Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Beyond Accuracy Rankings: Cross-Dataset Stability and Computational Cost in Plant Disease Classification

Nitin Kumar

kumarniti2004@gmail.com

Lovely Professional University https://orcid.org/0009-0007-3661-859X

Dr. Sakshi Dua

sakshi.26557@lpu.co.in

Lovely Professional University https://orcid.org/0000-0002-8143-2168

Research Article

Keywords: Plant disease classification, cross-dataset eval uation, transfer learning, convolutional neural networks, macro-F1, weighted-F1, computational efficiency, PlantDoc.

Posted Date: November 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11675885/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Beyond Accuracy Rankings: Cross-Dataset Stability and Computational Cost in Plant Disease Classification

Nitin Kumar[1], Dr. Sakshi Dua[2]

***Abstract*— On a specific dataset, achieving high accuracy in plant disease classification with CNN is possible, but it should also be evaluated on other datasets and the accuracy of the CNNs depend on the cost as well. This study tries to compare the performance of ResNet-50, MobileNetV2 and EfficientNet-B0 architectures with frozen feature transfer learning approach. The 38-class New Plant Diseases Dataset (Augmented) has been used to train and internally validate models consisting of 70,295 images from which 56,236 have been used to train the models and 14,059 used for tests. A total of 146 images of the PlantDoc were used for external evaluation for 17 classes. The accuracy, macro-F1, and weighted-F1 score were used to measure the classification performance, while the number of the parameters, batch-one inference latency and throughput were used as measurement for the computational efficiency. ResNet-50 got the best internal accuracy score (96.33%) and macro-F1 (96.22%). The best accuracy (21.23%) was obtained by ResNet-50 while the best macro-F1 (22.48%) and measured inference latency (4.97 ms) was obtained by MobileNetV2 on PlantDoc. In terms of measured throughput, MobileNetV2 had the lowest number of parameters (2.27 million) and the highest throughput of 201.2 images/s. The pairwise McNemar tests did not reveal any statistically significant differences with respect to the prediction of correctness. There is a restriction of the external sample and hardware specific benchmarking leading to limitation of generalization. The results suggest that it is worthwhile to combine a measure of predictive performance with a measure of computational efficiency.**

***Index Terms*— Plant disease classification, cross-dataset evaluation, transfer learning, convolutional neural networks, macro-F1, weighted-F1, computational efficiency, PlantDoc.**

I. INTRODUCTION

Proper time and proper identification of plant disease is crucial for management in agriculture, as plant diseases can be a major problem to yield. Current disease diagnosis is mainly based on visual diagnosis by farmers or agricultural experts, which may be influenced by human experience, similarity of disease symptom, etc. and affected by the environment. Thus, computer vision and deep learning have been a great subject of interest as automated plant disease recognition tools. The development of image-based classification systems with large image collections of healthy and diseased leaves has been facilitated by available public image repositories, such as the PlantVillage image repository [1], [2].

*This work was not supported by any organization

[1]N. Kumar is with the School of Computer Applications, Lovely Professional University, Phagwara, Punjab, India kumarniti2004@gmail.com

[2]S. Dua is with the School of Computer Applications, Lovely Professional University, Phagwara, Punjab, India sakshi_nancy@yahoo.in

Convolutional neural networks (CNNs) have shown to be effective in plant disease classification. In the work of Mohanty et al. [2], 99.35% accuracy was achieved on 54,306 images from PlantVillage dataset by training a deep CNN network on it. Later, Ferentinos et al. [3] tested the deep learning models on 87,848 images representing 58 combinations of plants and diseases, and achieved an accuracy of 99.53% on previously unseen images. These results exhibit the capability of plant disease recognition based on deep learning using images. But a high accuracy on a specific evaluation dataset is not guaranteed that it will do as well when using images taken under other circumstances.

An important challenge is that of image collections that are controlled versus images taken under realistic agricultural conditions. Numerous photographs of plants taken in the field against variable backgrounds with complex textures such as occlusion, and variations in image quality, may show quite different color characteristics from those taken with a background of uniform texture and lighting. Barbedo et al. [4] mentioned complexity of the background, lack of control over the conditions of capturing images and similarity of the symptoms of the disease as the biggest challenges in planting disease automatic identification. Singh et al. [5] developed a PlantDoc dataset of images taken in non-laboratory conditions that is expected to be suitable for plant disease recognition in less controlled field conditions. Such difficulties make it worth investigating models based on data other than validation data.

