This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

An Efficient and Scalable Framework for Lightweight Crop Disease Recognition in Low-Resource Settings

Tushar Shinde
Indian Institute of Technology Madras
Zanzibar, Tanzania
shinde@iitmz.ac.in

Abstract

Early and accurate detection of crop diseases is essential for sustainable agriculture and global food security, particularly in regions constrained by limited computational and data resources. While deep neural networks (DNNs) have achieved near-perfect accuracy on datasets such as PlantVillage, their deployment in real-world agricultural settings remains challenging due to high data demands and resource-intensive inference.

In this paper, we present a unified and deployable framework that systematically addresses these challenges through three integrated components: (i) Coreset Selection, employing random, margin-based, and forgetting-based strategies to construct compact training subsets with as little as 5% of the original data; (ii) Resolution Scaling, exploring downsampled image resolutions of 128×128, 64×64, and 32×32 to reduce computational complexity; and (iii) Model Compression, integrating fixed-bit quantization (8-, 4-, 2-, and 1-bit), adaptive quantization, fixed pruning at various sparsity levels, and adaptive pruning.

Comprehensive experiments on the PlantVillage dataset using ResNet18, GoogleNet, EfficientNet-B0, MobileNetV2, and SqueezeNet demonstrate that our framework achieves up to 90% data reduction and 93% model compression, with less than a 2% drop in classification accuracy. Our results offer new insights into the trade-offs between data selection, input resolution, and model sparsity, and offer a practical pathway toward scalable, lightweight, and field-deployable agricultural vision systems.

1. Introduction and Related Work

The global demand for sustainable agriculture and food security is increasingly dependent on early and accurate detection of plant diseases. Automated vision-based systems powered by deep neural networks (DNNs) have demonstrated high accuracy on benchmark datasets such as PlantVillage [9]. However, their practical deployment in real-world farming environments, particularly in low-resource regions, remains a significant challenge due to limited computational capacity, memory constraints, and the high cost of data annotation.

Despite their impressive performance, state-of-the-art DNNs are typically data-hungry, compute-intensive, and memory-heavy. These challenges are further amplified when deploying models on mobile edge devices such as drones or smartphones, commonly used in precision agriculture. Moreover, agricultural datasets often exhibit class imbalance, environmental variability, and limited diversity, calling for methods that can generalize robustly from compact, high-quality training subsets.

Recent efforts have explored efficient learning and deployment strategies. *Coreset selection* aims to reduce the size of training data while maintaining model performance [3, 16], proving effective in constrained settings [6]. *Resolution scaling* balances image fidelity and computational load [18], while *model compression* techniques such as pruning [7, 12] and quantization [4, 11] offer substantial reductions in model size and inference latency.

However, these strategies are typically explored in isolation. To our knowledge, no prior work integrates coreset selection, resolution scaling, and model compression into a unified pipeline for agricultural image classification. Furthermore, the collective impact of combining these techniques remains largely underexplored on real-world agricultural datasets.

Our Contributions. In this paper, we propose a novel and unified framework for resource-efficient agricultural image classification that jointly optimizes three core dimensions:

- We investigate three coreset selection strategies: random sampling, margin-based selection, and forgetting-based selection, across multiple training fractions (e.g., 50%, 10%, 5%) to identify compact yet effective training subsets.
- We study the robustness of vision models to input image

resolutions of 128×128, 64×64, and 32×32, capturing the impact of resolution on classification performance and computational efficiency.
- We explore both fixed-bit (8-, 4-, 2-, 1-bit) and adaptive layer-wise quantization (LAQ), fixed-level pruning, and adaptive layer-wise pruning (LAP) to substantially reduce model size and inference latency.

We conduct comprehensive experiments on the PlantVillage dataset using five popular CNN architectures, ResNet18 [8], GoogleNet [19], EfficientNet-B0 [20], MobileNetV2 [15], and SqueezeNet [10], to validate the generality and effectiveness of our approach. Our unified pipeline achieves up to 93% model compression and 90% data reduction with minimal performance loss ($< 2\%$ accuracy drop), demonstrating its suitability for real-world, edge-based agricultural deployment.

The rest of this paper is organized as follows: Section 2 details our proposed framework, including coreset strategies, resolution scaling, and model compression schemes. Section 3 presents experimental setup and evaluation metrics and Section 4 presents results and ablation studies. Section 5 concludes the paper and presents future research directions in efficient agricultural AI.

