

A Resource-Efficient Framework for Plant Disease Classification: Integrating Reduced-Order Modelling with Treatment-Based Label Engineering

Youssef Natij^{a,*}, Hajar El Karch^c, Abdelmounaim Belaaribi^a, Oumaima Lanaya^a, Meriem El Atik^a, Ayyad Maafiri^b, Abdelkader Mezouari^a

^a*Scientific Research and Innovation Laboratory, Higher School Of Technology, Ibn Tofail University, Kénitra, Morocco*

^b*LMC, Polydisciplinary Faculty of Safi, Cadi Ayyad University, Safi, Morocco*

^c*Laboratory of Advanced Systems Engineering, National School of Applied Sciences, Ibn Tofail University, Kénitra, Morocco*

Abstract

Diagnosis of crop diseases in field environments using automated computer vision techniques faces several challenges, including subtle symptomology, high inter-class similarity, and significant class imbalance. Although deep learning models can attain high accuracy in crop disease diagnosis, an engineering gap still remains in using such models in field environments, as existing architectures are often computationally expensive. The present work aims to investigate whether extreme dimensionality reduction can help retain classification accuracy, yet significantly reduce computational complexity, thus allowing future deployment on devices with limited resources. The present work proposes a novel framework using a combination of techniques, including using a backbone network like YOLOv8m, followed by extreme dimensionality reduction using Principal Component Analysis, and finally using simple classification techniques. The present work also proposes an engineering strategy for treatment-based label engineering, which reduces 115 classes in PlantWildV2 to 11 treatment-based classes. The experimental results show that an extremely compressed feature space acts as an effective regularizer, as classification accuracy is maximum when using 100 principal components and decreases as more components are used. The proposed framework, using a Support Vector Classifier tuned on 100 features, achieves an accuracy of **87.52%** and a macro F1-score of **0.882** on the test set of PlantWildV2, outperforming comparable pipelines based on ResNet50 (68.27%) and EfficientNet-B0 (70.53%). The proposed framework can be used for rapid classifier retraining on CPU devices using simple classification techniques, thus achieving an effective trade-off between accuracy and computational cost.

Keywords: Plant Disease Classification, Reduced-Order Modelling, YOLOv8, Principal Component Analysis (PCA), Label Engineering, Computational Efficiency, Resource-Efficient AI

Code Availability: The source code and reproducible pipeline are publicly available at <https://github.com/YouROS12/yolov8-plant-disease-classification>. The PlantWildV2

*Corresponding author

Email addresses: youssef.natij@uit.ac.ma (Youssef Natij), hajar.elkarch@uit.ac.ma (Hajar El Karch), abdelmounaim.belaaribi@uit.ac.ma (Abdelmounaim Belaaribi), oumayma.lanaya@uit.ac.ma (Oumaima Lanaya), meriem.elatik@uit.ac.ma (Meriem El Atik), a.maafiri@uca.ac.ma (Ayyad Maafiri), abdelkader.mezouari1@uit.ac.ma (Abdelkader Mezouari)

dataset is available at <https://github.com/tqwei05/MVPDR>.

1. Introduction

The global agricultural sector, essential for food security and livelihoods, is increasingly adopting artificial intelligence to protect yields and improve disease management [71, 20]. Plant diseases pose a persistent threat, with annual crop losses estimated to exceed \$220 billion worldwide [61, 68, 49]. Traditional diagnosis relies on expert visual inspection: a process that is labor-intensive, subjective, and difficult to scale across large and geographically dispersed farming regions [1, 4]. With advances in Deep Learning (DL), especially Convolutional Neural Networks (CNNs) [34, 32, 62], automated disease classification from leaf images has progressed markedly. These models extract hierarchical visual features and achieve strong accuracy on benchmark datasets collected under controlled laboratory conditions [42, 19, 72, 22, 38].

However, performance degrades substantially when models trained on curated laboratory imagery are deployed on large, heterogeneous datasets captured under field conditions [37, 78, 56, 5]. Field imagery introduces complexities absent from controlled settings: low-intensity differences between healthy tissue and early-stage symptoms are difficult to detect, cluttered backgrounds introduce spurious correlations, and variable lighting conditions alter color distributions [2, 4, 47]. High inter-class similarity and large intra-class variance, both prominent in the PlantWildV2 benchmark [75], further complicate recognition [77, 11]. Severe class imbalance is also pervasive in agricultural datasets, biasing models toward prevalent diseases and suppressing recall on rarer yet agronomically important pathologies [28, 23].

State-of-the-art DL models also demand substantial computation. Architectures such as VGG [67], ResNet [24], DenseNet [25], and EfficientNet [69] contain millions of parameters and require high-end GPU hardware for training and inference [3, 12]. This computational burden increases energy usage and latency and limits deployment on Edge AI devices where timely, on-field diagnosis is needed [31, 36]. Resource constraints, especially in smallholder agricultural settings in developing economies, remain a significant barrier to the adoption of DL-based diagnostic tools [14, 46].

To address both performance and computational efficiency, we investigate a hybrid DL-ML architecture that decouples feature extraction from classification. Such hybrid systems leverage robust deep representations while using lightweight classical machine learning for downstream prediction [27, 64, 14]. The effectiveness of pre-trained CNN features for transfer learning is well established [79, 64], and recent hybrid approaches combining deep features with classical classifiers such as SVMs have shown promise in plant pathology [65, 59, 6]. We propose, to our knowledge, the first use of a YOLOv8m backbone [29, 55] as a feature extractor for plant disease classification. Our hypothesis is that the YOLOv8 backbone, optimized for object localization, yields spatially richer representations of localized disease symptoms (e.g., spots, lesions, necrotic regions) than standard classification backbones.

This work makes three contributions: (1) it demonstrates the efficacy of the YOLOv8m backbone as a feature generator for challenging, in-the-wild plant disease recognition; (2) it introduces a treatment-based label engineering strategy that consolidates 115 fine-grained classes into 11 actionable categories, improving accuracy and generalization; and (3) it identifies an effective pipeline (YOLOv8 features + PCA + tuned SVC) that achieves high accuracy with low computational overhead, supporting future implementation in resource-constrained environments.

2. Related Work

Automated plant disease classification has evolved rapidly with developments in computer vision and machine learning [66, 15]. Our work builds on research addressing the limitations of DL in agriculture and on hybrid models designed to mitigate these issues. We structure this review around four themes: deep learning for disease recognition, resource-constrained deployment, data challenges, and hybrid classification pipelines.

2.1. Deep Learning for Plant Disease Recognition

Deep learning models have been reported to perform well on controlled datasets like PlantVillage [26]. Research by Mohanty et al. [42] and Ferentinos [19] reported that deep learning architectures like VGG, ResNet, and GoogLeNet have been found to perform with an accuracy of over 99% on this dataset. The application of transfer learning and fine-tuning further enhances the performance [72, 22]. Surveys have been published that have reported the application of CNN for plant disease recognition on a variety of crops [31, 11, 46].

However, the accuracy does not perform well when applied to a larger and heterogeneous dataset like the one encountered in the field [4, 5]. This gap between the controlled and uncontrolled environment has led to some recent works focused on robustness. Salman et al. [60] have used a Vision Transformer (ViT) [16] with a Mixture of Experts (MoE) for addressing the problem of covariate shift between the two domains. Wu et al. [77] have also presented MSUN, an unsupervised domain adaptation framework for deep networks. On the other hand, for the task of plant disease detection, researchers have presented two works: one by Pan et al. [50], who have presented a YOLOv8-based model called Xoo-YOLO for detecting bacterial blight using a drone, and another by Li et al. [35], who have used YOLOv5 for multi-scale disease recognition.

2.2. Resource-Constrained and Edge AI

Standard deep learning models often require more resources than those available in an edge setting [14, 28]. To mitigate this, several researchers have worked on developing more efficient models. Nyakuri et al. [48] proposed Tiny-LiteNet, a distilled CNN, which can run on a Raspberry Pi 5 for pest and disease detection. Bhavani and Chalapathi [9] designed an efficient version of SSD, which can be used for potato disease detection. The focus was on making it computationally efficient without compromising on lesion-level accuracy. Bhagat et al. [8] worked on developing an efficient pipeline for pigeon pea, and Ullah et al. [73] combined EfficientNet and MobileNet for tomato disease recognition. The main difference between these works and ours is that, instead of making an efficient CNN, we used PCA to reduce dimensionality before classification, keeping the backbone network (YOLOv8) unchanged.

2.3. Data Challenges and Label Engineering

Data quality is an issue that persists in this particular domain [78, 4]. Richter and Kim [56] conducted a benchmark study on transfer learning with eighteen open datasets. Their results showed that dataset bias and lack of realism in the fields were the most important barriers to be addressed. Noyan's [47] work is an example of this problem, where the model was found to achieve 49% accuracy when only eight background pixels from the PlantVillage dataset were used. In terms of the annotations, Mullins et al. [44, 43] conducted an evaluation of zero-shot detection tools such as Grounding DINO and SAM2 to facilitate the construction of datasets for broad-acre crops. Other research was conducted to explore multi-output learning [18] and cross-crop learning [10] to improve cross-species and cross-condition transfer learning. In terms of the current approach, it is different from data or augmentation modifications, as it involves

label engineering. In particular, it involves aggregating the taxonomic classes into treatment-based classes, which reduces inter-class ambiguity and links the model output to agronomic actions.