In addition to computational issues there are also considerations in model selection. There are multiple architectures that can deliver good predictive performance, but they vary in the number of parameters, inference time, throughput and memory consumption. These differences are significant for choosing models of limited computational resources for application. Efficient deep learning inference and hardware-aware architecture design has been studied indicating that the performance of the inference is not just determined by the size of the model [6], [7]. In this way, not only is the performance of the predictive model assessed but also the computational cost measured, giving more information to compare different architectures of CNN.

This paper aims to explore how well classification models perform on other datasets and differences in performance between three models – ResNet-50, MobileNetV2 and EfficientNet-B0 – and the efficiency of the computation. Same frozen feature transfer learning protocol used to evaluate the models. The New Plant Diseases Dataset (Augmented) is used for the training and internal validation

and has 38 classes. Mapped subset of 146 images of this PlantDoc dataset was used for external evaluation from 17 mapped classes. The accuracy, macro-F1 scores and weighted-F1 scores are used to evaluate the classification performance, whereas the total number of parameters, batch-one inference latency, throughput and peak GPU memory usage are used to evaluate the computational efficiency. Two-by-two comparisons of correct vs. incorrect predictions are also performed to analyze the differences in the model output for the external evaluation sample.

The experimental results demonstrate that the relative advantage of the three architectures vary with the evaluation setting and metric used. The ResNet-50 model has the best internal validation performance with a significant drop in the classification performance on the mapped PlantDoc subset for all three models. MobileNetV2 offers the best measured computational efficiency profile of the models tested, having both the least amount of parameters and the slowest latency, and the highest measured throughput. The results indicate that model evaluation and computing requirements should be considered instead of using only the validation accuracy of the model within itself.

The work of this study has three major contributions. First, it gives an equal comparison of ResNet-50, MobileNetV2 and EfficientNet-B0 using a common frozen-feature transfer learning protocol. Second, it compares source-dataset validation versus external evaluation on a mapped PlantDoc subset; highlighting the need to look at external evaluation as well as using the internal evaluation as a valid method. Third, it compares the three architectures based on complementary classification and inference measures paying attention to their respective predictive-performance and computational-efficiency trade-offs. The study is not meant to be used as a suggestion for a new network architecture, nor is it meant to determine the cause of decreased performance across the two different sets of data.

II. Related Work

Although deep learning has been widely applied in the classification of plant diseases with satisfying results, subsequent under different testing conditions may lead to a performance drop. Moupojou et al. [8] released FieldPlant, an image database of 5,170 images that were collected from the field and 8,629 annotated leaves in 27 disease classes. Ahmad et al. [9] tested five differently pretrained CNN models on five corn-disease datasets suggesting that when combined with images from lab and field or background removal, there can be an improvement in generalization across datasets. These studies illustrate the need to have realistic data and to use external evaluation to assess the reliability of plant disease classifiers.

Other approaches to attaining domain differences have been studied. To enhance plant disease recognition across multiple datasets, Wu et al. [10] tried to use multiple feature representations with the help of uncertainty regularization and unsupervised domain adaptation network. Yang et al. [11] measured the recognition performance of cross-domain few-shot learning, and concluded that the similarity of source–target domains and the number of target examples influence the learning performance. Conceptually, these methods explore the domain adaptation or few-shot learning scenarios, whereas the present study evaluates the performance of pretrained CNNs in an equal training protocol without changes and adaptations to the models for the target dataset.

More general reviews of the development of plant disease recognition have recently been conducted. To date, Mallick et al. [12] have analyzed a number of deep learning techniques such as architecture modifications and vision transformers and Upadhyay et al. [13] have explored over 278 studies, encompassing CNNs, transformers and various imaging technologies. Other examples include the rice-leaf classification that achieved 97% validation accuracy with augmentation with realistic designs and EfficientNet-B5 [14] and jute-leaf classification that achieved 94% accuracy with the transfer learning of ResNet-50 [15] for a binary leaf classification task. In this regard, Joshi [16] explored a combination of multi-task vision pipeline for analysing disease and orange classification with the prime concern of computational efficiency. None of the studies, however, used the same set of data and/or utilized the same class definitions and/or the same evaluation protocol to obtain their results, so direct comparison with the results of the present experiment are not possible.