2. Methodology

We propose a unified and efficient framework to develop lightweight yet accurate deep learning models for plant disease classification using the PlantVillage dataset [9]. To enable scalable deployment on edge agricultural devices, we explore three complementary strategies for model optimization: *coreset selection*, *resolution scaling*, and *model compression* via pruning and quantization. These components aim to reduce training and inference costs while maintaining or enhancing classification accuracy across diverse CNN architectures.

2.1. Coreset Selection

To reduce training costs and improve data efficiency, we employ coreset selection strategies to identify informative samples from the full dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i denotes a plant leaf image, and y_i its class label. A coreset $\mathcal{C} \subseteq \mathcal{D}$, with $|\mathcal{C}| = \alpha N$, is constructed such that its empirical loss approximates that of the full dataset:

$$L(\mathcal{C}) \approx L(\mathcal{D}), \quad \text{for } \alpha \in (0, 1.0] \tag{1}$$

We evaluate three strategies:

Random Selection. A simple baseline where $\mathcal{C}_{\text{random}}$ is uniformly sampled:

$$\mathcal{C}_{\text{random}} = \{(x_i, y_i)\}_{i \in \mathcal{S}}, \quad \mathcal{S} \subset \{1, \ldots, N\}, \ |\mathcal{S}| = \alpha N \tag{2}$$

While computationally efficient, this method often leads to performance drops at low sampling ratios.

Forgetting-Based Selection. Following [21], we track *forgetting events*, the number of times a sample transitions from correct to incorrect classification during training. The top-M samples with the highest forgetting counts are retained:

$$\mathcal{C}_{\text{forget}} = \text{Top-}M \text{ samples with highest forgetting events} \tag{3}$$

Margin-Based Selection. Inspired by [16], we select samples with the lowest prediction margins, i.e., the smallest difference between the top two class probabilities, indicating model uncertainty.

We analyze classification performance across $\alpha \in \{1.0, 0.5, 0.1, 0.05\}$ using five model architectures, assessing how selection quality impacts generalization.

2.2. Resolution Scaling

To simulate resource-constrained rural deployments and low-quality imaging hardware, we evaluate models on multiple image resolutions. Given a full-resolution input $x \in \mathbb{R}^{H \times W \times C}$, we resize to:

$$x_r \in \mathbb{R}^{H_r \times W_r \times C}, \quad \text{with } H_r = W_r \in \{32, 64, 128\} \tag{4}$$

We retrain each model on each resolution and assess the trade-off between accuracy and resolution-induced compression. While lower resolutions (e.g., 32×32) offer memory and computation savings, they may obscure visual disease cues (e.g., surface texture), potentially reducing accuracy. This analysis helps determine the optimal resolution for edge deployment with minimal performance loss.

2.3. Model Compression

To further reduce model size and computational cost, we apply both quantization and pruning. We explore fixed and adaptive strategies for each.

2.3.1. Quantization

Post-training quantization maps 32-bit weights to b-bit representations using uniform quantization. Given a weight w_i, the quantized value $\hat{w}_i$ is:

$$\hat{w}_i = \text{clip}\left(\text{round}\left(\frac{w_i}{\Delta}\right), q_{\min}, q_{\max}\right) \cdot \Delta \tag{5}$$

where Δ is the quantization step size, and $q_{\min}, q_{\max}$ are clipping bounds.

We evaluate fixed-bit quantization at $b \in \{8, 4, 2, 1\}$ and Layer-wise Adaptive Quantization (LAQ) [17], where each layer is assigned a custom bit-width b_i based on its importance, balancing model compactness and accuracy.

2.3.2. Pruning

We apply unstructured weight pruning, setting weights with magnitude below a threshold to zero:

$$\mathcal{M}_i = \{w \in W_i \mid |w| < \tau_i\}, \quad w \in \mathcal{M}_i \Rightarrow w = 0 \quad (6)$$

The threshold τ_i is set based on fixed pruning levels $k \in \{1.5, 1.0, 0.5, 0.25, 0\}$, where k is a sparsity multiplier [17] scaled by the standard deviation of layer i's weights.

We also evaluate Layer-wise Adaptive Pruning (LAP) [17], where pruning aggressiveness is governed by a custom *Layer Importance Score* to preserve critical layers.

