

Accuracy, Stability, and Computational Efficiency of Deep CNN Architectures for Apple Leaf Disease Classification

Brahim Ouben Hssain¹, Khalil Ladrham², Nouredine El Barbri³, Rachid El Ayachi⁴, Abdellah Halimi⁵

Laboratory of Science and Technology for the Engineer (LaSTI),
ENSA, Sultan Moulay Slimane University, Khouribga, Morocco^{1,3}

Intelligent Processing and Security of Systems-Faculty of Sciences,
Mohammed V University, Rabat, Morocco²

TIAD Laboratory-Department of Computer Science-Faculty of Sciences and Technology,
Sultan Moulay Slimane University, Morocco⁴

Laboratory of Health Sciences and Technologies-High Institute of Health Sciences,
Hassan 1st University Settat, Morocco⁵

Abstract—Reliable apple leaf disease classification requires high predictive performance and transparent computational reporting under heterogeneous field conditions. This study presents an orchard-image benchmark of four ImageNet-initialized convolutional architectures—Xception, EfficientNetB0, DenseNet121, and ResNet50V2—using images acquired at two Moroccan orchard locations (Midelt and Azrou), four cultivars, and three mobile devices. From a curated four-class corpus of 20,000 RGB images, 18,000 images (4,500 per class) formed the five-fold cross-validation benchmark; 2,000 images were isolated before fold construction as an untouched challenge reserve for future independent confirmation and were not evaluated in the present manuscript. All models used a common 224×224 input resolution, a harmonized classification head, Adam optimization, L1/L2 regularization, learning-rate reduction, and validation-loss early stopping. Evaluation combined accuracy, cross-entropy loss, macro-precision, macro-recall, macro-F1, specificity, Cohen’s kappa, Matthews correlation coefficient, fold stability, parameter count, model size, GMACs/GFLOPs, and measured CPU training time. All reported values and figures derive from completed HPC-MARWAN runs under the common protocol. Xception achieved the highest observed mean validation accuracy (97.99±0.37%) and aggregate out-of-fold macro-F1 (97.99%), whereas EfficientNetB0 reached 97.33±0.68% mean validation accuracy with 0.78 GFLOPs per image, 5.29 million parameters, and the shortest CPU training time. The results define a descriptive accuracy–cost trade-off within the benchmark; no inferential superiority or device-level deployment latency is claimed.

Keywords—Apple leaf disease; convolutional architectures; field conditions; Xception; EfficientNetB0; DenseNet121; ResNet50V2; computational efficiency

I. INTRODUCTION

Apple foliar diseases reduce photosynthetic activity, fruit quality, and marketable yield, and field appearance depends on disease stage, cultivar, illumination, leaf orientation, background vegetation, and acquisition device. Shafik et al. [1] enumerated dataset, classification, and deployment problems in plant-disease detection. Pacal et al. [2] highlighted the significance of data quality and evaluation design for the

performance of deep learning. Khan et al. [3] built a real-time deep-learning system for apple-specific diagnosis, and Wang et al. [4] proposed an evidence-fusion framework. These studies motivate evaluation under heterogeneous orchard conditions versus controlled imagery alone.

Deep convolutional neural networks (CNNs) learn features hierarchically directly from RGB images, but comparison of published studies is hampered by differences in image resolution, class balance, augmentation, partitioning, validation strategy, and reported metrics. Generalisation can be further affected by field variations in cultivar, illumination, background, and acquisition device. Therefore, a single hold-out split with high accuracy does not in itself validate model stability or transfer to unseen orchard conditions.

Computational reporting is a second limitation. Guan et al. [5] proposed a lightweight plant-disease classifier, Gao et al. [6] developed a lightweight YOLOv8 model for apple leaf disease detection, and Iftikhar et al. [7] integrated an enhanced CNN with a mobile application. These studies demonstrate the relevance of model size and computational demand. However, theoretical FLOPs and CPU training time are computational indicators rather than direct measurements of mobile or edge inference latency.

Information leakage is another methodological concern. Visually related photographs, near-duplicate acquisitions, or augmented derivatives originating from the same source image should not be distributed across training and validation partitions because such overlap can produce optimistic estimates. Shafik et al. [1] and Pacal et al. [2] identify dataset construction, transparent validation, and reproducible partitioning as central methodological requirements.

The contribution is methodological rather than the proposal of another CNN architecture. A common orchard-image and cost-aware benchmark is established for Xception, EfficientNetB0, DenseNet121, and ResNet50V2 using multi-site imagery, leakage-aware five-fold partitioning, a shared classification head, fold-level predictive and stability metrics,

and common computational indicators. Section II reviews related apple and plant-disease studies. Section III defines acquisition, partitioning, preprocessing, training, and evaluation. Section IV reports predictive and computational results; Section V interprets the measured trade-offs; Section VI states limitations and reproducibility requirements; and Section VII summarizes the conclusions and future validation priorities.

II. RELATED WORK

Apple disease classification has progressed from high-capacity CNNs toward attention, segmentation, hybrid, and resource-efficient designs. Yadav et al. proposed AFD-Net for multi-class apple foliar disease recognition and reported strong five-fold performance on the Plant Pathology datasets [8]. Gao et al. introduced BAM-Net for complex-background apple leaf recognition through attention-guided feature extraction [9]. Parashar and Johri combined Inception-v3 with image segmentation and stratified five-fold cross-validation [10]. Si et al. integrated a CNN branch with a Swin Transformer to

combine local and long-range representations [11]. Alqahtani et al. coupled EfficientNet with optimization-based feature selection [12], while Ait Nasser and Akhloufi studied a hybrid deep-learning architecture for apple foliar disease detection [13]. AppleLeafNet focused on compact classification [14]; MDCNet proposed constrained deformable convolutions [15]; Zhu et al. presented a two-stage DeepLabv3+ segmentation framework [16]; LightYOLO-AppleLeafDx focused on lightweight detection [17]; and Zhou et al. explored a multimodal lightweight Transformer [18]. Further context on efficiency is provided by ALNet [19], EnConv [20], edge-device CNN deployment [21], and ensemble hybrid optimisation [22]. The fruit-disease frameworks [23], [24], the crop-disease review by Ngugi et al. [25], the apple-specific review by Kannan et al. [26], and Karasu's optimised apple-scab classifier [27] are other examples of such methodological diversity. Therefore, Table I provides a contextual comparison, rather than a direct performance ranking, across incompatible datasets and protocols.

TABLE I. METHODOLOGICAL COMPARISON OF REPRESENTATIVE APPLE/PLANT DISEASE STUDIES

Study	Year	Model/method	Dataset/scope	Validation	Best reported result	Main limitation/distinction
Yadav et al. [8]	2022	AFD-Net (EfficientNet-B3/B4)	Plant Pathology 2020/2021	Five-fold	98.7% accuracy	High-capacity fusion; public datasets
Gao et al. [9]	2023	BAM-Net with attention	Complex-background apple leaves	Hold-out	95.64% accuracy; 95.25% F1	Attention-based; limited cost reporting
Parashar and Johri [10]	2024	Segmentation + Inception-v3	1,821 apple-leaf images; 4 classes	Stratified five-fold	94.76% accuracy	Class imbalance and preprocessing dependence
Si et al. [11]	2024	CNN + Swin Transformer	Apple disease images; 4 classes	Hold-out	97.32% accuracy	Hybrid local/global feature modelling
Alqahtani et al. [12]	2023	EfficientNet Sailfish optimization	9,714 apple-leaf images; 4 classes	Train/test	99.39% comparison result	Optimization-assisted high-complexity pipeline
Ali et al. [14]	2024	AppleLeafNet	Apple leaf disease images	Train/validation/test	98.25% condition identification	Lightweight deployment-oriented design
Zou et al. [17]	2025	LightYOLO-AppleLeafDx	Apple leaf disease detection	Detection benchmark	96.5% mAP@0.5	5.2 MB model; real-time orientation
Adithiyaa et al. [19]	2026	ALNet	Apple/grape/cherry leaf datasets	Five-fold + hold-out	99.95% apple accuracy	0.17 M-parameter lightweight model
Thanjaivadivel et al. [20]	2025	EnConv	39 plant-disease classes	Comparative classification	99.87% accuracy	Cross-crop enhanced CNN
Present benchmark	2026	Xception / EfficientNetB0 / DenseNet121 / ResNet50V2	Multi-site orchard corpus; 18,000 benchmark images	Leakage-aware five-fold	97.99% mean accuracy (Xception)	Joint accuracy-stability-cost analysis

The comparison highlights substantial variation in dataset origin, validation design, model complexity, and deployment evidence. The present work differs by treating predictive accuracy, fold stability, computational complexity, and runtime as joint selection criteria under one common orchard-image benchmark.