In the field of agriculture, AI can significantly boost productivity and decision-making capabilities, but it also comes with a cost that needs to be taken into account. Junaidi et al. [17] examined the deep learning and edge computing for the rice disease detection, addressing the inference latency and deployability constraints. Karkikatti et al. [18] reviewed the use of AI and IoT in precision agriculture, including real-world challenges like infrastructure expenses, edge computing and interpretability of AI models. These studies justify the use of metrics in addition to classification metrics when assessing a model's size and performance in making inferences. Measures taken with different types of equipment and setups, however, must not be compared as they are not directly comparable.

More directly, Mahapatra et al. [19] had built an attention-based CNN for five datasets and obtained the accuracy of 99.38% in an experiment on the same dataset and an average of 82.93% when doing cross-dataset classification for the corn disease. Their work shows training and generalization of cross-datasets are already studied. It is important to note here that this study does not aim to become the first report on the study of cross-dataset degradation but rather the first controlled comparison of the ResNet-50, MobileNetV2 and EfficientNet-B0 models with a frozen-feature transfer-learning paradigm. The models are tested for fitness on a set of source images called the 38-class dataset, and a dataset of mapped images of PlantDoc (146 images classified as 17 classes). In this study, the classifiers' metrics of classification, together with the number of parameters used

(count), inference and throughput, as well as GPU memory allocation are studied, allowing to characterize the predictive-performance and computational-cost trade-offs observed under the specific experimental conditions.

III. METHODOLOGY

In this study, three different convolutional neural network (CNN) architectures, namely ResNet-50, MobileNetV2 and EfficientNet-B0 are investigated for their classification performance and computational efficiency using a comparative experimental design. Experiments are conducted on the New Plant Diseases Dataset (Augmented) for in-hand training and validation of the model and a mapped subset of the PlantDoc for external validaion. All three architectures are initialized with the weights pre-trained on ImageNet and trained with transfer learning approach where the features are frozen. The experimental procedure involves the preparation of the datasets, the training of the model, as well as internal and external testing and analysis of computational efficiency. The common evaluation protocol enables a comparative study of architectures, which also can be examined alongside with computational cost as suggested by comparative CNN studies and hardware-aware neural architecture research [20], [19], [7], [30].

3.1. Datasets and Data Preprocessing

The source data-set used for training and the internal validation set is the Kaggle data-set "New Plant Diseases Dataset (Augmented)" taken from the Kaggle repository: 'vipoooool/new-plant-diseases-dataset'. It includes pictures with images divided into 38 classes (for healthy and disease leaves). The training notebook claims that the random seed value used is 42 for reproducibility and the images are split into training and validation sets with random 80:20 split. The subsets that result have 56,236 training images and 14,059 validation images. Split is not stratified, but rather, indices of the image are randomly divided into split sections, so this is referred to as a "seeded random split. The validation subset is used to evaluate the model performance on source-dataset images which are not used to update model parameters.

The training subset is dynamically enhanced with data augmentation in order to create variability in the appearance of the images. The transformations applied to the training set are 'RandomResizedCrop(224, scale=(0.7,1.0))', random horizontal and vertical flips, 'RandomRotation(45)', and a ColorJitter of brightness, contrast and saturation = 0.3. The random augmentation is not applied to the validation images, they are simply resized to 224×224 and converted to a tensor. Both subsets are normalized using the ImageNet channel-wise mean $[0.485, 0.456, 0.406]$ and standard deviation $[0.229, 0.224, 0.225]$. Dynamic augmentation for training with deterministic validation preprocessing helps for a more consistent measure of model performance. Training strategies have also been studied in the context of transfer learning and the importance of evaluating training strategies under clearly defined experimental conditions is also described [20].

The external evaluation is done using the PlantDoc dataset that is accessible from GitHub repository: pratikkayal/PlantDoc-Dataset. For external validation the number of images in the original evaluation folder is reported as 236 in the external-validation notebook. The selected labels of a PlantDoc folder are mapped under a predefined mapping dictionary (cross_dataset_map) and to the respective classes in the source model's vocabulary of 38 classes. After such mapping and filtering, 146 images will remain that will undergo external evaluation. The mapping is needed, since the class structure and the class folder names differ from the training dataset to the target dataset. Any categories that have no explicit mapping are simply not added to the mapped evaluation set, but they are not explicitly labeled "unsupported".