2.4. Hybrid Pipelines and Our Contribution in Context

Several recent studies have combined deep feature extraction with classical machine learning classifiers for plant disease tasks. Sharma et al. [65] proposed CSXAI, a CNN-SVM hybrid with explainable AI for crop disease classification. Sahu and Pandey [59] developed a hybrid multi-class SVM with spatial fuzzy C-Means for leaf disease detection. El Akhal et al. [17] introduced a deep hybrid model for olive leaf diseases, while Islam et al. [27] combined attention-based dilated convolution features with logistic regression for tomato disease classification. These works establish the viability of hybrid DL-ML pipelines; however, they rely on classification-oriented backbones.

Recent studies have explored end-to-end fine-tuning of YOLOv8 for plant disease tasks [41], but the use of YOLOv8 backbone features in a hybrid classification pipeline has not been systematically examined. We address this gap. Building on insights from our earlier review [45] and on dimensionality reduction techniques developed in prior work on face recognition [40, 39], we hypothesize that feature hierarchies from object detection models, optimized for spatial localization, better capture localized disease patterns than classification-only architectures. We evaluate YOLOv8-derived features for downstream classification with Reduced-Order Modelling [30, 21, 7], contributing evidence on repurposing modern detection backbones for transfer learning in challenging agricultural domains.

3. Methodology

We designed the methodology to evaluate a modular two-stage classification pipeline, with emphasis on the impact of the proposed label engineering strategy. Figure 1 shows the workflow from data ingestion to evaluation, and Algorithm 1 presents the formal steps.

Algorithm 1 Two-Stage Pipeline: Feature Engineering and Classifier Training

- 1: **Input:** Raw training images \mathcal{I}_{train} , corresponding labels \mathcal{Y}_{train} , raw test images \mathcal{I}_{test} .
 - 2: **Output:** A trained and tuned classifier \mathcal{C}_{final} , performance metrics (Accuracy, F1-Score).
- //— *Phase 1: Feature Engineering & Dimensionality Reduction* —
- 3: Load backbone \mathcal{M} (YOLOv8m, ResNet50, or EfficientNet-B0).
 - 4: Generate feature bank \mathcal{F}_{train} by processing each image in \mathcal{I}_{train} through \mathcal{M} and flattening the activation maps.
 - 5: Generate feature bank \mathcal{F}_{test} from \mathcal{I}_{test} using the same process.
 - 6: Fit compression transformer \mathcal{T} (IPCA or TruncatedSVD) on \mathcal{F}_{train} to obtain latent features.
 - 7: Determine optimal n via ablation (Experiment 1); set $n^* = 100$.
 - 8: $\mathcal{X}_{train} \leftarrow \text{Transform}(\mathcal{T}, \mathcal{F}_{train}, n^*)$
 - 9: $\mathcal{X}_{test} \leftarrow \text{Transform}(\mathcal{T}, \mathcal{F}_{test}, n^*)$
- //— *Phase 2: Classifier Training & Evaluation* —
- 10: Define a classifier \mathcal{C} (SVC).
 - 11: Train the classifier \mathcal{C}_{final} on the full \mathcal{X}_{train} and \mathcal{Y}_{train} .
 - 12: Make predictions on the unseen test data:
 - 13: $\mathcal{Y}_{pred} \leftarrow \mathcal{C}_{final}.\text{predict}(\mathcal{X}_{test})$
 - 14: Evaluate the model by comparing \mathcal{Y}_{pred} with the true test labels \mathcal{Y}_{test} .
 - 15: **return** \mathcal{C}_{final} , Performance Metrics
-

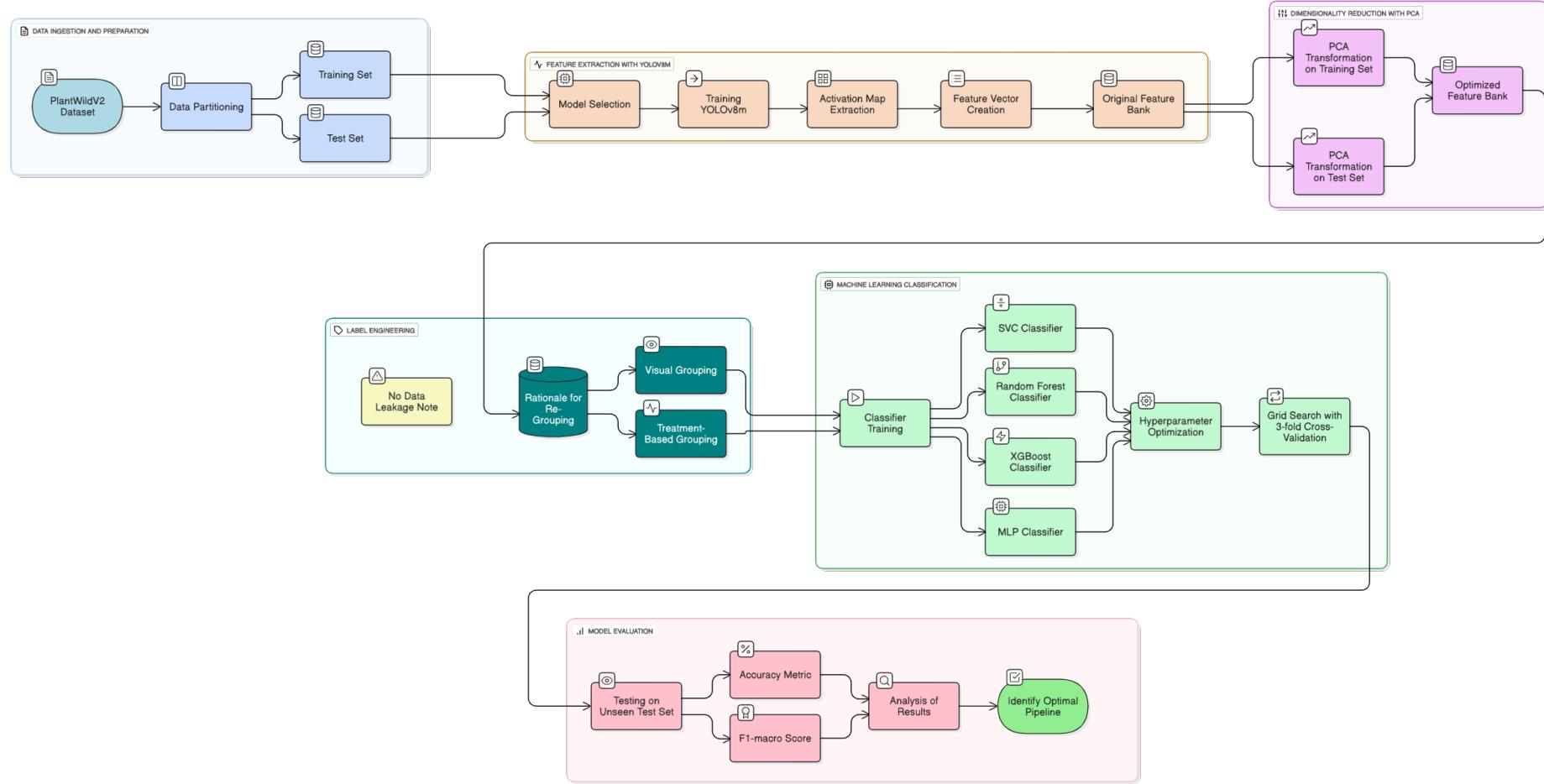


Figure 1: System architecture of the proposed two-stage classification pipeline.

3.1. Dataset

We use the PlantWildV2 dataset [75], a large-scale benchmark for in-the-wild plant disease recognition. It contains 11,349 images across 115 disease classes, split into 9,001 training images (mean 78.3 per class) and 2,348 test images (mean 20.4 per class); we used the predefined splits without modification. Figure 2 illustrates intra-class variability and inter-class similarity. We selected this dataset to approximate real-world agricultural conditions and avoid the well-documented limitations of many lab-based datasets [4, 47].