This compression setup aligns with deployment on real-world agricultural edge devices (e.g., Jetson Nano, microcontrollers), where both compute and memory resources are severely constrained.

3. Experimental Setup

In this section, we present the experimental setup and evaluation methodology for our exploration of efficient learning paradigms for plant disease classification. Our goal is to assess the interplay between data subset selection, resolution scaling, and model compression strategies—including fixed and adaptive quantization and pruning, toward building lightweight yet accurate models deployable on edge devices in agricultural environments.

3.1. Dataset

We conduct all experiments on the PlantVillage dataset [13], a large-scale benchmark designed for plant disease classification. The dataset comprises 54,306 RGB images of crop leaves collected under controlled conditions. These images span 38 distinct classes, which include 26 disease categories and 12 healthy leaf classes, across 14 crop species such as tomato, potato, corn, grape, apple, and pepper.

Each image is annotated with a single label indicating both the crop species and its health condition. While the controlled acquisition ensures consistent imaging backgrounds, often with plain textures distinct from leaf colors, the dataset poses significant challenges due to its high intra-class variability (arising from variations in lighting, leaf orientation, and disease severity) and high inter-class similarity (particularly among diseases affecting the same species). Moreover, the dataset suffers from class imbalance: some classes contain over 5,000 samples, while others have fewer than 200. This skewed distribution adds complexity to the classification task and makes the dataset a robust testbed for evaluating sample-efficient learning strategies such as coreset selection and model compression techniques like quantization and pruning.

To facilitate efficient training under varied data constraints, we pre-process the dataset by applying coreset selection and resolution downsampling. The dataset is split into 80% for training and 20% for testing. For each experimental configuration, we maintain a fixed validation set and report metrics on the disjoint test set.

3.2. Training Setup

Training Configuration. All experiments were conducted using PyTorch on NVIDIA Tesla P100 and G4 GPUs. We train each model using a batch size of 64, an initial learning rate of 0.001, and the Adam optimizer. Each model is trained for 10 epochs, with early stopping based on validation accuracy and a 3-epoch warm-up. All images are normalized using dataset-specific channel statistics.

Architectures. We consider a diverse set of CNN architectures to evaluate performance under constrained data and model regimes. *ResNet18* and *GoogLeNet* serve as high-accuracy backbones with residual and inception modules respectively. *EfficientNet-B0*, *MobileNetV2*, and *SqueezeNet* are employed as low-parameter networks designed for edge deployment.

Coreset Selection. To reduce training set size while maintaining representative coverage, we apply three coreset strategies: (1) *Random* sampling, (2) *Forgetting-based* selection, which prioritizes samples frequently misclassified across training epochs [21], (3) *Margin-based* selection, which samples based on the classification margin, choosing data with the smallest margins indicating high ambiguity.

Each method is evaluated at dataset fractions $f \in \{1.0, 0.5, 0.1, 0.05\}$. Let $\mathcal{D}_f \subseteq \mathcal{D}$ be the selected coreset such that $|\mathcal{D}_f| = f \cdot |\mathcal{D}|$. These subsets serve as the new training data while keeping test sets fixed.

Resolution Scaling. To assess the trade-off between input fidelity and model efficiency, images are resized to spatial resolutions $r \in \{32 \times 32, 64 \times 64, 128 \times 128\}$, corresponding to low-, medium-, and high-fidelity representations. Let $I_r \in \mathbb{R}^{r \times r \times 3}$ be the input tensor to the CNNs. Performance degradation across resolution scales is analyzed under both full and coreset-trained models.

Pruning Setup. We apply both fixed and adaptive pruning strategies. For fixed pruning, we use a global magnitude-based pruning function where a pruning level $k \in \{1.5, 1.0, 0.5, 0.25, 0\}$ controls the pruning threshold, as described in [17]. We also employ Layer-wise Adaptive Pruning (LAP) [17], which assigns a unique pruning level k_ℓ to each layer ℓ based on its layer importance.

Quantization Setup. For post-training quantization, we compare fixed-bit quantization at $\{8, 4, 2, 1\}$ bits and Layer-wise Adaptive Quantization (LAQ) presented in [17]. In fixed-bit schemes, each weight w is uniformly quantized to b-bit precision using symmetric or asymmetric scaling. In LAQ, each layer is assigned a bit precision $b_\ell \in \{8, 4, 2, 1\}$ proportional to its layer importance.