Individual images from the PlantDoc are resized to $224 \times 224 = 50176$ pixels and converted to tensors and normalized as per the ImageNet mean and std values used in the original images. During external assessment, random train augmentations are not injected. This gives rise to a subset of 146 images, which is then used to evaluate the performance on a set of images with a different source, but not to actually train the models. This evaluation is an empirical comparison of the performance on internal source – dataset versus the performance on the mapped target – dataset. The difference in performance that is observed could however be due to many factors such as image background, illumination, image quality, class composition or the conventions used for annotation; from the comparison alone it cannot be determined which factor was responsible for the difference. A summary of the dataset characteristics, including image counts, cross-dataset evaluation splits, and preprocessing configurations for both the source and target datasets, is provided in Table I.

3.2. Model Architectures and Transfer Learning

In this research, ResNet-50 model, MobileNetV2 model and EfficientNet-B0 model are evaluated to study the relationship between the classification performance, model complexity, and computational efficiency in plant disease classification. The architectures can be used to extract the features in two different ways and have different architecture goals. The great innovations in ResNet-50 include residual connections, which allow information to flow freely through deep networks; inverted residual blocks and depthwise separable convolutions were introduced in MobileNetV2 to lower the computation costs; and compound-scaling in EfficientNet-B0 enables scaling the network depth, width and input resolution. These differences offer a potential basis for comparing different architectures of conventional CNN with those that are computationally efficient. In the past, CNN evaluation and design using resources have been mentioned compared [2], [7].

ImageNet-pretrained weights are used to initialize all models, and they are adapted to the classification task of source-dataset classification (38 classes) using frozen-feature transfer learning. The feature representation, $f_\theta(x)$, is extracted from an input image x, with the pretrained backbone

TABLE I

SUMMARY OF DATASETS USED IN THE EXPERIMENTS

Dataset characteristic	New Plant Diseases Dataset (Augmented)	PlantDoc Dataset
Purpose	Model training and internal validation	Cross-dataset evaluation
Source	Kaggle	GitHub repository
Total images used	70,295	146
Number of classes	38	17 mapped classes
Training images	56,236	Not used for training
Validation images	14,059	Not used for training
Input image size	224×224 pixels	224×224 pixels
Preprocessing	Resize/crop, augmentation during training, normalization	Resize and normalization
Split/selection method	Random 80:20 split, seed 42	Selected subset with mapped labels

parameters θ. The classification head classifies the extracted class representation to class scores (parameterized by W and b):

$$z(x) = W f_\theta(x) + b \tag{1}$$

In this, $z(x) \in \mathbb{R}^{38}$ are logits corresponding to 38 different output classes. The softmax function is used to get the class probabilities:

$$p_k(x) = \frac{\exp(z_k(x))}{\sum_{j=1}^{38} \exp(z_j(x))} \tag{2}$$

(where $p_k(x)$ is the predicted probability in class k). While training, the parameters θ are kept fixed and parameters W and b are updated. This minimizes the parameters to be updated as opposed to updating the whole network. When the classification head is linear the approach is often referred to as frozen-feature transfer learning or linear probing. The impact of different parameter adaption methods on classification performance has been studied using the transfer learning paradigm [20].

The models are trained with the Adam optimizer, learning rate 0.001 and the categorical cross-entropy loss. The loss is given as follows for a set of N images:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{38} y_{ik} \log(p_{ik}) \tag{3}$$

where y_{ik} is the one-hot encoded label for the image i and p_{ik} is the probability of the image i belonging to the class k when using the model. The optimization scheme tunes the classification head to minimise this loss. The internal validation subset is used to monitor training; early stopping set as a patience of 5 epochs. The checkpoint with the best accuracy score when being validated is kept for evaluation. This selection procedure is applied based on the training implementation: the exact number of epochs to take and other settings on the run should be communicated from the notebook configuration.

The complexity of models is characterised by the number of parameters and the number of trainable parameters. The trainable proportion is defined as the ratio of $P_{\text{trainable}}$ to P_{total} (where P_{total} represents the total number of model parameters and $P_{\text{trainable}}$ represents the number of model parameters that are updated during training).

$$R_{\text{trainable}} = \frac{P_{\text{trainable}}}{P_{\text{total}}} \times 100\% \tag{4}$$

This distinction is done because there are parameters that are optimized during training, but frozen parameters remain part of the deployed model; they simply are not updated during training. The computational performance is assessed separately as it is not enough to specify the number of parameters to determine the memory or inference speed.

3.3. Evaluation Metrics

The internal validation subset and mapped PlantDoc external evaluation subset is used to evaluate their classification performance. The confusion matrix provides the accuracy, precision, recall and F1 score. True positives (TP) are the positive examples correctly classified by the model, false positives (FP) are the negative examples classified as positive by the model, and false negatives (FN) are the positive examples incorrectly classified as negative by the model. For multiclass classification, the following quantities are computed with a one-versus-rest interpretation for each class.