III. MATERIALS AND METHODS

A. Study Design and Field Acquisition

Images were collected in commercial apple orchards in Midelt and Azrou, Morocco. The two sites differ in altitude, local microclimate, orchard conditions, illumination, and background context. Photographs were taken outdoors in natural light with leaves still attached to the trees and at morning, midday, and afternoon acquisition times to maintain realistic illumination variability.

Four cultivars were represented in the field records: Gala, Golden Delicious, Starking, and Red Delicious. Three mobile devices were used: Samsung Galaxy S21 Ultra, Xiaomi Redmi 13 Pro, and iPhone 12 Pro Max. The analytical labels were image-level visual symptom categories: powdery mildew, healthy apple leaf, apple scab, and apple mosaic/chlorotic symptoms. The available benchmark records do not document laboratory confirmation or formal multi-expert adjudication for every image; consequently, the labels are interpreted as visual symptom categories rather than etiologically confirmed diagnoses. This limitation is particularly relevant to the mosaic/chlorotic category, which may have multiple causal mechanisms. Label errors identified during quality screening were removed; however, a separate adjudication record for ambiguous cases is not available for the present benchmark. Representative examples are shown in Fig. 1. Phytotoxicity was excluded because it represents abiotic or chemical stress rather than a single pathogen-defined category.

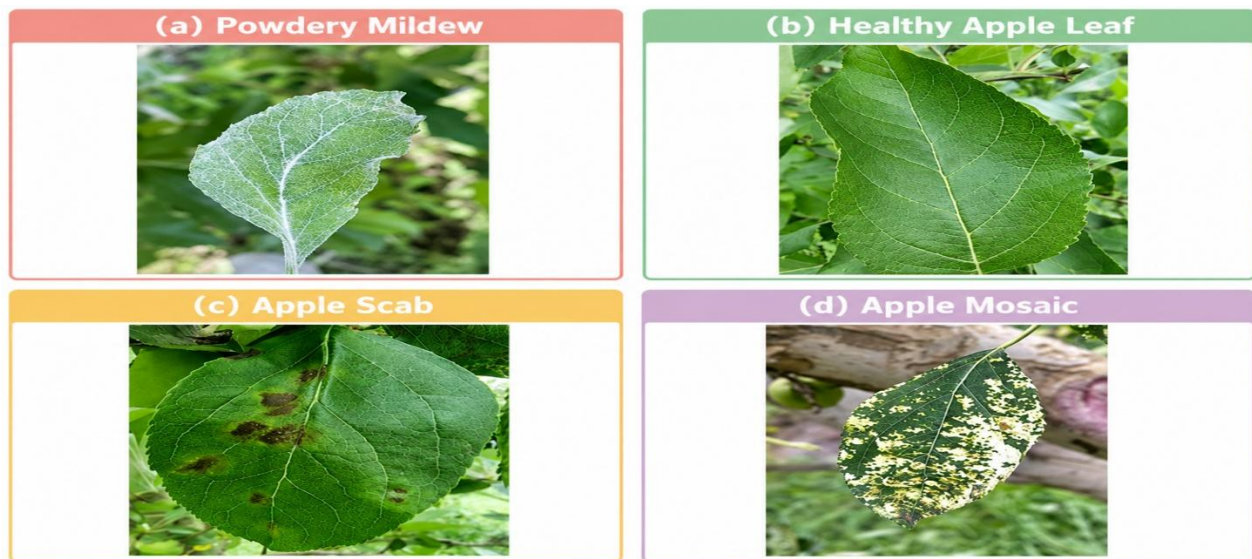


Fig. 1. Representative examples of the four analytical classes: powdery mildew, healthy apple leaf, apple scab, and apple mosaic/chlorotic symptoms.

B. Benchmark Construction and Leakage Control

The curated corpus contained 20,000 RGB images, corresponding to 5,000 images per class after controlled balancing and quality screening. A total of 18,000 images (4,500 per class) were assigned to five-fold cross-validation. The remaining 2,000 images were isolated before fold construction as an untouched challenge reserve. The reserve was excluded from architecture selection, early stopping, hyperparameter adjustment, fold construction, and all reported metrics. Its evaluation is deferred entirely to future independent confirmation, and no reserve-set performance is claimed in the present manuscript.

Before fold assignment, corrupted files, severe blur, label errors, exact duplicates, and visually redundant derivatives were

screened. The guaranteed grouping unit was the original source image together with its augmented derivatives; all derivatives from a source image remained in the same fold. Complete leaf- and tree-level identifiers were not available for all acquisitions, so strict leaf- or tree-level independence across folds cannot be guaranteed. Where capture-session identifiers were available, near-duplicate images were grouped to reduce acquisition-context leakage; orchard-level separation was not imposed. Each iteration used 14,400 training images and 3,600 validation images, and every benchmark image served as validation data once. The reported five-fold results should therefore be interpreted as leakage-aware source-image-level internal validation rather than leave-one-tree, leave-one-session, or leave-one-orchard external validation. Fig. 2 summarizes the partitioning sequence.

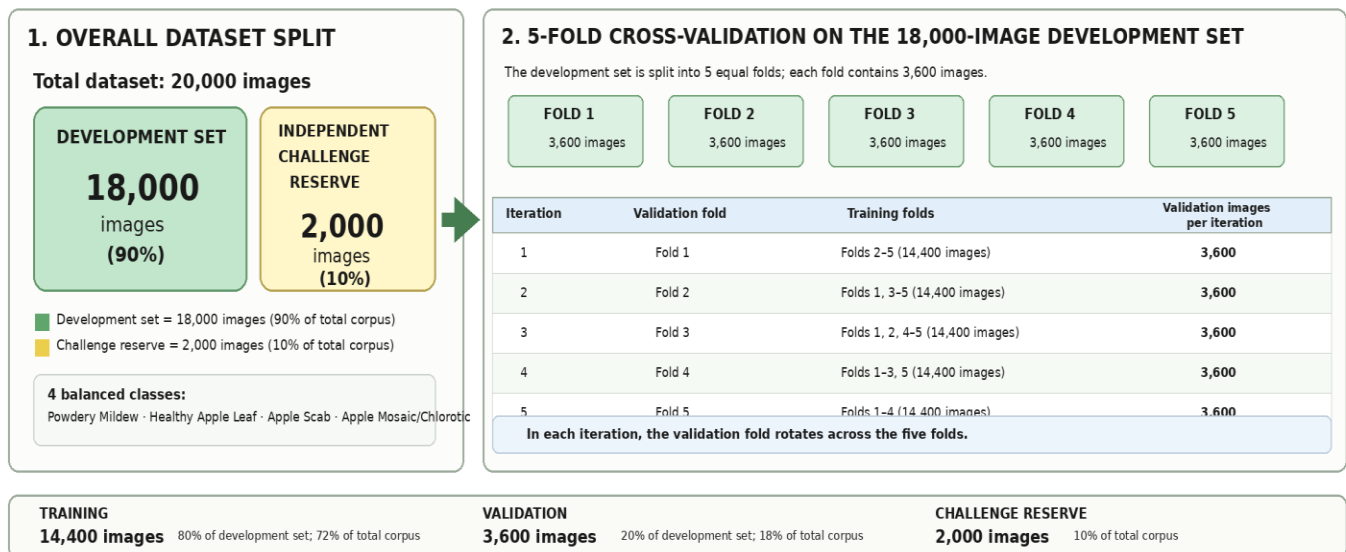


Fig. 2. Orchard-image benchmark design. The 2,000-image challenge reserve is isolated from the five-fold benchmark and is not used to derive the reported cross-validation results.