Table 1. Top-1 accuracy (%) of five CNN models evaluated across different dataset fractions (1.0, 0.5, 0.1) and image resolutions (32×32, 64×64, 128×128) using coreset selection strategies: Random, Forgetting-based, and Margin-based.

Model	Fraction	32x32			64x64			128x128		
		Random	Forgetting	Margin	Random	Forgetting	Margin	Random	Forgetting	Margin
ResNet18	1.00	96.04	96.87	96.11	99.03	99.10	99.15	99.58	99.54	98.51
	0.50	96.59	94.97	96.64	96.59	98.44	98.36	99.21	99.24	99.27
	0.10	89.61	89.63	89.86	94.89	94.79	94.92	97.33	97.27	97.10
GoogleNet	1.00	98.35	98.08	97.84	99.27	99.24	99.32	99.59	99.15	99.64
	0.50	94.45	95.19	94.72	97.70	96.87	98.64	98.67	99.31	98.51
	0.10	85.59	86.66	85.16	94.33	92.43	92.87	97.37	97.18	96.77
EfficientNetB0	1.00	98.64	97.74	98.64	99.01	99.57	99.53	99.73	99.78	99.80
	0.50	96.52	96.46	97.28	98.95	98.38	98.48	99.56	99.48	99.41
	0.10	87.12	87.38	86.75	94.75	94.64	94.49	97.59	97.89	97.62
MobileNetV2	1.00	97.11	97.88	96.78	99.40	99.36	99.48	99.19	99.06	99.19
	0.50	95.37	95.05	96.37	98.85	98.79	97.79	98.58	99.43	99.32
	0.10	86.57	90.09	86.34	95.70	93.42	95.48	97.64	97.94	97.50
SqueezeNet	1.00	89.09	86.15	90.45	98.47	98.17	97.88	99.19	98.90	99.23
	0.50	82.84	87.31	83.38	95.25	97.27	96.67	98.62	97.12	98.25
	0.10	74.06	69.78	69.38	88.31	90.89	84.19	93.97	94.83	95.33

3.3. Evaluation Metrics

Model performance and efficiency are jointly evaluated using the following metrics:

Accuracy. Defined as the top-1 classification accuracy (in %) on the test set:

$$\text{Accuracy} = \frac{100}{|\mathcal{D}_{\text{test}}|} \sum_{(x,y)\in\mathcal{D}_{\text{test}}} \mathbb{I}(\hat{y}(x) = y) \tag{7}$$

Compression Ratio (CR). The compression ratio of a model is computed as:

$$\text{CR} = \frac{\sum_{\ell=1}^{L} p_\ell \cdot 32}{\sum_{\ell=1}^{L} p_\ell \cdot s_\ell \cdot b_\ell} \tag{8}$$

where p_ℓ is the number of parameters in layer ℓ, $s_\ell \in [0, 1]$ is the sparsity (fraction of non-zero weights), and b_ℓ is the bit-width. For unpruned layers, $s_\ell = 1$; for fully pruned layers, $s_\ell = 0$.

4. Results and Analysis

We evaluate our proposed scalable and resource-efficient crop disease classification framework on the PlantVillage dataset [13], under various coreset selection strategies, resolution levels, and model compression techniques. Table 1 presents classification results across different coreset fractions and input resolutions.

4.1. Baseline Model Performance

Baseline performance corresponds to training on the full dataset (100%) without coreset selection. At the highest resolution (128×128), all models achieve near-saturation accuracy, reflecting the ease of classification in PlantVillage with high-resolution and full data. EfficientNetB0 consistently outperforms others across all resolutions, reaching up to 99.80% accuracy. GoogleNet and ResNet18 perform comparably, while SqueezeNet shows relatively lower performance, which is expected given its lightweight architecture.

As input resolution decreases, classification accuracy declines across models. Notably, the degradation is more severe for lightweight models. For instance, SqueezeNet's accuracy drops from 99.19% at 128×128 to 89.09% at 32×32, highlighting the importance of resolution in enabling compact models to extract sufficient discriminative features.

4.2. Effect of Coreset Selection

We investigate three coreset selection strategies: Random, Forgetting-based, and Margin-based, evaluated at 100%, 50%, and 10% data fractions. As expected, reduced training data leads to performance degradation, though advanced selection methods alleviate the drop.