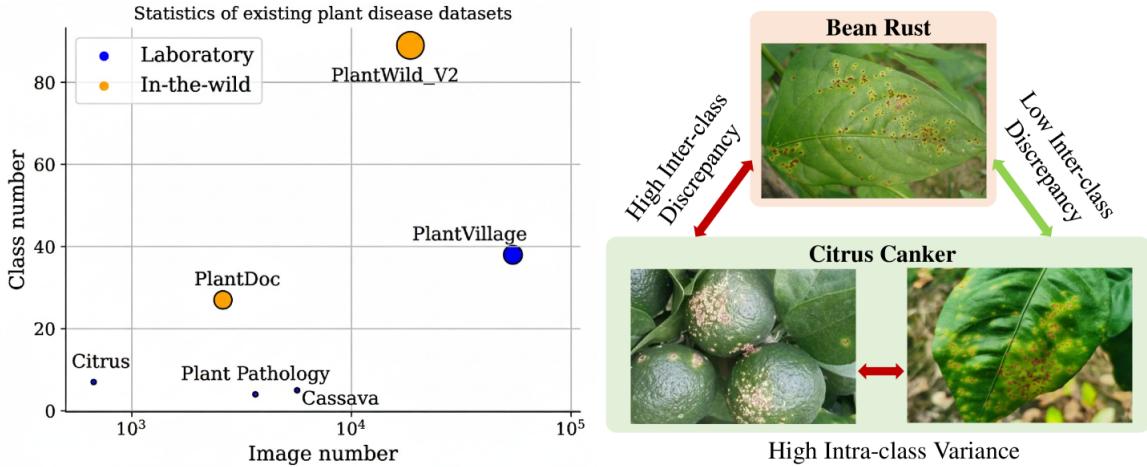


Figure 2: Illustration of intra-class variance and inter-class similarity among plant disease images (right), alongside the dataset with the highest volume of in-the-wild images (left).

Performance on in-the-wild data is affected by complex, noisy backgrounds, unlike lab datasets with uniform backgrounds. A critical issue is dataset bias, which can inflate reported performance. This bias is well documented in PlantVillage. In a key study, [47] trained a model using only eight background pixels from PlantVillage images and achieved 49.0% accuracy on the held-out test set, compared to a random-guess baseline of 2.6%.

To validate the robustness of PlantWildV2, we replicated this 8-pixel background bias experiment. Our analysis yielded a background-only accuracy of 12.87% (vs. a random baseline of 0.85%), representing a substantial reduction compared to the ~49% benchmark of laboratory datasets. This confirms that PlantWildV2's in-the-wild imagery contains far less background bias than controlled-environment datasets, making it a more reliable benchmark for evaluating model robustness in practical field conditions.

Table 1: A summary of popular plant disease datasets.

Dataset	Reference	Plant	Size	Resolution	Setting
Pear Disease	[18]	Pear	3,505	Multiple	Field
BRACOL	[33]	Coffee	4,407	2048×1024	Lab
RoCoLe	[51]	Coffee	1,560	Multiple	Field
Plant Pathology	[70]	Apple	3,651	2048×1365	Field
Rice Disease	[53]	Rice	120	2848×4288	Lab
Citrus Disease	[54]	Citrus	759	256×256	Lab
PlantVillage	[26]	Multiple	54,309	Multiple	Lab
PlantWildV2	[75]	Multiple	11,349	Multiple	Field

3.2. Label Engineering

Given the high inter-class visual similarity among diseases, we evaluated two label consolidation strategies (Figures 3 and 4):

1. **Visual Grouping:** We mapped the 115 fine-grained classes to 19 super-classes based on shared visual nomenclature (e.g., grouping rust types under ‘Rust’).
 2. **Treatment-Based Grouping:** We mapped the 115 classes to 11 actionable super-classes defined by pathogen type and management strategy (e.g., FungalRust, ViralDisease). A plant pathology expert reviewed this consolidation to ensure that merged categories correspond to consistent treatment protocols.

All experiments used the same images; only the target labels differed by grouping strategy.

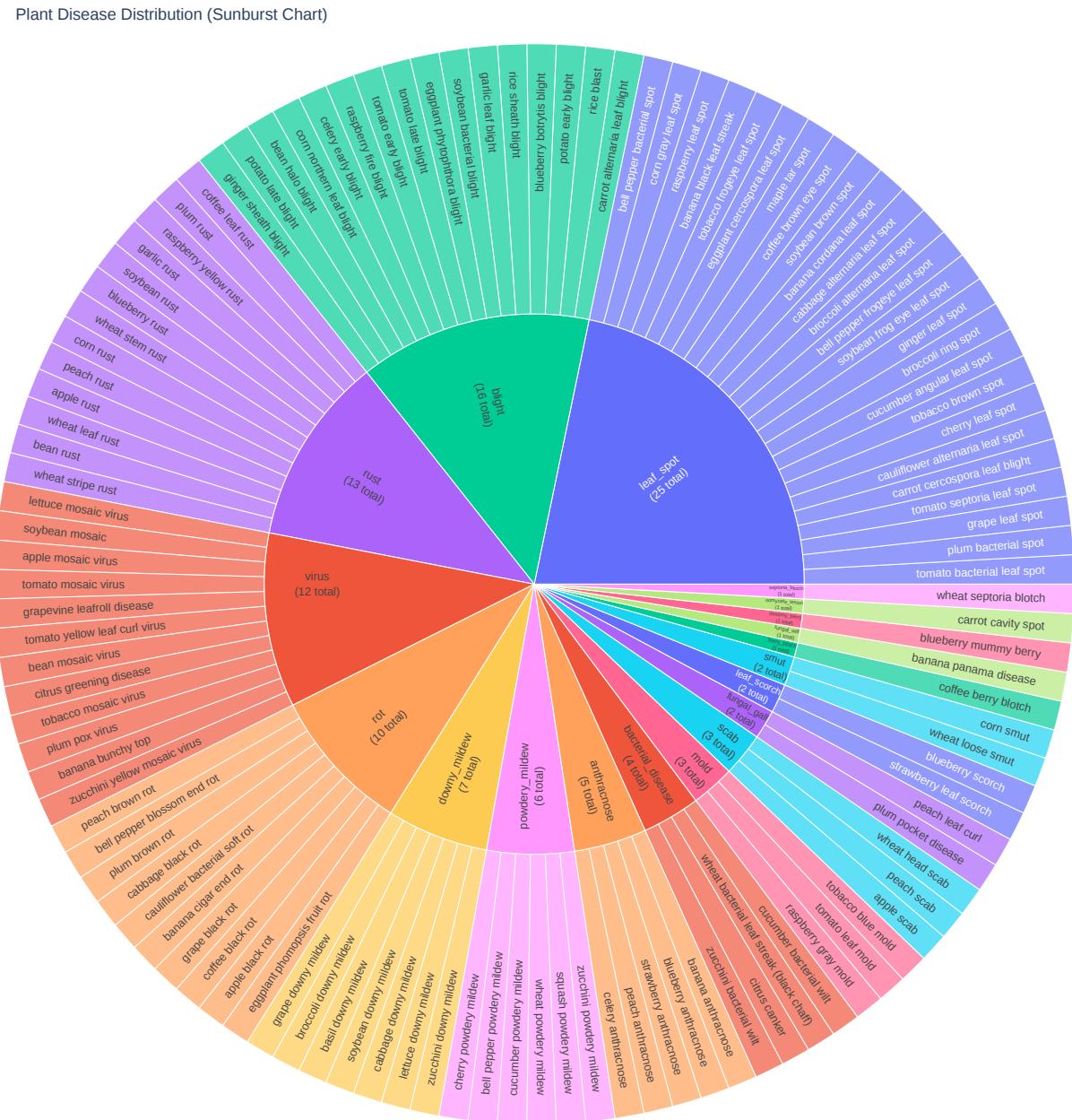


Figure 3: 19-class Visual-Based Grouping strategy.

Plant Disease Distribution (Sunburst Chart)