C. Preprocessing and Augmentation

Images were decoded in RGB format and resized to $224 \times 224 \times 3$ pixels. Aspect-ratio-preserving padding was applied when necessary to reduce geometric distortion. Architecture-specific input normalization followed the preprocessing conventions of the corresponding Keras backbones. Validation images underwent deterministic resizing and normalization only.

Mild online augmentation was restricted to training folds. The transformation set included horizontal flipping, rotations within $\pm 15^\circ$, translations up to $\pm 8\%$, zoom up to $\pm 10\%$, and moderate brightness/contrast adjustment. The limits were selected to preserve disease morphology while increasing robustness to leaf orientation and illumination. Augmentation was performed after fold assignment so that transformed variants could not cross the training-validation boundary.

D. CNN Architectures and Shared Classification Head

The four backbones were selected to represent established image-classification design families under one common protocol: Xception uses depthwise separable convolutions, EfficientNetB0 uses compound scaling, DenseNet121 uses dense feature reuse, and ResNet50V2 uses pre-activation

residual learning. Shafik et al. [1] and Pacal et al. [2] provide broad plant-disease context; ALNet [19] and EnConv [20] illustrate recent lightweight and cross-crop alternatives; Ngugi et al. [25] review crop-disease diagnosis; Kannan et al. [26] summarise apple-leaf advances. The chosen set is representative, not exhaustive, and is not claimed to include the strongest current architecture. LightYOLO-AppleLeafDx was retained as a literature comparator, but not inserted into the controlled classification benchmark, as detection-specific or materially different pipelines.

For each backbone, the original classification layer was removed and replaced by global average pooling, batch normalization, a 256-unit ReLU dense layer, dropout, and a four-unit Softmax output. L1 and L2 penalties were applied to the dense layer, while pretrained batch-normalization layers remained in inference mode during fine-tuning. Table II reports architecture-specific batch sizes, dropout values, learning rates, fine-tuned regions, and convergence limits. These settings were selected to obtain stable training for each backbone; consequently, the comparison represents performance under defined architecture-specific configurations rather than an ablation that isolates backbone architecture as the sole causal factor.

TABLE II. ARCHITECTURE-SPECIFIC TRAINING CONFIGURATION

Model	Batch	Dropout	Head LR	Fine-tune LR	Fine-tuned region	Max. epochs	Observed stop range
Xception	8	0.40	1×10^{-4}	3×10^{-6}	Exit flow + upper middle-flow blocks	50	30–35
EfficientNetB0	24	0.30	3×10^{-4}	1×10^{-5}	Last 30–40 layers	50	40–47
DenseNet121	12	0.35	2×10^{-4}	5×10^{-6}	Final dense block	50	42–48
ResNet50V2	16	0.35	2×10^{-4}	5×10^{-6}	Final residual stage	50	40–48

E. Training and Convergence Control

Training and all recorded fold outputs used in Sections IV–V were generated with TensorFlow/Keras on HPC-MARWAN using eight CPU cores and 16 GB RAM. Adam used $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-7}$. Each fold started from the same ImageNet initialization instead of inheriting weights from a previous fold. The classification head was optimized first, followed by low-learning-rate fine-tuning of an architecture-specific upper region. A single training run was conducted for each fold; repeated runs across multiple random seeds were not performed, so stochastic variability beyond fold-to-fold variation is not estimated.

A maximum of 50 epochs was initially specified for every architecture to provide sufficient optimization capacity. Validation loss was monitored with early stopping (patience = 5 epochs; minimum improvement = 10^{-4}), and the best validation-loss weights were restored. ReduceLROnPlateau used a reduction factor of 0.2, patience of two epochs, a one-epoch cooldown, and a minimum learning rate of 10^{-7} .

For Xception, convergence occurred earlier than the maximum training limit: the five folds stopped at epochs 30, 33, 31, 35, and 34. The training process was therefore not arbitrarily restricted to 35 epochs. Instead, the early-stopping criterion terminated optimization when validation accuracy and

validation loss had reached a stable plateau over the predefined patience interval. Continuing beyond this point would have increased computational cost without providing a meaningful validation improvement, while potentially increasing overfitting risk.

F. Evaluation Metrics and Computational Analysis

Out-of-fold predictions were used to compute accuracy, macro-precision, macro-recall, macro-F1, macro-specificity, Cohen's kappa, and the Matthews correlation coefficient (MCC). For class c, TPc, TNc, FPc, and FNc denote the standard confusion-matrix quantities. Macro-averaging gives equal weight to each class:

$$\text{Accuracy} = \sum_c \text{TPc} / N, \quad (1)$$

$$\text{Pmacro} = (1/C) \sum_c [\text{TPc} / (\text{TPc} + \text{FPc})], \quad (2)$$

$$\text{Rmacro} = (1/C) \sum_c [\text{TPc} / (\text{TPc} + \text{FNc})], \quad (3)$$

$$\text{F1macro} = (1/C) \sum_c [2\text{PcRc} / (\text{Pc} + \text{Rc})] \quad (4)$$

Categorical cross-entropy was used as the data loss. Fold stability was summarized by the standard deviation and coefficient of variation (CV) of validation accuracy, and the mean train-validation accuracy gap was reported descriptively. Because the five folds share overlapping training observations, they are not independent statistical samples. Complete leaf- and

tree-level group identifiers were also unavailable for all images; therefore, independent-sample significance tests and group-level resampling confidence intervals were not reported. In particular, the 0.66-percentage-point difference between Xception and EfficientNetB0 is treated as a descriptive benchmark difference and not as evidence of statistical superiority.

Computational cost was characterized by parameter count, model size, multiply-accumulate operations (GMACs), forward-pass GFLOPs, and measured wall-clock CPU training time per epoch. A convention of one MAC $\approx$ two FLOPs was applied consistently. Validation-accuracy points per GFLOP, per million parameters, and per CPU training minute were calculated only as descriptive within-benchmark ratios. CPU training time was measured on HPC-MARWAN; derived throughput and five-fold burden are arithmetic summaries. These values do not represent mobile/edge inference latency, energy, or peak memory.

IV. RESULTS

A. Fold-Wise Predictive Performance

Predictive results and fold-specific stopping epochs are provided in Table III. The original four-panel learning-curve format is preserved: mean training accuracy, mean validation accuracy, mean training loss, and mean validation loss are shown separately to allow direct assessment of convergence and train-validation agreement. For visualisation, folds with different early-stopping points are aligned to a normalised epoch-progress scale, which is only relevant for the graphic display and not the original stopping epochs or reported metrics. The curves demonstrate progressive saturation of accuracy and reduction of loss without significant late-stage divergence. Xception showed the best validation trajectory among the tested configurations, while the other architectures converged at slightly lower levels. The four trajectories are shown in Fig. 3, and the fold-specific stopping epochs are listed in Table III.