Forgetting-based selection often yields the best results. For example, at 10% data and 128×128 resolution, SqueezeNet achieves 95.33% accuracy with Forgetting-based coresets, outperforming Random (93.97%) and Margin (94.83%). While Random selection can be competitive at higher resolutions and with deeper models, Forgetting and Margin-based selections are more consistently effective across architectures and resolutions.

Overall, Forgetting-based coreset selection offers a strong balance between efficiency and accuracy, especially under stringent data constraints.

4.3. Impact of Resolution Scaling

Resolution scaling significantly affects classification accuracy, especially in low-data regimes. Across all models, higher resolutions consistently yield better performance. For instance, at 10% data with Margin-based selection, EfficientNetB0 improves from 86.75% at 32×32 to 97.62% at 128×128.

The trend holds across models: higher resolution provides richer spatial details that aid classification. However, models like MobileNetV2 and SqueezeNet, designed for efficiency, are more sensitive to resolution changes. SqueezeNet, in particular, shows substantial performance variation; for example, at 10% data with Forgetting-based selection, its accuracy jumps from 69.78% (32×32) to 94.83% (128×128).

These results confirm that resolution scaling can serve as an effective trade-off tool. In high-resource settings, using 128×128 yields optimal accuracy. In constrained environments, coreset selection combined with moderate resolution (64×64) still achieves high accuracy with significantly reduced computational load.

4.4. Effect of Model Compression

Model compression is essential for deploying deep neural networks on edge devices, particularly in resource-constrained agricultural environments such as drones, mobile phones, and field sensors. We evaluate two principal compression strategies: quantization and pruning. Experiments were conducted using the EfficientNetB0 backbone on the PlantVillage dataset with different coreset selection methods (Random, Forgetting-based, and Margin-based) across various data fractions (1.0, 0.5, and 0.1). Additionally, we assess adaptive compression methods: Layer-wise Adaptive Quantization (LAQ) and Layer-wise Adaptive Pruning (LAP) presented in [17].

4.4.1. Fixed and Adaptive Quantization

We first examine fixed uniform quantization using 8-bit, 4-bit, 2-bit, and 1-bit representations applied uniformly across all network layers. As anticipated, lower bit-widths lead to significant model size reductions, up to $32\times$ compression with 1-bit quantization. However, this aggressive compression incurs substantial performance penalties. Under full data (1.0 fraction), 1-bit quantization results in average accuracy drops exceeding 95% across all coreset strategies. In contrast, 8-bit quantization achieves a favorable trade-off, yielding up to $4\times$ compression with negligible accuracy degradation ($< 0.25\%$ on average).

Coreset selection strategies exhibit minimal influence under extreme quantization (1-2 bits), indicating that quantization noise dominates sample selection benefits. With moderate quantization (4 or 8 bits), Margin-based coresets consistently preserve accuracy better than Random or Forgetting-based selections.

To further enhance performance under quantization, we evaluate Layer-wise Adaptive Quantization (LAQ) strategy, which selects bit-widths per layer based on layer-wise importance. LAQ outperforms fixed-bit quantization across all data fractions and coreset strategies. For instance, under the 0.1 data fraction with Forgetting-based coreset, LAQ achieves only a 0.31% drop in accuracy compared to baseline, while still achieving a compression ratio of approximately $6.7\times$, outperforming fixed 8-bit quantization both in accuracy and size.

4.4.2. Fixed and Adaptive Pruning

We perform pruning using fixed pruning levels defined by a layer sparsity control parameter $k_l \in \{1.5, 1.0, 0.5, 0.25\}$. As expected, higher sparsity (pruning levels) lead to significant model size reductions, up to $11\times$ compression.

At pruning level $k_l = 0.25$, although the compression ratios are around $2\times$, but it provides best performance among compared higher pruning level results, since, accuracy degradation increases non-linearly with higher sparsity. For example, with Random coreset at full data, pruning with $k_l = 0.25$ results in nearly 65% accuracy drop, worsening to over 89% at $k_l = 1.5$.

Among the coreset methods, Margin-based selection appears more sensitive to aggressive pruning, likely due to overfitting on high-confidence instances. In contrast, Forgetting-based coresets demonstrate more robustness, particularly at moderate pruning levels, indicating better generalization under compression.