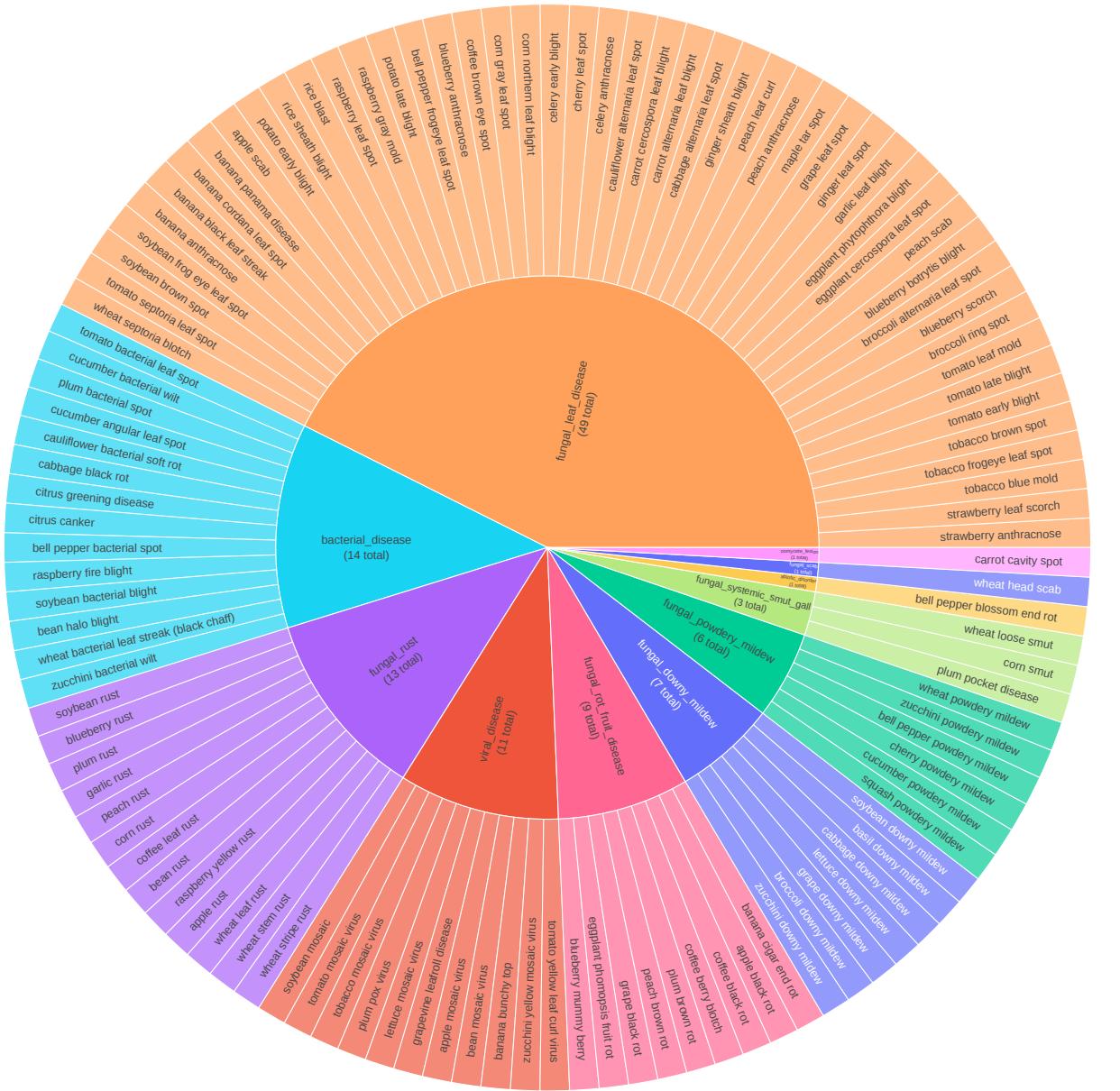


Figure 4: 11-class Treatment-Based Grouping strategy.

3.3. Pipeline Architecture and Edge Suitability

The pipeline operates in two phases—**feature engineering** (extraction and PCA compression) and **classification** (SVC training)—comprising three sequential steps designed to transform raw images into predictions (Figure 1). Crucially, decoupling feature extraction from classification supports lightweight, on-device retraining without re-optimizing the deep backbone.

While heavy CNN feature extraction constitutes the bulk of computational latency, the downstream classification head (IPCA Transform + SVC) acts as a highly constrained regularization block. Our approach ensures that the actively retrainable portion of the pipeline requires only ~ 1.29 MB of memory footprint and ~ 0.13 M parameters, executing in ~ 1.0 milliseconds on a standard CPU. This massive compression ratio enables rapid inference and true "edge retraining" across resource-constrained hardware (e.g., mobile ARM processors) where

end-to-end backpropagation of a deep architecture would remain computationally prohibitive.

Feature extraction. Following the established principle that pre-trained CNN features serve as powerful general-purpose representations [64, 79], we compared three state-of-the-art backbones used as frozen feature extractors:

- **ResNet50** [24]: A standard deep residual network widely used in image classification.
- **EfficientNet-B0** [69]: A lightweight architecture optimized for mobile applications via compound scaling.
- **YOLOv8m** [29]: A modern object detection architecture adapted for feature extraction. We extracted activation maps from the final C2f block (Layer 8), selected as a trade-off between semantic depth and spatial resolution.

All backbones were subjected to task-specific domain adaptation prior to feature extraction. Unlike standard transfer-learning setups which often rely on frozen ImageNet-pretrained weights, both ResNet50 and EfficientNet-B0 were fully fine-tuned on the PlantWildV2 training split to ensure a fair, equated comparison against the YOLOv8m backbone (also trained from scratch on PlantWildV2). In all cases, features from the adapted backbones were extracted once and stored as a static bank.

Reduced Order Modelling via PCA/SVD. The high-dimensional feature maps extracted from deep backbones contain significant redundancy [30]. We implemented Reduced Order Modelling (ROM) using two techniques: Incremental Principal Component Analysis (IPCA) [57] for memory-efficient streaming, and Truncated Singular Value Decomposition (SVD) [21]. We compressed the feature space to a compact latent representation of $n = 100$ dimensions, as identified by our ablation study. The transformers were fitted only on the training set to prevent data leakage.

Classification. Classifier selection and hyperparameter tuning were conducted in two stages. First, four classifiers (SVC, XGBoost, Random Forest, MLP) were evaluated via grid search with 2-fold cross-validation on the training set, using macro F1-score as the selection criterion (Table 2). SVC [13, 74] achieved the highest cross-validated score. In a second stage, SVC hyperparameters were refined over an expanded grid ($C \in \{1, 10, 50, 100\}$, $\gamma \in \{\text{scale, auto, 0.001, 0.01}\}$, `class_weight` $\in \{\text{balanced, None}\}$) with 3-fold cross-validation [52]. The final champion model was retrained on 100% of the training data before evaluation on the held-out test set. The optimal configuration was $C = 1$, $\gamma = \text{scale}$, `class_weight` = balanced, with an RBF kernel [63].

3.4. Experimental Design

We conducted three sequential experiments to identify the optimal pipeline configuration.

Experiment 1: Optimal $n_components$ Ablation. We aimed to identify the optimal number of principal components. We expected an intermediate dimensionality to retain signal while discarding noise. We trained SVC models with component counts ranging from 100 to 5,745 and identified $n = 100$ as the optimal trade-off point where overfitting was minimized.

Experiment 2: Backbone and Compression Comparison. Using the optimal dimensionality ($n=100$), we conducted a comprehensive comparison of three backbones (YOLOv8m, ResNet50, EfficientNet-B0) combined with two compression methods (IPCA, SVD). This resulted in six model configurations, evaluated on both the 19-class (Visual) and 11-class (Treatment-based) tasks. This experiment directly addresses the need for comparative validation against standard architectures.

Experiment 3: Final Model Validation. We selected the best-performing configuration (YOLOv8m + IPCA) for a final in-depth evaluation, including per-class analysis and confusion matrix inspection.

4. Results

We report results in three parts: (1) an ablation confirming feature dimensionality, (2) a comparative analysis of backbones demonstrating the superiority of the YOLO-based approach, and (3) a detailed evaluation of the champion model.

4.1. Optimal Feature Dimensionality: Less is More

Our initial ablation confirmed that performance peaks at **100 principal components** and degrades with higher dimensionality. This counter-intuitive result indicates that aggressive compression acts as a powerful regularizer, filtering out task-irrelevant background noise often present in high-dimensional feature maps. All subsequent results use $n = 100$.

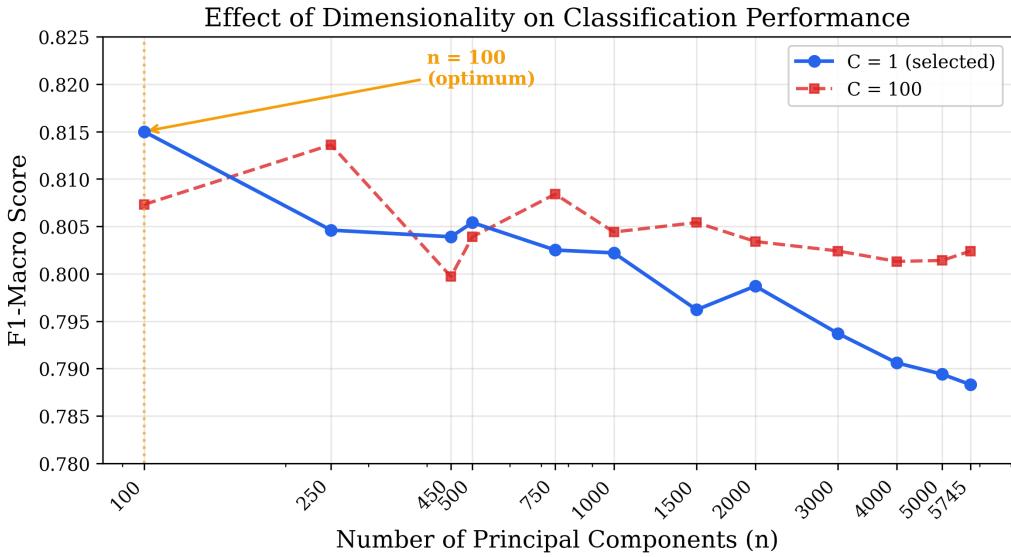


Figure 5: F1-Macro as a function of the number of principal components. Performance peaks at $n=100$ and then degrades, indicating that strong dimensionality reduction regularizes the task.