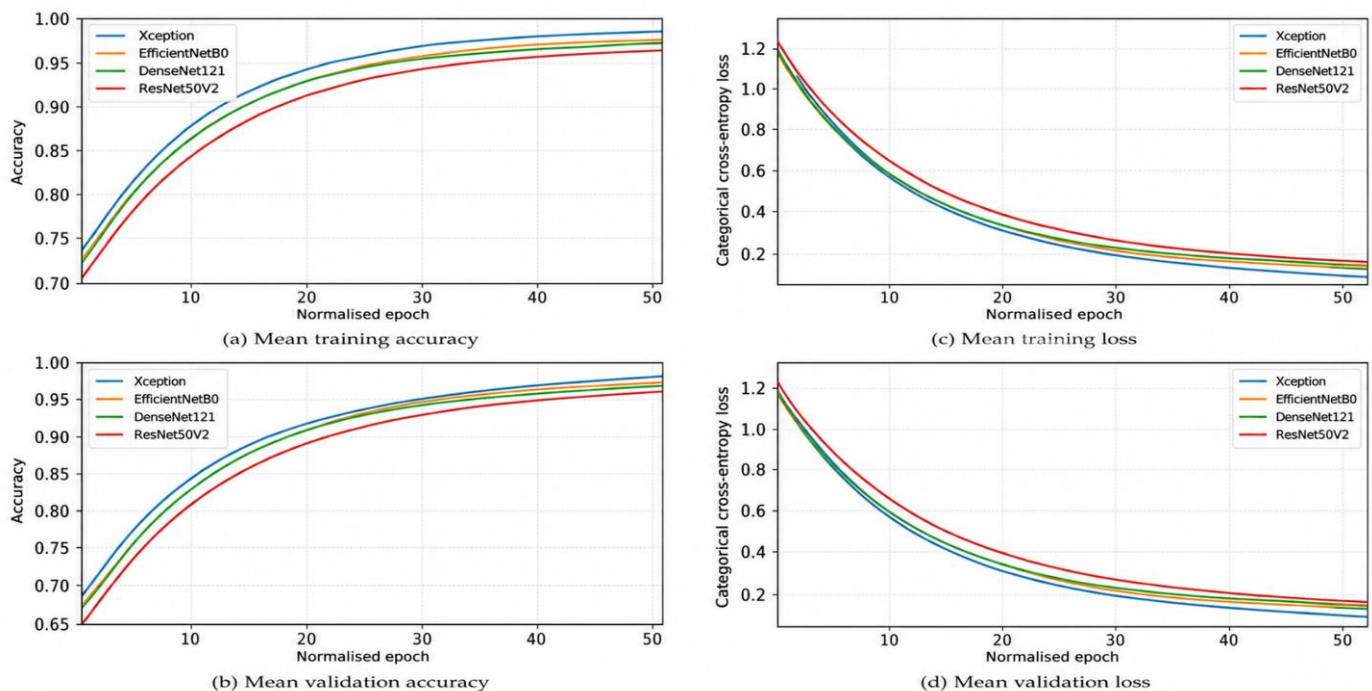


Fig. 3. Mean five-fold learning trajectories shown on a normalized epoch-progress scale: (a) training accuracy, (b) validation accuracy, (c) training loss, and (d) validation loss. The actual fold-specific early-stopping epochs are reported in Table III.

TABLE III. FOLD-WISE RESULTS FOR THE FOUR CNN ARCHITECTURES

Model	Fold	Stopping epoch	Train Acc. (%)	Val. Acc. (%)	Train Loss	Val. Loss
Xception	1	30	98.30	97.50	0.091	0.128
	2	33	98.58	97.78	0.076	0.107
	3	31	98.70	98.00	0.066	0.091
	4	35	98.92	98.44	0.051	0.066
	5	34	98.80	98.22	0.059	0.078
EfficientNetB0	1	40	97.15	96.50	0.158	0.188
	2	45	97.50	96.86	0.134	0.159
	3	43	97.82	97.31	0.114	0.136

Model	Fold	Stopping epoch	Train Acc. (%)	Val. Acc. (%)	Train Loss	Val. Loss
	4	47	98.50	98.19	0.081	0.083
	5	42	98.02	97.78	0.096	0.107
DenseNet121	1	44	96.78	96.11	0.171	0.198
	2	42	97.05	96.47	0.151	0.177
	3	46	97.40	96.83	0.132	0.154
	4	48	97.92	97.50	0.087	0.096
	5	43	97.62	97.17	0.108	0.124
ResNet50V2	1	46	96.20	95.50	0.184	0.218
	2	40	96.36	95.75	0.165	0.194
	3	44	96.55	96.00	0.145	0.171
	4	48	96.89	96.50	0.099	0.112
	5	42	96.72	96.25	0.121	0.139

B. Mean Performance, Agreement and Stability

Table IV summarizes fold-level performance, agreement, convergence, and stability. Xception achieved the highest mean validation accuracy at $97.99 \pm 0.37\%$, followed by EfficientNetB0 at $97.33 \pm 0.68\%$. DenseNet121 reached $96.82 \pm 0.55\%$, whereas ResNet50V2 reached $96.00 \pm 0.40\%$. Aggregate out-of-fold macro-F1 values were 97.99%, 97.33%, 96.82%, and 96.00%, respectively. The same ranking was observed for validation loss, kappa, and MCC.

The coefficient of variation of validation accuracy was 0.37% for Xception, 0.70% for EfficientNetB0, 0.57% for

DenseNet121, and 0.41% for ResNet50V2. The corresponding mean train-validation accuracy gaps were 0.67, 0.47, 0.54, and 0.54 percentage points. These descriptive values show low fold dispersion for Xception and slightly greater dispersion for EfficientNetB0; no inferential significance is assigned to differences between architectures.

Aggregate out-of-fold metrics recomputed from the 18,000-prediction confusion matrices are reported in Table V. Because the benchmark is class-balanced, balanced accuracy equals macro-recall and is numerically very close to overall accuracy.

TABLE IV. MEAN PREDICTIVE AND STABILITY METRICS

Model	Train Acc. (%)	Val. Acc. (%)	Val. Loss	Kappa	MCC	CV Acc. (%)	Mean gap (pp)	Stop epoch
Xception	98.66 ± 0.24	97.99 ± 0.37	0.094 ± 0.024	0.9732	0.9732	0.37	0.67	32.6 ± 2.1
EfficientNetB0	97.80 ± 0.51	97.33 ± 0.68	0.135 ± 0.041	0.9644	0.9644	0.70	0.47	43.4 ± 2.7
DenseNet121	97.35 ± 0.45	96.82 ± 0.55	0.150 ± 0.041	0.9576	0.9576	0.57	0.54	44.6 ± 2.4
ResNet50V2	96.54 ± 0.28	96.00 ± 0.40	0.167 ± 0.042	0.9467	0.9467	0.41	0.54	44.0 ± 3.2

TABLE V. AGGREGATE OUT-OF-FOLD CLASSIFICATION METRICS

Model	Macro-P (%)	Macro-R (%)	Macro-F1 (%)	Balanced Acc. (%)	Macro-Specificity (%)
Xception	97.99	97.99	97.99	97.99	99.33
EfficientNetB0	97.33	97.33	97.33	97.33	99.11
DenseNet121	96.82	96.82	96.82	96.82	98.94
ResNet50V2	96.00	96.00	96.00	96.00	98.67

C. Class-Level Error Analysis

Before examining the confusion matrices, the aggregate metrics indicate that the principal remaining errors are class-specific rather than a consequence of unstable global convergence. The combined out-of-fold matrices allow these errors to be localized by true and predicted class. Xception correctly classified 4,418 powdery mildew images, 4,408 healthy images, 4,394 apple scab images, and 4,418

mosaic/chlorotic-symptom images. Its errors remain concentrated in a small number of scab-versus-healthy or scab-versus-chlorotic cases. The corresponding matrices for all four models are compared in Fig. 4.