Overall accuracy is the percentage of correctly classified images:

$$\text{Accuracy} = \frac{\sum_{k=1}^{C} TP_k}{N} \tag{5}$$

If a total of N images were evaluated and C classes, then the number of evaluated classes is called C. Accuracy gives a general impression, but may mask performance across classes, especially when classes are of unequal size.

The precision and recall for each class k is given by $\text{PRECISION}(k)$ and $\text{RECALL}(k)$, respectively.

$$\text{Precision}_k = \frac{TP_k}{TP_k + FP_k} \tag{6}$$

$$\text{Recall}_k = \frac{TP_k}{TP_k + FN_k} \tag{7}$$

Precision is the percentage of the number of correct predictions made for a class, while recall is the percentage of the number of ground-truth examples correctly predicted for a class. The harmonic mean of the precision and recall is taken for each class and it is referred to as Class-wise F1 Score.

$$F1_k = 2\frac{\text{Precision}_k \cdot \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k} \tag{8}$$

If the denominator is zero, how the metric implementation deals with the division needs to be clearly defined, and applied uniformly.

For class-wise performance, macro-averaged precision, recall and F1-scores are obtained, which are unweighted means of the corresponding scores across the C evaluated classes. Macro-F1 is represented by

$$\text{Macro-F1} = \frac{1}{C}\sum_{k=1}^{C} F1_k \tag{9}$$

Unlike accuracy, macro-F1 treats each evaluated class equally regardless of different counts of samples; it works well when there are different numbers of examples in each class. Weighted-F1 scores each class's F1 score by the number of ground-truth examples in a class (n_k) with the same number:

$$\text{Weighted-F1} = \sum_{k=1}^{C} \frac{n_k}{N} F1_k \tag{10}$$

They are reported all together to give complementary aspects of overall classification performance and class level behavior. These are computed only from the 146 images kept from PlantDoc (for external evaluation). The external metrics therefore describe the performance on this mapped subset of the data, not on all 38 source classes or the entire PlantDoc data set. The number of classes assessed and class support should thus be reported along with the results. There is a distinction between these two that is relevant when comparing the performance of models on different data sets [9]-[11].

3.4. Computational-Efficiency Benchmarking

The number of parameters in the model, the inference latency for batch-one, the inference throughput, and the peak allocation of the GPU memory for the model are used to measure computational efficiency for ResNet-50, MobileNetV2, and EfficientNet-B0 [16]. The benchmark is based on an input tensor of size $(1 \times 3 \times 224 \times 224)$ which corresponds to one RGB image with 224×224 pixels. All models are executed before putting them in evaluation mode, and 50 warm-up forward passes are made before actually timing them to prevent the initial execution overhead from affecting the results. The implementation is based on inference and has no gradient computation and it synchronizes CUDA operations to timing with the presence of a GPU.

Let T be the total amount of time (in seconds) used for forward passes R. To get the average batch-one inference latency, the calculation is described below:

$$L = \frac{T}{R} \times 1000 \tag{11}$$

where L denotes the average inference latency in milliseconds per image, T is the total elapsed time in seconds, and R is the number of forward passes measured. Inference throughput is calculated as

$$Q = \frac{R}{T} \tag{12}$$

where Q denotes inference throughput in images per second. These measurements document single-image model inference in the specific environment in which the images were recorded, and are "batch one" inputs for the benchmark. These do not have a separate image loading/pre-processing time measurement.

To obtain the total number of model parameters, it is necessary to add up the number of parameters in each model layer:

$$P_{\text{total}} = \sum_{j=1}^{M} P_j \tag{13}$$

where P_{total} denotes the total number of model parameters, P_j is the number of parameters in component j, and M is the total number of model components included in the sum. The parameter count is expressed in millions by dividing P_{total} by 10^6. When CUDA is available, the peak GPU memory allocation is recorded in megabytes—otherwise, the GPU memory is reported as unavailable and not interpreted as the zero memory usage.

