieves competitive performance (98.02% accuracy, 0.9762 F1-score) without external pre-trained weights; (2) Validation of two-phase training strategy yielding 26.7% F1 improvement over frozen encoder baseline; (3) Near-perfect discrimination capability (ROC-AUC 0.9998) across 10 disease classes; and (4) Production-ready inference pipeline with confidence-based thresholding for deployment flexibility.

The proposed framework offers significant business benefits for the agricultural sector: (1) Cost Reduction: Automated disease detection reduces dependency on expensive laboratory testing and expert consultations, potentially saving farmers \$200-500 per hectare annually in diagnostic costs; (2) Yield Improvement: Early disease identification enables timely intervention, potentially reducing crop losses by 15-25% and increasing net farmer income; (3) Scalability: The lightweight model (16.9M parameters) enables deployment on edge devices and smartphones, making advanced diagnostics accessible to small-scale farmers in resource-constrained environments; (4) Sustainable Agriculture: Targeted treatment based on accurate diagnosis reduces pesticide overuse by 20-30%, promoting environmentally responsible farming practices aligned with UN SDG 2 (Zero Hunger) and SDG 12 (Responsible Production); (5) Supply Chain Optimization: Real-time disease monitoring across farms enables better crop forecasting and supply chain planning, reducing market volatility.

Future research directions include: (1) Mobile deployment through TensorFlow Lite or ONNX conversion for smartphone applications; (2) Extension to multi-crop scenarios using the same methodology; (3) Real-field validation on PlantDoc and other challenging datasets; (4) Disease severity estimation beyond binary classification; (5) Integration of explainability techniques such as Grad-CAM for interpretable predictions; and (6) Federated learning implementation for privacy-preserving distributed training across multiple farms.