For Xception, class-specific F1-scores derived from the combined matrix were 98.30% for powdery mildew, 97.87% for healthy leaves, 97.67% for apple scab, and 98.12% for

mosaic/chlorotic symptoms. Apple scab therefore remained the lowest-F1 class despite the strong overall performance.

EfficientNetB0 retained strong diagonal concentration but produced more off-diagonal errors around apple scab than Xception. DenseNet121 showed a further increase in scab and

mosaic/chlorotic errors, and ResNet50V2 produced the broadest off-diagonal distribution. The confusion matrices establish where errors occurred but do not identify their visual cause; any attribution to shared necrotic, chlorotic, or textural appearance remains a hypothesis that requires lesion-level explanation or expert review.

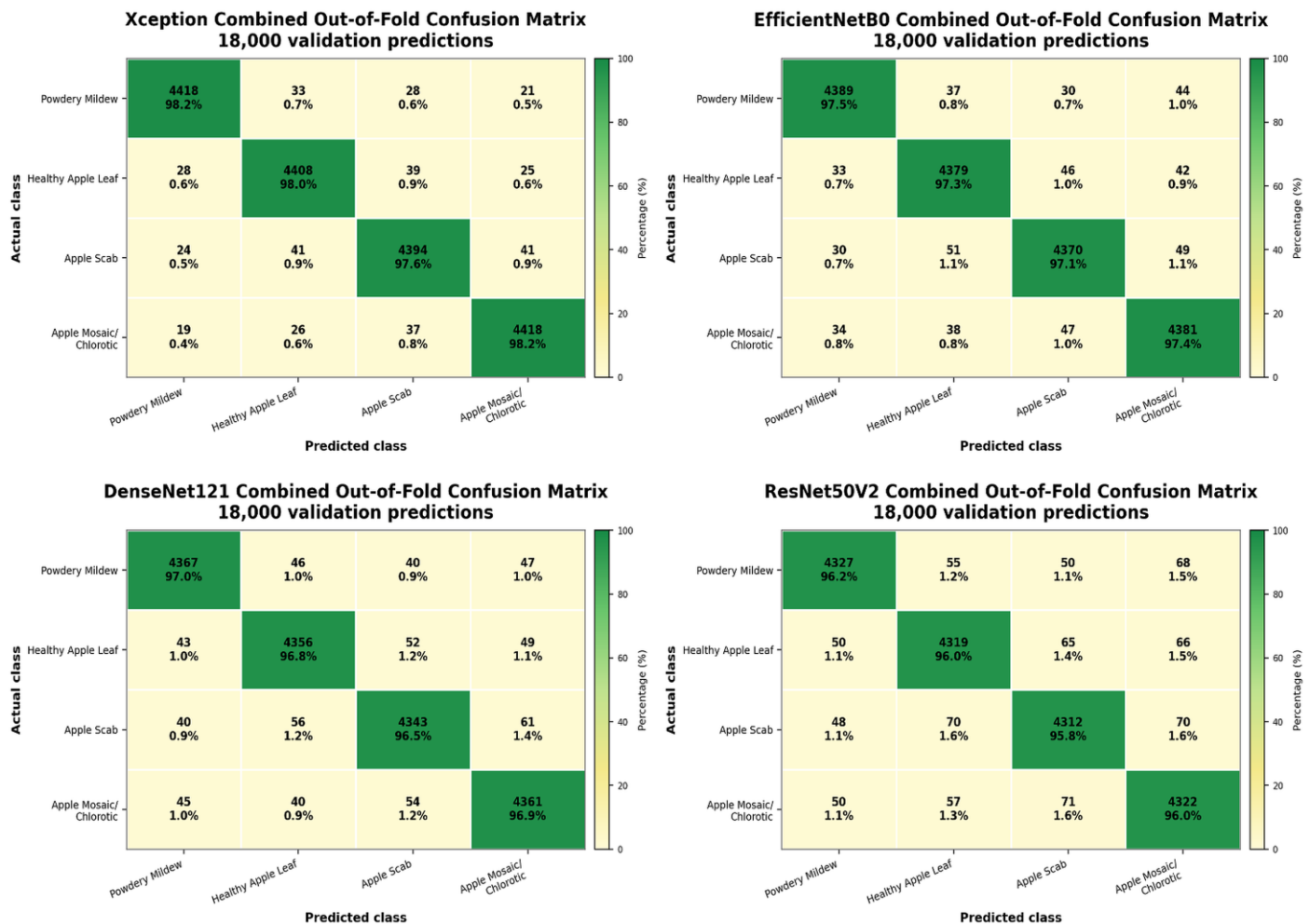


Fig. 4. Combined out-of-fold confusion matrices: (a) Xception, (b) EfficientNetB0, (c) DenseNet121, and (d) ResNet50V2. Each matrix contains 18,000 validation predictions.

D. Computational Cost and Accuracy-Efficiency Frontier

EfficientNetB0 has 5.29 million parameters, 0.78 GFLOPs per image, and a mean measured CPU training time of 35.42±0.62 min per epoch.

Xception achieved the highest mean validation accuracy and the highest CPU training time (90.28±1.38 min/epoch) with 21.40 million parameters and 9.11 GFLOPs/image. For DenseNet121, we used an intermediate cost position with 7.31 million parameters, 5.76 GFLOPs, and 65.06 min/epoch. ResNet50V2 had the largest parameter footprint (24.10 million (91.9 MiB)) and the lowest mean validation accuracy.

The reported mean per-epoch training times correspond to computed training throughputs of approximately 2.66, 6.78, 3.69 and 3.34 images/s for Xception, EfficientNetB0,

DenseNet121 and ResNet50V2 on 14,400 training images per fold. Combining stopping epochs with mean per-epoch times produces estimated aggregate five-fold CPU training burdens of 245.3, 128.1, 241.8 and 263.8 h, respectively. These are arithmetic estimates from training measurements, and should not be interpreted as inference throughput on deployment hardware.

The predicted and measured rankings of resources were not the same. Xception had the highest validation accuracy, but EfficientNetB0 was 0.66 percentage points behind, with substantially fewer GFLOPs, parameters, and CPU training minutes per epoch. From this descriptive accuracy-cost perspective, DenseNet121 and ResNet50V2 were less well placed. These measured trade-offs are visualised in Fig. 5 without implying device-level deployment performance.