To further enhance performance under pruning, we evaluate Layer-wise Adaptive Pruning (LAP) strategy, which help in mitigating the accuracy drop under high sparsity. LAP assigns sparsity levels per layer based on their computed importance. Under the 0.5 data fraction and Forgetting-based coreset, LAP reduces the accuracy drop to just 0.02% while achieving comparable compression ($2.7\times$) to the fixed $k_l = 0.5$ pruning, which suffered over 95% degradation. These results highlight the advantage of importance-guided adaptive sparsity allocation over uniform pruning.

4.4.3. Interaction of Coreset and Compression

To analyze the interaction between sample selection and model compression, we jointly evaluate coreset methods under varying compression levels and data fractions. At high data fractions (1.0 and 0.5), 8-bit quantization yield good trade-offs—achieving $4\times$ compression with minimal accuracy loss across all coreset strategies.

At low data fractions (e.g., 0.1), both quantization and pruning independently lead to notable performance loss. For example, with 2-bit quantization at 0.1 fraction, accuracy drops by over 76%, even for the best-performing Margin-based coreset. Similar trend is observed for aggressive pruning levels ($k_l \leq 1.0$) under sparse data. This indicates that compression and data reduction are not always complementary and require careful balancing.

Notably, adaptive compression techniques like LAQ and LAP substantially alleviate performance degradation under low-data regimes. For instance, LAQ under 0.1 data fraction and Margin-based coreset retains baseline accuracy.

Table 2. Compression performance of the EfficientNetB0 model at 32×32 resolution across various compression strategies. We report results for quantization (Q), pruning (P), and their adaptive variants (LAQ and LAP), evaluated using three coreset selection methods: Random, Forgetting, and Margin. Performance is assessed using three metrics: accuracy degradation (Acc. diff.), compression ratio (CR), and Huffman coding-based compression ratio (HCR).

Fraction	Compression Method	Random			Forgetting			Margin		
		Acc. diff.	CR	HCR	Acc. diff.	CR	HCR	Acc. diff.	CR	HCR
1.0	Q 1-bit	-95.38%	32.0	32.0	-96.58%	32.0	32.0	-97.51%	32.0	32.0
	Q 2-bit	-96.83%	16.0	27.8	-94.48%	16.0	27.8	-88.49%	16.0	28.3
	Q 4-bit	-82.54%	8.0	20.0	-72.98%	8.0	20.0	-75.74%	8.0	20.6
	Q 8-bit	-0.02%	4.0	7.7	-0.25%	4.0	7.2	-0.11%	4.0	7.9
	P level 1.5	-89.27%	11.0	11.0	-88.37%	10.1	10.1	-88.49%	11.4	11.4
	P level 1.0	-89.27%	5.7	5.7	-88.37%	5.2	5.2	-89.27%	6.0	6.0
	P level 0.5	-96.21%	3.0	3.0	-95.49%	2.7	2.7	-96.28%	3.1	3.1
	P level 0.25	-64.77%	2.2	2.2	-79.53%	2.0	2.0	-70.61%	2.3	2.3
	Baseline	0.00%	1.0	1.0	0.00%	1.0	1.0	0.00%	1.0	1.0
	LAQ	-0.05%	6.8	11.6	-0.07%	5.9	9.6	-0.13%	8.0	14.2
	LAP	0.00%	3.1	3.1	0.03%	2.2	2.2	0.01%	2.7	2.7
0.5	Q 1-bit	-95.36%	32.0	32.0	-93.20%	32.0	32.0	-96.12%	32.0	32.0
	Q 2-bit	-94.71%	16.0	26.6	-94.74%	16.0	26.4	-94.02%	16.0	26.0
	Q 4-bit	-86.65%	8.0	17.0	-88.55%	8.0	17.5	-89.42%	8.0	16.3
	Q 8-bit	-0.01%	4.0	5.8	-0.01%	4.0	5.9	-0.18%	4.0	5.6
	P level 1.5	-87.15%	8.5	8.5	-87.09%	8.6	8.6	-87.91%	8.3	8.3
	P level 1.0	-87.15%	4.1	4.1	-94.65%	4.2	4.2	-96.33%	3.9	3.9
	P level 0.5	-94.72%	2.1	2.1	-95.38%	2.1	2.1	-95.33%	2.0	2.0
	P level 0.25	-78.34%	1.5	1.5	-60.86%	1.5	1.5	-77.19%	1.4	1.4
	Baseline	0.00%	1.0	1.0	0.00%	1.0	1.0	0.00%	1.0	1.0
	LAQ	-0.06%	6.2	9.7	-0.01%	6.1	9.7	-0.22%	6.4	10.4
	LAP	0.00%	2.6	2.6	0.02%	2.7	2.7	0.02%	2.2	2.2
0.1	Q 1-bit	-85.96%	32.0	32.0	-86.22%	32.0	32.0	-85.59%	32.0	32.0
	Q 2-bit	-83.86%	16.0	24.4	-84.12%	16.0	24.3	-76.60%	16.0	24.3
	Q 4-bit	-83.27%	8.0	14.4	-82.15%	8.0	14.4	-79.49%	8.0	14.4
	Q 8-bit	-0.84%	4.0	5.3	-0.78%	4.0	5.3	-0.20%	4.0	5.3
	P level 1.5	-86.83%	7.9	7.9	-87.09%	7.9	7.9	-86.46%	7.9	7.9
	P level 1.0	-86.83%	3.5	3.5	-87.09%	3.5	3.5	-86.46%	3.5	3.5
	P level 0.5	-84.58%	1.8	1.8	-84.28%	1.8	1.8	-83.28%	1.8	1.8
	P level 0.25	-68.69%	1.3	1.3	-71.76%	1.3	1.3	-72.28%	1.3	1.3
	Baseline	0.00%	1.0	1.0	0.00%	1.0	1.0	0.00%	1.0	1.0
	LAQ	-0.53%	6.1	9.3	-0.31%	6.7	10.4	-0.19%	4.4	6.0
	LAP	0.00%	2.2	2.2	0.00%	2.2	2.2	0.06%	1.2	1.2