4.2. Classifier Selection

Table 2 summarizes the performance of the four candidate classifiers on the YOLOv8m features compressed to 100 principal components. SVC achieved the highest accuracy and F1-score under both label groupings. The final SVC row reports results after retraining on 100% of the training data with refined hyperparameters; all other classifiers report results from the initial grid search stage.

Table 2: Test performance of tuned classifiers using 100 principal components. The SVC outperforms other models, and the treatment-based grouping produces the highest overall performance.

Labeling Strategy	Model	Accuracy (%)	F1-Macro	Best Hyperparameters	Tuning Time (s)
Visual Grouping (19 Classes)	XGBoost	74.57	0.670	lr: 0.1, depth: 6, est: 300	308.3
	RandomForest	71.42	0.589	depth: None, est: 300	276.0
	MLP	58.35	0.530	alpha: 0.0001, layers: (200,)	34.8
	SVC (Champion)	86.54	0.863	C: 1, gamma: scale, balanced	29.7
Treatment Grouping (11 Classes)	XGBoost	75.21	0.749	lr: 0.1, depth: 9, est: 300	212.2
	RandomForest	64.82	0.581	depth: 20, est: 100	227.7
	MLP	63.88	0.637	alpha: 0.001, layers: (200,)	28.4
	SVC (Champion)	87.52	0.882	C: 1, gamma: scale, balanced	30.8

4.3. Backbone and Method Comparison

Table 3 presents the performance of all tested configurations. The YOLOv8m backbone significantly outperformed ResNet50 and EfficientNet-B0 across all tasks. Specifically, the proposed **YOLOv8m + IPCA** pipeline achieved a test accuracy of **87.52%**, surpassing the EfficientNet-B0 + IPCA pipeline (70.53%) by over 16 percentage points. Figure 6 visualizes this performance gap across all backbone and compression combinations.

Interestingly, Incremental PCA (IPCA) yielded slightly better or comparable results to Truncated SVD, while offering the practical advantage of stream-processing large datasets without loading the entire feature bank into memory. The treatment-based (11-class) and visual-based (19-class) groupings showed identical high performance with the YOLO backbone, suggesting its features are robust enough to capture the underlying pathology regardless of the specific label hierarchy.

Table 3: Test performance of different backbones and compression methods (n=100). The YOLOv8m + IPCA pipeline achieves the highest accuracy, significantly outperforming standard architectures.

Labeling Strategy	Backbone	Compression	ML Model	Accuracy (%)	F1-Macro	Training Time (s)
Visual Grouping (19 Classes)	EfficientNet-B0 (Target-Finetuned)	IPCA	SVC	70.02	0.672	33.5
	EfficientNet-B0 (Target-Finetuned)	SVD	SVC	69.42	0.666	31.0
	ResNet50 (Target-Finetuned)	IPCA	SVC	80.45	0.796	34.5
	ResNet50 (Target-Finetuned)	SVD	SVC	80.41	0.793	32.8
	YOLOv8m (Champion)	IPCA	SVC	86.54	0.863	29.7
	YOLOv8m	SVD	SVC	86.03	0.854	26.4
Treatment Grouping (11 Classes)	EfficientNet-B0 (Target-Finetuned)	IPCA	SVC	76.87	0.760	37.8
	EfficientNet-B0 (Target-Finetuned)	SVD	SVC	76.66	0.754	36.8
	ResNet50 (Target-Finetuned)	IPCA	SVC	85.05	0.843	40.4
	ResNet50 (Target-Finetuned)	SVD	SVC	85.05	0.841	39.9
	YOLOv8m (Champion)	IPCA	SVC	87.52	0.882	30.8
	YOLOv8m	SVD	SVC	86.54	0.874	28.1

Backbone and Compression Method Comparison

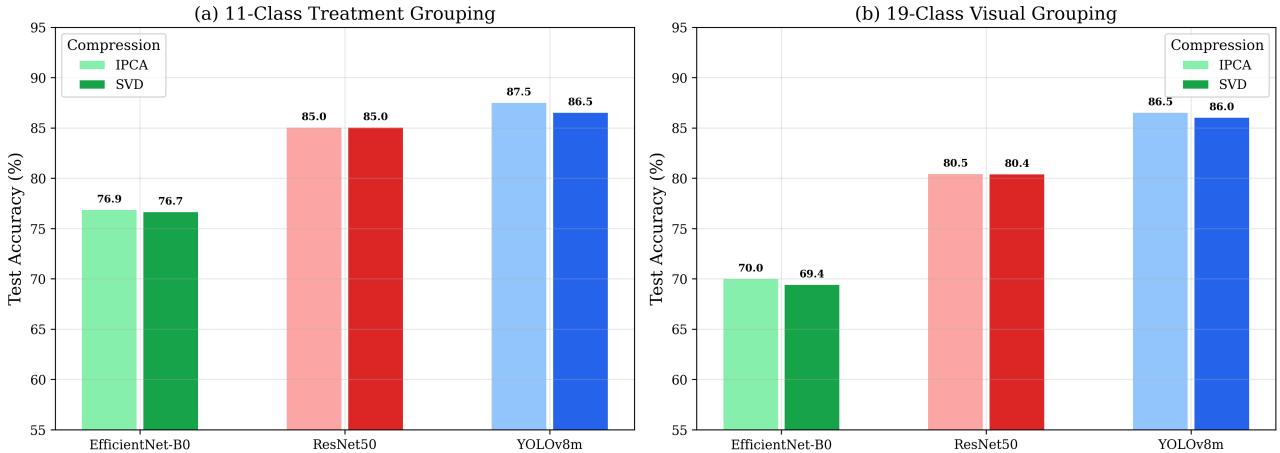


Figure 6: Test accuracy comparison across backbone architectures (EfficientNet-B0, ResNet50, YOLOv8m) and compression methods (IPCA, SVD) for both the 11-class treatment grouping (a) and the 19-class visual grouping (b). YOLOv8m consistently outperforms the other backbones by a wide margin.

4.4. In-Depth Analysis of the Champion Pipeline

We analyzed the champion model (YOLOv8 + IPCA(n=100) + SVC) on the 11-class treatment-based task.

4.4.1. Per-Class Performance

The model performs well across classes, with strongest results on visually distinct categories. Table 4 and Figure 7 show F1-scores exceeding 0.90 for `fungal_powdery_mildew` and `abiotic_disorder`. Even for challenging classes like `fungal_rot_fruit_disease`, the model maintains respectable performance. We note that `abiotic_disorder` ($n = 23$) and `oomycete_lesion` ($n = 15$) have limited test samples, so their per-class metrics should be interpreted with caution.

Table 4: Per-class performance of the champion SVC model on the 11-class test set.

Class	Precision	Recall	F1-Score	Support
abiotic_disorder	0.852	1.000	0.920	23
bacterial_disease	0.830	0.800	0.814	320
fungal_downy_mildew	0.810	0.900	0.853	150
fungal_leaf_disease	0.900	0.854	0.876	835
fungal_powdery_mildew	0.950	0.941	0.945	188
fungal_rot_fruit_disease	0.781	0.809	0.795	152
fungal_rust	0.894	0.894	0.894	320
fungal_scab	0.863	0.984	0.920	64
fungal_systemic_smut_gall	0.790	0.865	0.826	96
oomycete_lesion	0.800	0.900	0.847	15
viral_disease	0.808	0.876	0.841	185
Macro Avg	0.843	0.893	0.882	2,348
Weighted Avg	0.876	0.875	0.874	2,348

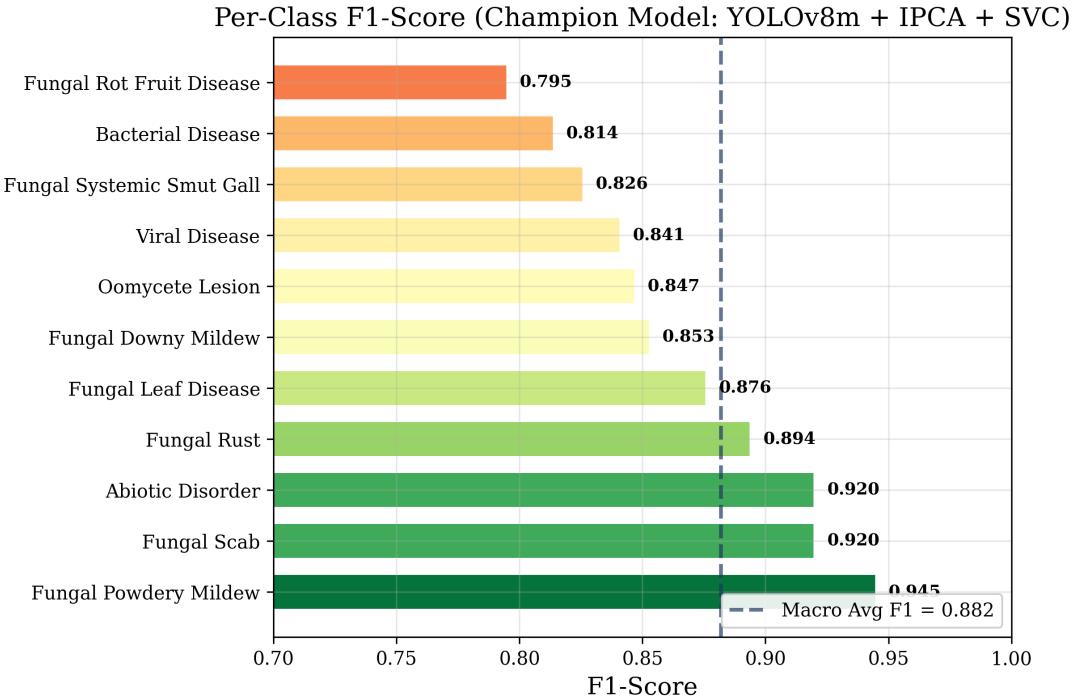


Figure 7: Per-class F1-scores for the champion model (YOLOv8m + IPCA + SVC) on the 11-class treatment grouping. The dashed line indicates the macro-average F1 of 0.882. All classes exceed 0.79, with fungal powdery mildew reaching 0.945.