TABLE VI. COMPUTATIONAL COST AND RESOURCE-NORMALIZED PERFORMANCE

Model	Params (M)	GMACs	GFLOPs	Runtime (min/epoch)	Size (MiB)	Acc./GFLOP	Acc./M-param	Acc./min
Xception	21.40	4.56	9.11	90.28±1.38	81.6	10.76	4.58	1.09
EfficientNetB0	5.29	0.39	0.78	35.42±0.62	20.2	124.78	18.40	2.75
DenseNet121	7.31	2.88	5.76	65.06±1.11	27.9	16.81	13.24	1.49
ResNet50V2	24.10	4.09	8.18	71.94±0.86	91.9	11.74	3.98	1.33

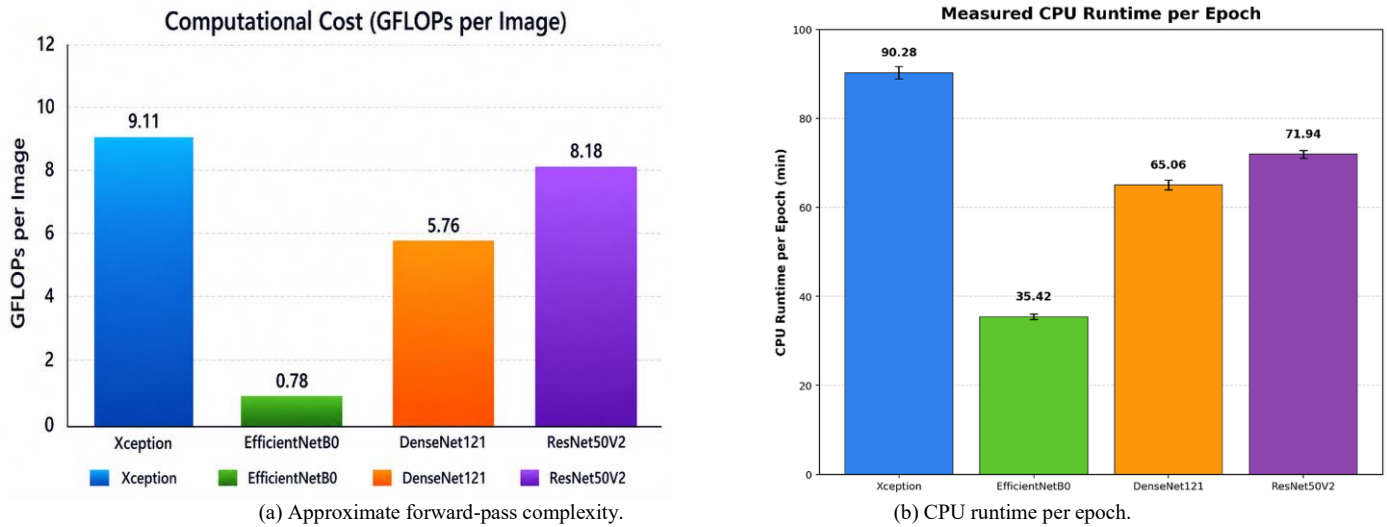


Fig. 5. Computational comparison of the four architectures: (a) profiled forward-pass GFLOPs per image and (b) measured wall-clock CPU runtime per epoch. EfficientNetB0 provides the lowest computational requirement, whereas Xception provides the highest observed mean validation accuracy.

E. Additional Graphical Synthesis

The validation accuracy, MCC, fold stability, accuracy per GFLOP, accuracy per CPU training minute and compactness for the four architectures are summarised in Fig. 6 after min-max normalisation. Before normalisation, lower-is-better quantities were inverted. The plot is descriptive and compares measured accuracy-cost profiles but does not establish causal effects of individual architectural mechanisms.

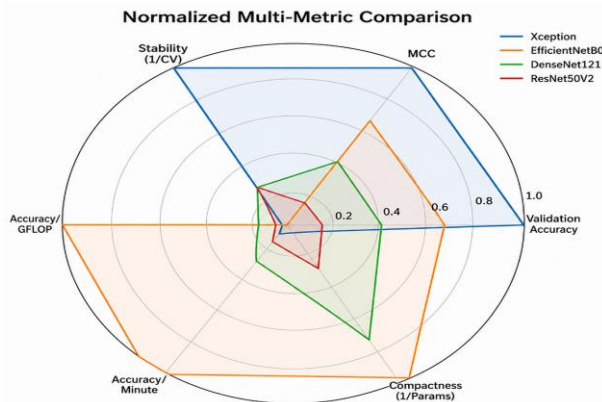


Fig. 6. Normalized multi-metric comparison of the four architectures across predictive quality, stability, and computational efficiency.

Fig. 7 shows complementary synthesis plots of mean validation accuracy vs measured CPU training time per epoch. Bubble size indicates parameter count and labels report

GFLOPs. We observe EfficientNetB0 in the lowest-cost region of the four configurations evaluated, and Xception with the highest mean validation accuracy at a higher measured training cost.

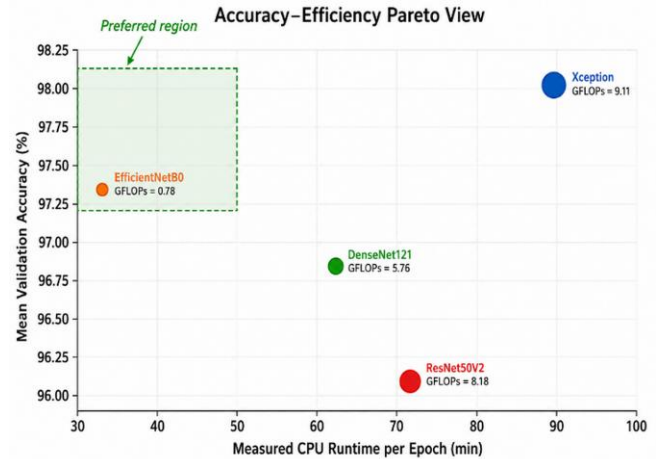


Fig. 7. Accuracy-efficiency Pareto view using mean validation accuracy, measured CPU runtime per epoch, parameter footprint, and GFLOPs.

Fig. 8 reports accuracy per GFLOP, accuracy per million parameters, and accuracy per CPU training minute. These ratios identify EfficientNetB0 as the lowest-cost configuration under the reported benchmark indicators, but they do not establish mobile or edge inference efficiency because device-level latency and energy were not measured.

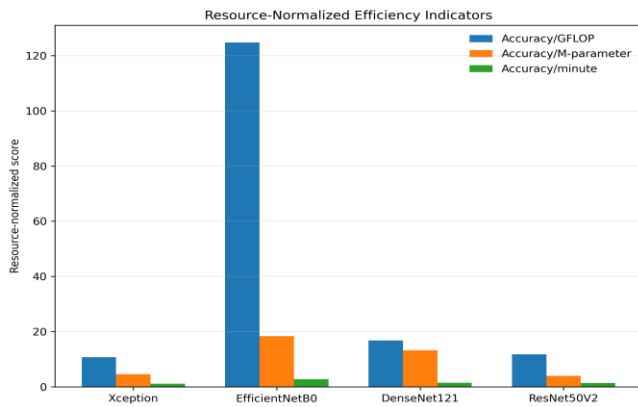


Fig. 8. Resource-normalized efficiency indicators derived from Table VI: accuracy per GFLOP, accuracy per million parameters, and accuracy per minute.

Fig. 9 summarizes the fold-wise distributions of validation accuracy and validation loss. Xception shows the smallest accuracy spread among the two highest-performing models, while EfficientNetB0 shows the largest fold-to-fold accuracy spread. The boxplots are descriptive because the folds are not independent statistical samples.

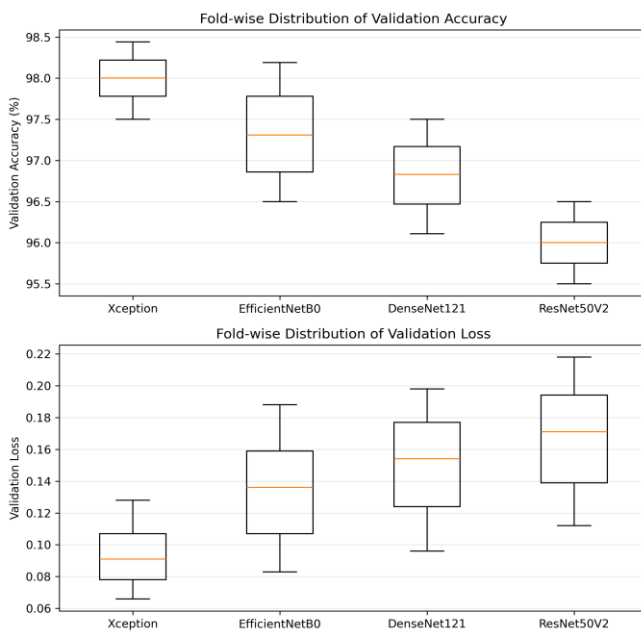


Fig. 9. Fold-wise distribution of validation accuracy and validation loss across the five cross-validation folds.