ResNet-50 is reported to have a benchmark of 6.59ms latency, 151.7 images per second and 23.59 million parameters and 151.04 MB peak GPU memory. MobileNetV2 has been reported to have 2.27 million parameters, 4.97 ms latency, 201.2 images per second and 150.23 MB of peak GPU memory. EfficientNet-B0 has a total of 4.06 million parameters, 8.16 ms latency, 122.6 images per second and a peak GPU memory usage of 150.23 MB. This is the values that are outputted in the training notebook, and they should be interpreted in terms of what it executed. The measurements in particular reflect the implementation used for the benchmarked model and do not represent the performance of the mobile phones and dedicated edge devices.

3.5. The experimental protocol and reproducibility

All the experiments use the same training and evaluation procedure for all three CNN architectures. The original dataset has 70,295 images across 38 classes, including 56,236 images in the training set and 14,059 images in the validation set, which were split based on a seeded random

split with a seed of 42. The batch size for the training and validation data loaders is 32 with shuffling True for the training loader and False for the validation loader. Validation and external-evaluation images are resized deterministically, and trained with normalization from ImageNet while training images are augmented as described in Section 3.1.

All three models are pretrained with the weights from the ImageNet Database and the parameters in the backbone network are fixed during the last classification layer fine-tuning process for the 38-class task. The Adam optimizer, with a learning rate of 0.001, and cross-entropy loss is used for training the models. The training procedure is set up for a maximum of 25 epochs and employed early stopping with the patience of 5 epochs. The checkpoint that performs the best on the in-sample data is saved to be evaluated later. The best checkpoint validation on the in-sample data is saved to be evaluated later. All three training configurations will be consistent to allow for a comparison.

Upon completion of the training, the best checkpoint is tested on the internal validation subset and mapped external-evaluation subset of PlantDoc. External evaluation is carried out with 146 images from which images are selected according to the predefined class-mapping procedure outlined in Section 3.1. Predictive capabilities are measured using accuracy and the class-specific metrics, macro-F1 and weighted-F1, respectively. The computational benchmark features the measurement of model parameters, inference latency, throughput and peak GPU memory allocation. Results from the evaluation are analyzed separately for the source and target data sets since the external subset mapped corresponds to different distribution of images and composition of classes.

The implementation uses 42 as the random seed for splitting the source data; it chooses the CUDA hardware device for computation based on its availability. The reported results should be linked to the model checkpoints, the mapping of the dataset to the model, configuration of the data preprocessing and the experimental setup to perform the benchmark procedure. These reported computational measurements are only applicable to the specific environment of their measurement; any comparison to other measurements of calculations across different hardware or software configurations should include the conditions of the measurement and keep them in mind.

IV. Results & Discussion

4.1. Internal Validation Performance

Three pretrained convolutional neural network (CNN) architectures were tested on the built-in validation set of 14,059 images of the New Plant Diseases Dataset (Augmented). The best checkpoint for each architecture was chosen on the basis of validation accuracy. The results that were measured are shown in Table II.

The internal validation results of the three architectures with respect to macro-F1 scores are shown in Fig.1.

ResNet-50 had the best internal validation results; with 96.33%, 96.22%, and 96.32% for accuracy, macroF1, and weightedF1 respectively. EfficientNet-B0 achieved an accuracy of 95.42%, macro-F1 of 95.34%, and weighted-F1 of 95.41%. MobileNetV2 achieved an accuracy of 94.26%, macro-F1 of 94.12%, and weighted-F1 of 94.22%. All three models obtained very good performances in the internal validation, but this does not guarantee the accuracy of the models in terms of generalisation to images from another dataset. External assessment is thus required to investigate their performances in various data scenarios.

TABLE II

Internal validation performance on the source dataset

Model	Accuracy (%)	Macro-F1 (%)	Weighted-F1 (%)
ResNet-50	96.33	96.22	96.32
MobileNetV2	94.26	94.12	94.22
EfficientNet-B0	95.42	95.34	95.41

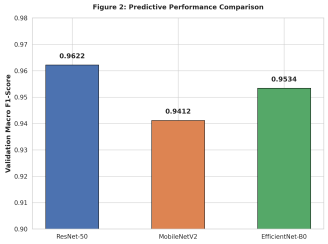


Fig. 1. Internal validation macro-F1 scores of the three CNN architectures.

4.2. Cross-Dataset Evaluation on PlantDoc

To assess the behaviour of trained models when working with different source data, the results for 146 mapped PlantDoc images (17 mapped classes) were tested. Images in this collection were never used for training the models. Table III gives a summary of the external-evaluation results recorded.

Figure 2 compares accuracy, macro-F1, and weighted-F1 across the three architectures on the mapped PlantDoc evaluation subset.