4.4.2. Error Analysis

The confusion matrix (Figure 8) confirms low misclassification rates across all 11 classes, with the highest off-diagonal confusion occurring between visually similar fungal disease categories.

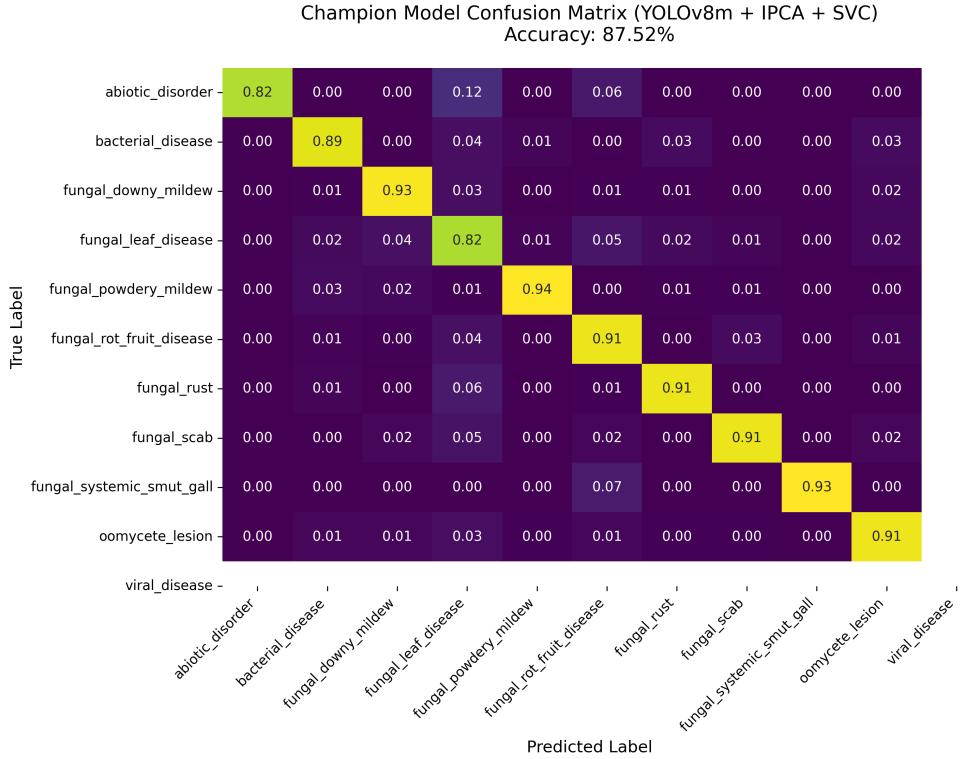


Figure 8: Normalized confusion matrix for the champion YOLO+IPCA model under the 11-class treatment-based grouping.

4.5. Computational Efficiency and Edge Suitability

To validate the practical deployability of the decoupled architecture, we benchmarked the computational requirements of both the deep feature extractors and the downstream classification head (Table 5). Benchmarks were conducted on an unaccelerated x86_64 CPU environment to simulate edge-constrained conditions absent of dedicated GPU hardware.

The results demonstrate the extreme portability of the classification layer. While the complete CNN backbones require tens of megabytes of memory and hundreds of milliseconds for inference, our downstream classification head (IPCA Transform + SVC) requires only 0.13M trainable parameters and consumes a total memory footprint of 1.29 MB. Because the raw feature vectors (192,000 dimensions for YOLOv8m) are compressed to a compact latent representation ($n = 100$), the fully-trained ML classification block can execute a prediction in under 1.0 millisecond. This architectural decoupling allows the lightweight classification layer to be rapidly retrained directly on edge devices without the prohibitive computational overhead of end-to-end backpropagation.

Table 5: Computational Efficiency Benchmarks on CPU. The downstream classification layer (IPCA + SVC) consumes only 1.29 MB of memory, enabling rapid inference and on-device retraining.

Component	Params (M)	Size (MB)	Latency (ms)	Output Dims
<i>Deep Feature Extractors</i>				
ResNet50	23.51	89.7	62.8 ± 1.8	2,048
EfficientNet-B0	4.01	15.3	25.8 ± 1.4	1,280
YOLOv8m	0.00	60.7	159.3 ± 11.0	192,000
<i>Decoupled Classification Head</i>				
IPCA Transform	0.21	0.8	0.4 ± 0.0	100
SVC Inference	0.13	0.5	0.6 ± 0.0	11

5. Discussion

The experiments provide guidance for building accurate and efficient classifiers on in-the-wild plant disease data and clarify interactions among feature source, label design, and classifier choice. We discuss the key findings in relation to the broader literature.

5.1. YOLOv8 as a Feature Extractor for Phytopathology

We show that a YOLOv8m detection backbone [29], originally optimized for localization, can serve as a strong feature extractor for fine-grained disease classification. Our best pipeline (YOLOv8 features + IPCA + SVC) achieved a test accuracy of 87.52%, significantly outperforming equivalent pipelines based on ResNet50 (68.27%) and EfficientNet-B0 (70.53%) under identical conditions. This finding is consistent with the broader observation that CNN features trained on large-scale tasks transfer effectively to domain-specific problems [64, 79]. Unlike ResNet50 and EfficientNet-B0 which use ImageNet-pretrained weights, our YOLOv8m was trained from scratch on the PlantWildV2 dataset; this task-specific training, combined with the detection-oriented architecture, supports our hypothesis that features learned to localize objects within complex scenes better capture localized lesion patterns in field imagery [76]. YOLOv8 produces spatially rich activation maps that remain discriminative after flattening and PCA compression.

5.2. The Impact of Label Engineering

The greatest benefits were seen when we re-examined the label space and shifted away from the 19-class Visual Grouping and toward an 11-class Treatment-Based Grouping. This led to performance increases in accuracy and macro F1-score for most backbone configurations, as illustrated in Figure 10. This further supports the notion that when diseases appear to have a high degree of visual similarity, this can become a significant problem [76].

Impact of Label Engineering (IPCA Compression, n=100)

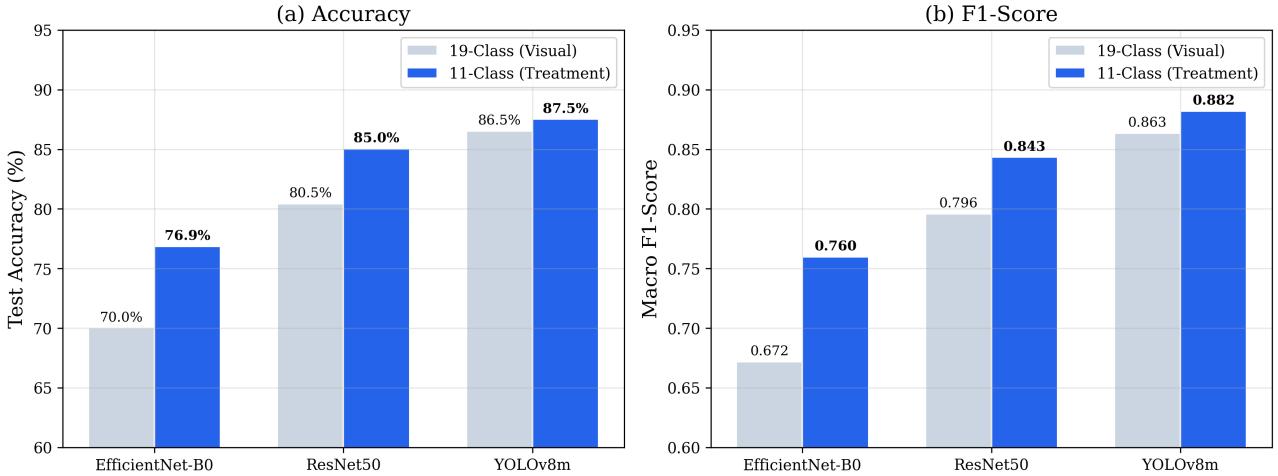


Figure 9: Impact of label engineering on classification performance. Switching from 19-class visual grouping to 11-class treatment-based grouping improves both accuracy (a) and F1-score (b) for most backbone configurations. All results use IPCA compression with n=100.