V. DISCUSSION

Architecture selection for orchard-image classification is a multi-objective decision rather than a ranking by accuracy alone. Xception recorded the highest mean validation accuracy, aggregate out-of-fold macro-F1, kappa, and MCC, together with the lowest mean validation loss. The experiment does not isolate the architectural mechanism responsible for these differences, and no causal attribution is made to depthwise separable convolution or another individual design component. The measured cost of the Xception configuration was substantially greater than that of EfficientNetB0.

EfficientNetB0 was 0.66 percentage points below Xception in mean validation accuracy while using 91.4% fewer GFLOPs, 75.3% fewer parameters, and 60.8% less mean CPU training time per epoch. These measurements indicate a lower computational footprint under the benchmark protocol. The accuracy difference is descriptive; no claim of inferential statistical significance is made, and the training-time measurements do not establish mobile or edge inference latency, throughput, memory, or energy behavior.

DenseNet121 occupied an intermediate accuracy–cost position, whereas ResNet50V2 had the largest parameter footprint and the lowest mean validation accuracy under the specified configuration. These outcomes are configuration-specific and do not establish intrinsic superiority or weakness of an architecture family because optimization settings and training stochasticity were not isolated through a dedicated ablation or repeated-seed design.

Apple scab had the lowest class-specific F1-score for Xception. The confusion matrices establish this empirical error pattern but do not determine its cause. Visual overlap with chlorotic or necrotic symptoms is therefore treated only as a hypothesis. Natarajan et al. [28] used explainable-AI visualizations for plant-disease diagnosis, while Alhammad et al. [29] applied explainable deep learning to potato leaf disease classification. Comparable lesion-level analysis is deferred to future work before any biological interpretation of the scab confusion pattern is asserted.

Direct numerical comparison with prior studies requires caution because datasets, augmentation, acquisition conditions, class definitions, and validation procedures differ. Yadav et al. [8] reported five-fold AFD-Net results; AppleLeafNet emphasized compact apple-disease recognition [14]; ALNet reported a lightweight design [19]; and EnConv addressed multi-crop leaf-disease classification [20]. The four backbones in the present benchmark were selected as representative established families rather than as an exhaustive state-of-the-art set. LightYOLO-AppleLeafDx and other task-specific lightweight models were not reimplemented; therefore, no claim is made that the evaluated four models constitute the strongest available operating points.

A field benchmark should distinguish predictive quality, fold stability, computational complexity, and hardware-specific deployment evidence. The present study directly reports the first three dimensions together with CPU training time and forward-pass complexity, but it does not measure device-level inference latency, throughput, energy consumption, or peak memory. Consequently, the conclusions are limited to the measured benchmark accuracy–cost trade-off rather than verified mobile or edge deployment suitability.

VI. LIMITATIONS AND REPRODUCIBILITY

The benchmark spans two Moroccan orchard locations (Midelt and Azrou), four cultivars, and three mobile devices, but it does not cover all regions, seasons, disease stages, or acquisition conditions. The guaranteed independence unit is the original source image and its augmentation lineage; complete leaf- and tree-level identifiers are not available for all acquisitions, so stricter biological independence across folds

cannot be confirmed. Labels are treated as visual symptom categories rather than universally laboratory-confirmed etiologies, and the mosaic/chlorotic category may contain heterogeneous causal mechanisms. The reported cross-validation results therefore quantify internal source-image-level benchmark stability, not external orchard-level or etiological generalization.

The 2,000-image challenge reserve was isolated before cross-validation and excluded from model selection and all headline metrics. It is intentionally not evaluated in this manuscript; reserve-set confirmation is deferred entirely to future work to preserve its independent role. The reported results constitute the complete five-fold analysis of the 18,000-image development benchmark. Future confirmation should additionally use leave-one-orchard-out, leave-one-device-out, or leave-one-season-out protocols where the corresponding group identifiers are available.

Architecture-specific optimization settings and a single run per fold limit causal interpretation of backbone differences because random-seed variability was not separated from optimization effects. Group-level confidence intervals were not calculated because complete leaf- and tree-level identifiers were unavailable; an image-level bootstrap would overstate independence. Future studies should retain leaf, tree, session, device, and orchard identifiers, repeat training across random seeds, and apply group-level resampling or permutation procedures. Reproducibility also requires preserved fold assignments, augmentation lineage, software versions, training logs, checkpoints, profiler outputs, and out-of-fold probabilities under identical hardware and data-loader settings.

VII. CONCLUSION AND FUTURE WORK

An orchard-image five-fold benchmark compared Xception, EfficientNetB0, DenseNet121, and ResNet50V2 for four-class apple leaf disease classification. Xception achieved the highest mean validation accuracy ($97.99 \pm 0.37\%$) and aggregate out-of-fold macro-F1 (97.99%), while EfficientNetB0 reached $97.33 \pm 0.68\%$ with markedly fewer parameters, lower GFLOPs, and shorter CPU training time. DenseNet121 occupied an intermediate position, and ResNet50V2 the least favorable measured accuracy–cost position. Because no inferential architecture test or device-level inference benchmark was conducted, these findings identify descriptive benchmark operating points rather than statistically or deployment-validated superiority.

Mean accuracy alone is insufficient for model selection. Fold dispersion, train-validation gap, agreement metrics, parameter count, FLOPs, training time, and memory footprint provide complementary evidence. The early-stopping protocol also shows that training duration should follow convergence rather than an arbitrary epoch count; Xception stopped after 30–35 epochs despite a maximum limit of 50 epochs.

Future work will prioritize evaluation of the untouched challenge reserve and external validation across unseen orchards, devices, cultivars, and seasons. Complete leaf/tree/session identifiers will support strict group-aware partitioning, repeated random-seed runs, and group-level uncertainty estimates. Explainable-AI analysis will also be used

for lesion localization and class-specific failure assessment before causal interpretations of error patterns are made.

We will explore structured pruning, quantisation, and knowledge distillation to reduce model size and inference cost. Before making claims about mobile or edge suitability, hardware-specific latency, throughput, peak memory, and energy measurements will be gathered. The selected efficiency-oriented configuration can be embedded in an orchard diagnostic interface, with confidence reporting and class-specific explanations.