ResNet-50 had the highest possible accuracy of 21.23 % among these, with MobileNetV2 having 20.55 % and EfficientNet-B0 having 18.49 % accuracy. But it is not so when macro-F1 was taken into account. MobileNetV2 achieved the best macro-F1 score of 14.70%, while ResNet-50 and EfficientNet-B0 had scores of 13.18% and 10.10%, respectively.

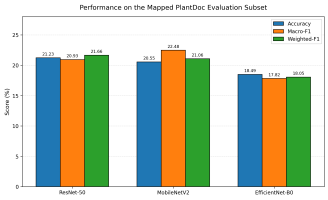


Fig. 2. Classification performance of the three CNN architectures on the mapped PlantDoc evaluation subset.

TABLE III

PERFORMANCE ON THE MAPPED PLANTDOC EVALUATION SUBSET

Model	Accuracy (%)	Macro-F1 (%)	Weighted-F1 (%)
ResNet-50	21.23	20.93	21.66
MobileNetV2	20.55	22.48	21.06
EfficientNet-B0	18.49	17.82	18.05

Actually, the class set used for macro-F1 calculation was a fixed set of 17 mapped classes and classes that are not divided into scores of zero. The accuracy predictions were calculated by simply counting how often predictions were made in classes that were not mapped, and these predictions were not added to the mapped-class F1 scores as positive predictions.

The difference highlights the need to use accuracy and macro-F1 simultaneously. The accuracy is the sum of all the images classified correctly while macro-F1 is the mean of the F1 scores obtained from each class of the images included in the calculation of the macro-F1 score. A model can have a higher overall accuracy but be less consistent in accuracy for individual classes.

There were 17 classes and 146 mapped PlantDoc images for the external evaluation. The results show ResNet-50 with a top accuracy of 21.23%, MobileNetV2 with 20.55% and EfficientNet-B0 with 18.49%. However, MobileNetV2 outperformed ResNet-50 and EfficientNet-B0 with a macro-F1 score of 22.48% while ResNet-50 scored 20.93% and EfficientNet-B0 scored 17.82%. The weighted-F1 scores were 21.66%, 21.06%, and 18.05%, respectively. These results reveal that the relative performance of the models can vary with the evaluation metric; for example, ResNet-50 achieved the highest accuracy while MobileNetV2 achieved the highest macro-F1.

The external assessment results were far behind the internal validation results. The two evaluations, however, are using two different sets of data and group compositions, limiting the ability to conclude that differences are due solely to domain shift. In addition, the internal macro-F1 scores include 38 classes from the source, while the external macro-F1 scores are based on 17 classes of the mapped source. It would be inappropriate, therefore, to make a direct numerical comparison of these macro-F1 values without re-calculating the macro-F1 from the sources and over the same 17 classes. This helps the understanding of the need to test and validate classification models out of their source data set. However, performance variations do not provide sufficient evidence of whether the main reasons were background variation, illumination, image quality, appearance of the disease, differences in the classes, differences in annotation, or other factors.

The difficulty of a test set has been pointed out in research on out of distribution (OOD) evaluation: how different test set examples are from examples in training. In materials science, for instance, some nominally OOD tests could be well covered by the training representations and hence the generalization may be overly great [6]. This will be helpful when considering and interpreting external assessments, but it does not directly confirm the behaviour of plant disease classifiers in their experiments.

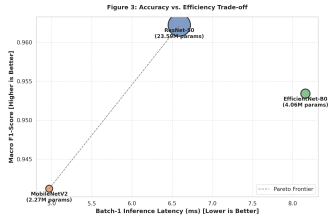


Fig. 3. Predictive performance-latency trade-off for the three CNN architectures.

4.3. Paired Comparison of PlantDoc Predictions

Using the same PlantDoc images, the paired correct and incorrect predictions of each model pair were used in McNemar's tests. The recorded p-values are given in table IV.

The values of all three p values were higher than the 0.05 level of significance. As a result, there were no statistically significant differences in the paired percentage of correct versus incorrect outcomes of predictions in this subset of the evaluation.Does not show that the models are the same. This only holds for the images used for evaluation, and the actual results of the predictions as recorded in the experiment. Higher number of external data sets and unified assessment of classes would give better evidence to compare the models.

4.4. Computational Efficiency and Model Complexity

The results of computational efficiency are tabulated in Table V. It used a batch size of 1, and an input tensor of dimensions $1 \times 3 \times 224 \times 224$, using 50 warm-up forward passes before measuring the timings. The reported measurements are for the particular implementation environment in the experiment.