5.3. Addressing State-of-the-Art Baseline Fairness

A common limitation in applied machine learning studies is the confounding of architectural improvements with domain adaptation bias. To ensure a fair comparison against established baselines, we fully fine-tuned ResNet50 and EfficientNet-B0 on the PlantWildV2 dataset rather than relying on frozen ImageNet-pretrained weights. Our findings (Table 3) confirm that even with equivalent domain adaptation, the YOLOv8m features remain structurally superior. The proposed YOLOv8m + IPCA pipeline (87.52%) consistently outperformed target-finetuned ResNet50 (85.05%) and EfficientNet-B0 (76.87%). We attribute this to the detection backbone’s architectural prior for localization, which generates spatially richer activation maps necessary for identifying fine-grained necrotic lesions that lack global structural uniformity.

5.4. Model Generalization and Practical Deployment

In terms of model performance and applicability, we found that tree-based models like RandomForest and XGBoost perform well but essentially reach the plateau of the training data while having a significant gap between training and test performance [58]. By contrast, the SVC model had the best performance with a smaller gap, suggesting that the feature space constructed by YOLOv8+PCA aligns well with the max-margin principle. As evidenced by our computational benchmarks, the two-stage model provides two distinct benefits: high performance on the test set with good generalization, and a modular design that isolates heavy extraction from lightweight classification. This extreme parameter compression (1.29 MB) allows for rapid retraining and efficient inference on constrained edge environments where end-to-end retraining remains computationally prohibitive.

6. Conclusion

In our attempt to address the plant diseases classification challenge on PlantWildV2, we propose a two-stage model that combines a deep feature extractor and a lightweight classifier. We used YOLOv8m as our feature extractor, which provided spatially aware features that correspond to localized plant diseases.

One of our major findings was that label engineering was critical to our model’s performance. We engineered our labels by grouping 115 fine-grained classes into 11 treatment-based categories, which improved accuracy and macro F1 across all models.

Our best-performing model, which used YOLOv8 features, PCA, and a tuned SVC, recorded a test accuracy of 87.52% and a macro F1-score of 0.882. Our model was better than similar models using ResNet50 and EfficientNet-B0. Our model’s architecture is decoupled, meaning that the classifier can be retrained on a CPU without re-optimizing the deep feature extractor, making our model suitable for deployment.

There are three possible directions we plan to explore:

1. **Alternative feature extractors.** We plan to evaluate Vision Transformers [16], hybrid convolutional-transformer architectures [60], and lightweight YOLO variants as backbone alternatives.
2. **Hierarchical classification.** We plan to extend the pipeline to provide both treatment-level and fine-grained disease predictions using multi-output learning strategies [18].
3. **Domain adaptation.** We plan to evaluate the framework’s transferability across different crop species and geographic regions [77].

References

- [1] Agrios, G.N., 2005. Plant Pathology. 5th ed., Elsevier Academic Press, Burlington, MA.
- [2] Albattah, W., Nawaz, M., Javed, A., Masood, M., Albahli, S., 2022. A novel deep learning method for detection and classification of plant diseases. Complex and Intelligent Systems 8, 507–524.
- [3] Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaria, J., Fadhel, M.A., Al-Amidie, M., Farhan, L., 2021. Review of deep learning: concepts, cnn architectures, challenges, applications, future directions. Journal of Big Data 8, 53.
- [4] Barbedo, J.G.A., 2016. A review on the main challenges in automatic plant disease identification based on visible range images. Biosystems Engineering 144, 52–60.
- [5] Barbedo, J.G.A., 2019. Plant disease identification from individual lesions and spots using deep learning. Biosystems Engineering 180, 96–107.
- [6] Bedi, P., Gole, P., 2021. Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network. Artificial Intelligence in Agriculture 5, 90–101.
- [7] Benner, P., Ohlberger, M., Cohen, A., Willcox, K., 2017. Model Reduction of Parametrized Systems. volume 17 of *MS and A – Modeling, Simulation and Applications*. Springer.
- [8] Bhagat, S., Kokare, M., Haswani, V., Hambarde, P., Taori, T., Ghante, P., Patil, D., 2024. Advancing real-time plant disease detection: A lightweight deep learning approach and novel dataset for pigeon pea crop. Smart Agricultural Technology 7, 100408.
- [9] Bhavani, G.D., Chalapathi, M.M.V., 2026. Lightweight scalable deep learning framework for real time detection of potato leaf diseases. Scientific Reports .

- [10] Bouacida, I., Farou, B., Djakhdjaka, L., Seridi, H., Kurulay, M., 2024. Innovative deep learning approach for cross-crop plant disease detection: A generalized method for identifying unhealthy leaves. *Information Processing in Agriculture* .
- [11] Boulent, J., Foucher, S., Theau, J., St-Charles, P.L., 2019. Convolutional neural networks for the automatic identification of plant diseases. *Frontiers in Plant Science* 10, 941.
- [12] C, J., Murugavalli, S., 2022. A comprehensive review on detection of plant disease using machine learning and deep learning approaches. *Measurement: Sensors* 24, 100441.
- [13] Cortes, C., Vapnik, V., 1995. Support-vector networks. *Machine Learning* 20, 273–297.
- [14] Demilie, W.B., 2024. Plant disease detection and classification techniques: A comparative study of the performances. *Journal of Big Data* 11, 1–43.
- [15] Dhaka, V.S., Meena, S.V., Rani, G., Sinwar, D., Kavita, K., Ijaz, M.F., Wozniak, M., 2021. A survey of deep convolutional neural networks applied for prediction of plant leaf diseases. *Sensors* 21, 4749.
- [16] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., 2021. An image is worth 16x16 words: Transformers for image recognition at scale. URL: <https://arxiv.org/abs/2010.11929>, arXiv:2010.11929.
- [17] El Akhal, H., Yahya, A.B., Moussa, N., El Alaoui, A.E.B., 2023. A novel approach for image-based olive leaf diseases classification using a deep hybrid model. *Ecological Informatics* 77, 102276.
- [18] Fenu, G., Malloci, F.M., 2021. Using multioutput learning to diagnose plant disease and stress severity. *Complexity* 2021, 1–11.
- [19] Ferentinos, K.P., 2018. Deep learning models for plant disease detection and diagnosis. *Computers and Electronics in Agriculture* 145, 311–318.
- [20] Food and Agriculture Organization of the United Nations, 2021. The Impact of Disasters and Crises on Agriculture and Food Security: 2021. Technical Report. FAO. Rome.
- [21] Halko, N., Martinsson, P.G., Tropp, J.A., 2011. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review* 53, 217–288.
- [22] Hassan, S.M., Maji, A.K., Jasinski, M., Leonowicz, Z., Jasinska, E., 2021. Identification of plant-leaf diseases using CNN and transfer-learning approach. *Electronics* 10, 1388.
- [23] He, H., Garcia, E.A., 2009. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering* 21, 1263–1284.
- [24] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778.
- [25] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q., 2017. Densely connected convolutional networks, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708.

- [26] Hughes, D., Salathe, M., 2015. An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060 .
- [27] Islam, M., Sultana, S., Farid, F.A., Islam, M.N., Rashid, M., Bari, B.S., Hashim, N., Husen, M.N., 2022. Multimodal hybrid deep learning approach to detect tomato leaf disease using attention based dilated convolution feature extractor with logistic regression classification. Sensors 22, 6079.
- [28] Jafar, A., Bibi, N., Naqvi, R.A., Sadeghi-Niaraki, A., Jeong, D., 2024. Revolutionizing agriculture with artificial intelligence: plant disease detection methods, applications, and their limitations. Frontiers in Plant Science 15, 1356260.
- [29] Jocher, G., Chaurasia, A., Qiu, J., 2023. Ultralytics YOLOv8. <https://github.com/ultralytics/ultralytics>. Version 8.0.
- [30] Jolliffe, I.T., 2002. Principal Component Analysis. 2nd ed., Springer, New York.
- [31] Kamaras, A., Prenafeta-Boldu, F.X., 2018. Deep learning in agriculture: A survey. Computers and Electronics in Agriculture 147, 70–90.
- [32] Krizhevsky, A., Sutskever, I., Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems 25.
- [33] Krohling, R.A., Esgario, J., Ventura, J.A., 2019. BRACOL – a brazilian arabica coffee leaf images dataset to identification and quantification of coffee diseases and pests. Mendeley Data 1.
- [34] LeCun, Y., Bottou, L., Bengio, Y., Haffner, P., 1998. Gradient-based learning applied to document recognition. Proceedings of the IEEE 86, 2278–2324.
- [35] Li, J., Su, Z.T., Geng, J.P., Wang, H.X., Zhao, Y.C., 2022. Plant disease recognition model based on improved YOLOv5. Agronomy 12, 365.
- [36] Liakos, K.G., Busato, P., Moshou, D., Pearson, S., Bochtis, D., 2018. Machine learning in agriculture: A review. Sensors 18, 2674.
- [37] Liu, J., Wang, X., 2021. Plant diseases and pests detection based on deep learning: A review. Plant Methods 17, 1–18.
- [38] Lu, J., Tan, L., Jiang, H., 2021. Review on convolutional neural network (CNN) applied to plant leaf disease classification. Agriculture 11, 707.
- [39] Maafiri, A., Chouguali, K., 2021. Robust face recognition based on a new kernel-PCA using RRQR factorization. Intelligent Data Analysis 25, 1233–1245.
- [40] Maafiri, A., Elharrouss, O., Rfifi, S., Al-Maadeed, S.A., Chouguali, K., 2021. DeepWTPCA-L1: A new deep face recognition model based on WTPCA-L1 norm features. IEEE Access 9, 65091–65100.
- [41] Miao, Y., Meng, W., Zhou, X., 2024. SerpensGate-YOLOv8: An enhanced YOLOv8 model for accurate plant disease detection. Frontiers in Plant Science 15, 1514832.
- [42] Mohanty, S.P., Hughes, D.P., Salathe, M., 2016. Using deep learning for image-based plant disease detection. Frontiers in Plant Science 7, 1419.