LAP similarly improves performance by distributing pruning pressure more intelligently across layers.

In a nutshell, both Forgetting-based and Margin-based coreset selection consistently outperforms Random-based approaches under both fixed and adaptive compression schemes. Moreover, LAQ and LAP offer compelling advantages in balancing compression and accuracy, especially under moderate to extreme constraints. These results emphasize the importance of data- and layer-aware strategies for efficient edge deployment.

4.5. Comparison with Existing Works

We compare our proposed framework against several state-of-the-art models evaluated on the PlantVillage dataset, considering both classification accuracy and compression ratio (CR). Existing works typically focus on achieving high accuracy using full training data and uncompressed models, which limits their deployability in edge and resource-constrained environments.

Table 3 presents a comparative results with prior methods such as EfficientNetB4 [1] and MobileNetV2 [22] report impressive accuracy, they do not perform any model compression or data reduction (CR = 1.0). In contrast, our framework achieves a superior balance between efficiency and accuracy. Specifically, our baseline EfficientNetB0 model achieves the highest accuracy (0.9980) using the full dataset. Furthermore, the LAQ-compressed EfficientNetB0 (Fraction = 1.0) maintains a high accuracy of 0.9967 while achieving an $8\times$ compression ratio, and with only 50% of the training data (Fraction = 0.5), we still attain

Table 3. Comparison with existing models on PlantVillage. Accuracy refers to classification accuracy, and CR denotes compression ratio.

Method	Accuracy	CR
AlexNet (Fraction = 1.0) [13]	0.9927	1.0
GoogleNet (Fraction = 1.0) [13]	0.9934	1.0
GoogleNet (Fraction = 0.5) [13]	0.9914	1.0
GoogleNet (Fraction = 1.0) [2]	0.9935	1.0
DenseNet121 (Fraction = 1.0) [5]	0.9303	1.0
ResNet50 (Fraction = 1.0) [5]	0.8794	1.0
MobileNetV2 (Fraction = 1.0) [14]	0.9900	1.0
EfficientNetB4 (Fraction = 1.0) [1]	0.9938	1.0
MobileNetV2 (Fraction = 1.0) [22]	0.9947	1.0
Ours Baseline GoogleNet (Fraction = 1.0)	0.9964	1.0
Ours Baseline GoogleNet (Fraction = 0.5)	0.9931	1.0
Ours Baseline MobileNetV2 (Fraction = 1.0)	0.9919	1.0
Ours Baseline EfficientNetB0 (Fraction = 1.0)	**0.9980**	1.0
LAQ EfficientNetB0 (Fraction = 1.0)	0.9967	**8.0**
LAQ EfficientNetB0 (Fraction = 0.5)	0.9943	6.4

0.9943 accuracy with a CR of 6.4.

These results clearly demonstrate the effectiveness of our unified framework in significantly reducing data and model size while preserving competitive performance, establishing a practical solution for lightweight, field-deployable agricultural vision systems.