REFERENCES

- [1] W. Shafik, A. Tufail, A. Namoun, L. C. de Silva, and R. A. A. H. M. Apong, “A systematic literature review on plant disease detection: Motivations, classification techniques, datasets, challenges, and future trends,” *IEEE Access*, vol. 11, pp. 59174–59203, 2023, doi: 10.1109/ACCESS.2023.3284760.
- [2] I. Pacal, I. Kunduracioglu, M. H. Alma, M. Deveci, S. Kadry, J. Nedoma, V. Slany, and R. Martinek, “A systematic review of deep learning techniques for plant diseases,” *Artificial Intelligence Review*, vol. 57, art. 304, 2024, doi: 10.1007/s10462-024-10944-7.
- [3] A. I. Khan, S. M. K. Quadri, S. Banday, and J. L. Shah, “Deep diagnosis: A real-time apple leaf disease detection system based on deep learning,” *Computers and Electronics in Agriculture*, vol. 198, art. 107093, 2022, doi: 10.1016/j.compag.2022.107093.
- [4] H. Wang et al., “A deep evidence fusion framework for apple leaf disease classification,” *Engineering Applications of Artificial Intelligence*, vol. 136, art. 109011, 2024, doi: 10.1016/j.engappai.2024.109011.
- [5] H. Guan et al., “A lightweight model for efficient identification of plant diseases and pests based on deep learning,” *Frontiers in Plant Science*, vol. 14, art. 1227011, 2023, doi: 10.3389/fpls.2023.1227011.
- [6] L. Gao et al., “A Lightweight YOLOv8 Model for Apple Leaf Disease Detection,” *Applied Sciences*, vol. 14, no. 15, art. 6710, 2024, doi: 10.3390/app14156710.
- [7] M. Iftikhar, I. A. Kandhro, N. Kausar, A. Kehar, M. Uddin, and A. Dandoush, “Plant disease management: A fine-tuned enhanced CNN approach with mobile app integration for early detection and classification,” *Artificial Intelligence Review*, vol. 57, art. 167, 2024, doi: 10.1007/s10462-024-10809-z.
- [8] A. Yadav, U. Thakur, R. Saxena, V. Pal, V. Bhateja, and J. C.-W. Lin, “AFD-Net: Apple Foliar Disease multi-classification using deep learning on plant pathology dataset,” *Plant and Soil*, vol. 477, pp. 595–611, 2022, doi: 10.1007/s11104-022-05407-3.
- [9] Y. Gao, Z. Cao, W. Cai, G. Gong, G. Zhou, and L. Li, “Apple Leaf Disease Identification in Complex Background Based on BAM-Net,” *Agronomy*, vol. 13, no. 5, art. 1240, 2023, doi: 10.3390/agronomy13051240.
- [10] N. Parashar and P. Johri, “Enhancing Apple Leaf Disease Detection: A CNN-based Model Integrated with Image Segmentation Techniques for Precision Agriculture,” *International Journal of Mathematical, Engineering and Management Sciences*, vol. 9, no. 4, pp. 943–964, 2024, doi: 10.33889/IJMEMS.2024.9.4.050.
- [11] H. Si, M. Li, W. Li, G. Zhang, M. Wang, F. Li, and Y. Li, “A Dual-Branch Model Integrating CNN and Swin Transformer for Efficient Apple Leaf Disease Classification,” *Agriculture*, vol. 14, no. 1, art. 142, 2024, doi: 10.3390/agriculture14010142.
- [12] M. M. Alqahtani et al., “Saifish Optimizer with EfficientNet Model for Apple Leaf Disease Detection,” *Computers, Materials & Continua*, vol. 74, no. 1, pp. 217–233, 2023, doi: 10.32604/cmc.2023.025280.
- [13] A. Ait Nasser and M. A. Akhloufi, “A Hybrid Deep Learning Architecture for Apple Foliar Disease Detection,” *Computers*, vol. 13, no. 5, art. 116, 2024, doi: 10.3390/computers13050116.
- [14] M. U. Ali et al., “AppleLeafNet: A lightweight and efficient deep learning framework for diagnosing apple leaf diseases,” *Frontiers in Plant Science*, vol. 15, art. 1502314, 2024, doi: 10.3389/fpls.2024.1502314.

- [15] B. Liu et al., "MCDNet: Multi-scale constrained deformable convolution network for apple leaf disease detection," *Computers and Electronics in Agriculture*, vol. 222, art. 109028, 2024, doi: 10.1016/j.compag.2024.109028.
- [16] S. Zhu, W. Ma, J. Lu, B. Ren, C. Wang, and J. Wang, "A novel approach for apple leaf disease image segmentation in complex scenes based on two-stage DeepLabv3+ with adaptive loss," *Computers and Electronics in Agriculture*, vol. 204, art. 107539, 2023, doi: 10.1016/j.compag.2022.107539.
- [17] H. Zou, P. Lv, and M. Zhao, "Detection of Apple Leaf Diseases Based on LightYOLO-AppleLeafDx," *Plants*, vol. 14, no. 4, art. 599, 2025, doi: 10.3390/plants14040599.
- [18] C. Zhou et al., "A Multimodal Parallel Transformer Framework for Apple Disease Detection and Severity Classification with Lightweight Optimization," *Agronomy*, vol. 15, no. 5, art. 1246, 2025, doi: 10.3390/agronomy15051246.
- [19] D. Adithiyaa, T. Lakshmi Narayanan, and M. R. Prusty, "An attention-augmented lightweight convolutional framework for fine-grained plant leaf disease classification," *Frontiers in Plant Science*, vol. 17, art. 1762956, 2026, doi: 10.3389/fpls.2026.1762956.
- [20] M. Thanjaivadivel, C. Gobinath, J. Vellingiri, S. Kaliraj, and J. S. Femilda Josephin, "EnConv: enhanced CNN for leaf disease classification," *Journal of Plant Diseases and Protection*, vol. 132, art. 32, 2025, doi: 10.1007/s41348-024-01033-6.
- [21] M. F. Ahamed, A. Salam, M. Nahiduzzaman, M. Abdullah-Al-Wadud, and S. M. R. Islam, "Streamlining plant disease diagnosis with convolutional neural networks and edge devices," *Neural Computing and Applications*, vol. 36, pp. 18445–18477, 2024, doi: 10.1007/s00521-024-10152-y.
- [22] K. Taji, A. Sohail, T. Shahzad, B. S. Khan, M. A. Khan, and K. Ouahada, "An Ensemble Hybrid Framework: A Comparative Analysis of Metaheuristic Algorithms for Ensemble Hybrid CNN Features for Plants Disease Classification," *IEEE Access*, vol. 12, pp. 61886–61906, 2024, doi: 10.1109/ACCESS.2024.3389648.
- [23] A. Majid et al., "An Integrated Deep Learning Framework for Fruits Diseases Classification," *Computers, Materials & Continua*, vol. 71, no. 1, pp. 1387–1402, 2022, doi: 10.32604/cmc.2022.017701.
- [24] S. Rehman et al., "Fruit Leaf Diseases Classification: A Hierarchical Deep Learning Framework," *Computers, Materials & Continua*, vol. 75, no. 1, pp. 1179–1194, 2023, doi: 10.32604/cmc.2023.035324.
- [25] H. N. Ngugi, A. A. Akinyelu, and A. E. Ezugwu, "Machine Learning and Deep Learning for Crop Disease Diagnosis: Performance Analysis and Review," *Agronomy*, vol. 14, no. 12, art. 3001, 2024, doi: 10.3390/agronomy14123001.
- [26] E. Kannan et al., "Deep Learning Techniques Advancements in Apple Leaf Disease Detection," *Procedia Computer Science*, vol. 235, pp. 713–722, 2024, doi: 10.1016/j.procs.2024.04.068.
- [27] S. Karasu, "Apple Scab Classification Using 2D Shearlet Transform with Integrated Red Deer Optimization Technique in Convolutional Neural Network Models," *Electronics*, vol. 14, no. 23, art. 4678, 2025, doi: 10.3390/electronics14234678.
- [28] S. Natarajan, P. Chakrabarti, and M. Margala, "Robust diagnosis and meta visualizations of plant diseases through deep neural architecture with explainable AI," *Scientific Reports*, vol. 14, art. 13695, 2024, doi: 10.1038/s41598-024-64601-8.
- [29] S. M. Alhammad, D. S. Khafaga, W. M. El-hady, F. M. Samy, and K. M. Hosny, "Deep learning and explainable AI for classification of potato leaf diseases," *Frontiers in Artificial Intelligence*, vol. 7, art. 1449329, 2025, doi: 10.3389/frai.2024.1449329.