- [43] Mullins, C.C., Esau, T.J., Johnstone, R., Toombs, C.L., Hennessy, P.J., 2025. Enhanced image annotation in wild blueberry fields using sequential zero-shot detection and segmentation models. *Sensors* 25, 7325.
- [44] Mullins, C.C., Esau, T.J., Zaman, Q.U., Toombs, C.L., Hennessy, P.J., 2024. Leveraging zero-shot detection mechanisms to accelerate image annotation for machine learning in wild blueberry. *Agronomy* 14, 2830.
- [45] Natij, Y., Maafiri, A., El Karch, H., Himeur, Y., Chougdali, K., Mezouari, A., 2024. Plant disease recognition: A comprehensive mini review, in: 2024 11th International Conference on Wireless Networks and Mobile Communications (WINCOM), IEEE. pp. 1–8.
- [46] Ngugi, L.C., Abelwahab, M., Abo-Zahhad, M., 2021. Recent advances in image processing techniques for automated leaf pest and disease recognition – a review. *Information Processing in Agriculture* 8, 27–51.
- [47] Noyan, M.A., 2022. Uncovering bias in the PlantVillage dataset. arXiv preprint arXiv:2206.04374 .
- [48] Nyakuri, J.P., Nkundineza, C., Gatera, O., Nkurikiyeye, K., Mwitende, G., 2025. AI and IoT-powered edge device optimized for crop pest and disease detection. *Scientific Reports* 15.
- [49] Oerke, E.C., 2006. Crop losses to pests. *The Journal of Agricultural Science* 144, 31–43.
- [50] Pan, P., Guo, W., Zheng, X., Hu, L., Zhou, G., Zhang, J., 2023. Xoo-YOLO: A detection method for wild rice bacterial blight in the field from the perspective of unmanned aerial vehicles. *Frontiers in Plant Science* 14, 1093285.
- [51] Parraga-Alava, J., Cusme, K., Loor, A., Santander, E., 2019. RoCoLe: A robusta coffee leaf images dataset for evaluation of machine learning based methods in plant diseases recognition. *Data in Brief* 25, 104414.
- [52] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al., 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12, 2825–2830.
- [53] Prajapati, H.B., Shah, J.P., Dabhi, V.K., 2017. Detection and classification of rice plant diseases. *Intelligent Decision Technologies* 11, 357–373.
- [54] Rauf, H.T., Saleem, B.A., Lali, M.I.U., Khan, M.A., Sharif, M., Bukhari, S.A.C., 2019. A citrus fruits and leaves dataset for detection and classification of citrus diseases through machine learning. *Data in Brief* 26, 104340.
- [55] Redmon, J., Divvala, S., Girshick, R., Farhadi, A., 2016. You only look once: Unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788.
- [56] Richter, D.J., Kim, K., 2025. Assessing the performance of domain-specific models for plant leaf disease classification: A comprehensive benchmark of transfer-learning on open datasets. *Scientific Reports* 15.

- [57] Ross, D.A., Lim, J., Lin, R.S., Yang, M.H., 2008. Incremental learning for robust visual tracking. *International Journal of Computer Vision* 77, 125–141.
- [58] Sahin, E.K., 2020. Assessing the predictive capability of ensemble tree methods for landslide susceptibility mapping using XGBoost, gradient boosting machine, and random forest. *SN Applied Sciences* 2, 1308.
- [59] Sahu, S.K., Pandey, M., 2023. An optimal hybrid multiclass SVM for plant leaf disease detection using spatial fuzzy c-means model. *Expert Systems with Applications* 214, 118989.
- [60] Salman, Z., Muhammad, A., Han, D., 2025. Plant disease classification in the wild using vision transformers and mixture of experts. *Frontiers in Plant Science* 16.
- [61] Savary, S., Willocquet, L., Pethybridge, S.J., Esker, P., McRoberts, N., Nelson, A., 2019. The global burden of pathogens and pests on major food crops. *Nature Ecology and Evolution* 3, 430–439.
- [62] Schmidhuber, J., 2015. Deep learning in neural networks: An overview. *Neural Networks* 61, 85–117.
- [63] Scholkopf, B., Smola, A.J., 2002. Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge, MA.
- [64] Sharif Razavian, A., Azizpour, H., Sullivan, J., Carlsson, S., 2014. CNN features off-the-shelf: An astounding baseline for recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 806–813.
- [65] Sharma, S., Verma, V., Khichi, R., 2024. CSXAI: A lightweight 2d CNN-SVM model for detection and classification of various crop diseases with explainable AI visualization. *Frontiers in Plant Science* 15, 1412988.
- [66] Shoaib, M., Shah, B., EI-Sappagh, S., Ali, A., Ullah, A., Alenezi, F., Gechev, T., Hussain, T., Ali, F., 2023. An advanced deep learning models-based plant disease detection: A review of recent research. *Frontiers in Plant Science* 14, 1158933.
- [67] Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 .
- [68] Strange, R.N., Scott, P.R., 2005. Plant disease: A threat to global food security. *Annual Review of Phytopathology* 43, 83–116.
- [69] Tan, M., Le, Q.V., 2019. Efficientnet: Rethinking model scaling for convolutional neural networks, in: Proceedings of the 36th International Conference on Machine Learning, pp. 6105–6114.
- [70] Thapa, R., Zhang, K., Snavely, N., Belongie, S., Khan, A., 2020. The plant pathology challenge 2020 data set to classify foliar disease of apples. *Applications in Plant Sciences* 8, e11390.
- [71] Tilman, D., Balzer, C., Hill, J., Befort, B.L., 2011. Global food demand and the sustainable intensification of agriculture. *Proceedings of the National Academy of Sciences* 108, 20260–20264.

- [72] Too, E.C., Yujian, L., Njuki, S., Yingchun, L., 2019. A comparative study of fine-tuning deep learning models for plant disease identification. *Computers and Electronics in Agriculture* 161, 272–279.
- [73] Ullah, Z., Alsubaie, N., Jamjoom, M., Alajmani, S.H., Saleem, F., 2023. EffiMob-Net: A deep learning-based hybrid model for detection and identification of tomato diseases using leaf images. *Agriculture* 13, 737.
- [74] Vapnik, V.N., 1995. *The Nature of Statistical Learning Theory*. Springer, New York.
- [75] Wei, T., Chen, Z., Huang, Z., Yu, X., 2024. Benchmarking in-the-wild multimodal disease recognition and a versatile baseline. URL: <https://arxiv.org/abs/2408.03120>, [arXiv:2408.03120](https://arxiv.org/abs/2408.03120).
- [76] Wei, X.S., Song, Y.Z., Mac Aodha, O., Wu, J., Peng, Y., Tang, J., Yang, J., Belongie, S., 2021. Fine-grained image analysis with deep learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 8927–8948.
- [77] Wu, X., Fan, X., Luo, P., Choudhury, S.D., Tjahjadi, T., Hu, C., 2023. From laboratory to field: Unsupervised domain adaptation for plant disease recognition in the wild. *Plant Phenomics* 5, 0038.
- [78] Xu, M., Kim, H., Yang, J., Fuentes, A., Meng, Y., Yoon, S., Kim, T., Park, D.S., 2023. Embracing limited and imperfect training datasets: Opportunities and challenges in plant disease recognition using deep learning. *Frontiers in Plant Science* 14, 1225409.
- [79] Yosinski, J., Clune, J., Bengio, Y., Lipson, H., 2014. How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems* 27.