5. Conclusion

This work presents a scalable and resource-efficient deep learning framework for crop disease classification, specifically tailored for deployment in edge environments. We systematically investigate the interplay between coreset selection strategies, resolution scaling, and model compression techniques on the PlantVillage dataset. Our experiments demonstrate that: (i) advanced coreset selection methods such as forgetting-based and margin-based sampling consistently outperform random sampling, particularly under low-data regimes; (ii) resolution plays a critical role in preserving classification accuracy for lightweight models, with 64×64 emerging as a promising trade-off point for constrained settings; and (iii) adaptive model compression methods, Layer-wise Adaptive Quantization (LAQ) and Layer-wise Adaptive Pruning (LAP), significantly outperform fixed-bit or uniform sparsity baselines, preserving accuracy while achieving substantial compression.

Moreover, we show that while aggressive data reduction and compression individually degrade performance, their joint optimization via adaptive strategies can mitigate the compound impact. The proposed framework thus enables accurate crop disease classification using minimal data, reduced resolution, and highly compressed models, making it highly suitable for deployment on edge devices in real-world agricultural scenarios.

Future work will explore dynamic coreset selection during training, integration with multi-sensor data (e.g., hyperspectral, thermal), and real-time deployment on resource-constrained hardware platforms such as drones and mobile devices in the field.

References

[1] Ümit Atila, Murat Uçar, Kemal Akyol, and Emine Uçar. Plant leaf disease classification using efficientnet deep learning model. *Ecological Informatics*, 61:101182, 2021. 6

[2] Yasamin Borhani, Javad Khoramdel, and Esmaeil Najafi. A deep learning based approach for automated plant disease classification using vision transformer. *Scientific Reports*, 12(1):11554, 2022. 6

[3] Zalán Borsos, Mojmir Mutny, and Andreas Krause. Coresets via bilevel optimization for continual learning and streaming. *Advances in neural information processing systems*, 33: 14879–14890, 2020. 1

[4] Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. Low-bit quantization of neural networks for efficient inference. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3009–3018. IEEE, 2019. 1

[5] Mariana Risco Cosavalente, Sulei Jackeline Chang Román, and Carlos Arturo Cortegana Silva. Benchmarking cnn-based systems for corn leaf pest detection using fine-tuning. In *Latinx in AI@ NeurIPS 2024*, 2024. 6

[6] Chengcheng Guo, Bo Zhao, and Yanbing Bai. Deepcore: A comprehensive library for coreset selection in deep learning. In *International Conference on Database and Expert Systems Applications*, pages 181–195. Springer, 2022. 1

[7] Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015. 1

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 2

[9] David Hughes, Marcel Salathé, et al. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060*, 2015. 1, 2

[10] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and ¡0.5mb model size. In *arXiv preprint arXiv:1602.07360*, 2016. 2

[11] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713, 2018. 1

[12] Zhuang Liu, Mingjie Sun, Tinghui Zhou, Gao Huang, and Trevor Darrell. Rethinking the value of network pruning. *arXiv preprint arXiv:1810.05270*, 2018. 1

[13] Sharada P Mohanty, David P Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. *Frontiers in plant science*, 7:215232, 2016. 3, 4, 6

[14] Emmanuel Moupojou, Appolinaire Tagne, Florent Retraint, Anicet Tadonkemwa, Dongmo Wilfried, Hyppolite Tapamo, and Marcellin Nkenlifack. Fieldplant: A dataset of field plant images for plant disease detection and classification with deep learning. *IEEE Access*, 11:35398–35410, 2023. 6

[15] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4510–4520, 2018. 2

[16] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. *arXiv preprint arXiv:1708.00489*, 2017. 1, 2

[17] Tushar Shinde. Adaptive quantization and pruning of deep neural networks via layer importance estimation. In *Workshop on Machine Learning and Compression, NeurIPS 2024*, 2024. 2, 3, 5

[18] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020. 1

[19] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 2

[20] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6105–6114, 2019. 2

[21] Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. *arXiv preprint arXiv:1812.05159*, 2018. 2, 3

[22] Jianping Yao, Son N Tran, Saurabh Garg, and Samantha Sawyer. Deep learning for plant identification and disease classification from leaf images: multi-prediction approaches. *ACM Computing Surveys*, 56(6):1–37, 2024. 6