The performance of the models evaluated, and their batch-one inference latency are shown in Fig. 3 in a trade-off manner.

With 2.27 million parameters, the smallest of all the models, and at just 4.97 ms per image, the lowest measured inference latency, MobileNetV2 was the smallest and the fastest of all the models. It had a measured throughput of 201.2 images per second. MobileNetV2 had about 90.4% fewer parameters than ResNet-50, a latency of 24.6% lower, and a throughput of 32.6% higher than ResNet-50 when compared to ResNet-50, which had 23.59 million parameters,

TABLE IV

MCNEMAR'S TESTS FOR PAIRED MODEL COMPARISONS

Model comparison	p-value	Result at $\alpha = 0.05$
ResNet-50 vs. MobileNetV2	1.0000	Not statistically significant
MobileNetV2 vs. EfficientNet-B0	0.7353	Not statistically significant
ResNet-50 vs. EfficientNet-B0	0.6069	Not statistically significant

TABLE V
RECORDED COMPUTATIONAL-EFFICIENCY MEASUREMENTS

Model	Parameters (millions)	Latency (ms)	Throughput (images/s)
ResNet-50	23.59	6.59	151.7
MobileNetV2	2.27	4.97	201.2
EfficientNet-B0	4.06	8.16	122.6

a latency of 6.59 ms, and a throughput of 151.7 images per second. The efficiency of EfficientNet-B0 was 4.06M parameters, 8.16ms of latency and 122.6 images per second of throughput. Even though the total number of parameters in the model was higher than that of MobileNetV2, ResNet-50 had the highest total number of parameters among the three architectures.

The above measurements illustrate that size is not the only measure of inference efficiency. The performance of the model architectures can be compared using complementary information for parameter count, latency and throughput [6], [7]. Nevertheless, the hardware model used with the benchmark was not documented and measurements were taken in a CUDA environment. So, the results should be taken as a measure of the tested environment and not as being representative of performance on mobile phones or embedded edge devices.

4.5. Discussion

Three primary observations are made in the experiments: In addition, ResNet-50 model had the best internal validation accuracy of all three models. Second, when addressing the mapped subset of the PlantDoc evaluation set, ResNet-50 was the most accurate with 21.23% accuracy and the most accurate macro-F1 score was obtained by the MobileNetV2 model with 22.48%. The accuracy and macro-F1 score for EfficientNet-B0 were 18.49% and 17.82% respectively. Third, in the test environment, MobileNetV2 had the smallest number of parameters, the smallest measured inference latency and the greatest measured throughput.

The results suggest that the model selection should not be based solely on the accuracy of the model in the internal validation or the number of its parameters. Each measure (accuracy, macro-F1) gives complementary information on classification performance, whereas the number of parameters, inference latency, and throughput offer information on various aspects of computational efficiency. This means that the preferred architecture will vary based on the application and computational requirements. However, the external evaluation only covered 146 images in 17 classes and computational measurements were taken in a specific hardware setup. The conclusions should, therefore, be only as valid as the datasets and experimental setup for this study.

The previous work has demonstrated that it is possible to use spurious correlations where the correlation between the class labels and background features are exploited by image classifiers [24]. Izmailov et al. also studied the learning mechanism of task-relevant and spurious features by neural networks [25] and Wang et al. studied the shortcuts related with frequency in image classification and showed that the shortcuts can transfer between datasets [26]. The various studies mentioned above give some useful background for the present cross-dataset study to investigate performance degradation experienced. But the current experiments are not able to confirm if the effects they observed on PlantDoc performance were due to background bias or frequency shortcuts. Greatly more controlled experiments would be needed to find underlying causes.

An alternative stream of research on distribution shift has focused on strategies to make the model more robust when making predictions. Schneider et al. showed that using batch normalisation statistics that are adapted to better handle common image corruptions makes them more robust [27]. However, distribution shifts between different evaluation conditions are still problematic, as motivated by Lim et al. they proposed a method called test-time normalization (TTN) that utilizes source-domain and test-batch statistics [30]. These studies indicate some areas in which further work in classification of plants diseases is needed. However, in this paper the former two methods were not used as one of the three main comparisons and advantages mentioned in those papers cannot be considered as transferable advantages to this study's data sets.

As a whole, the results demonstrate the value of considering these three points in evaluating the performance of predictive models, namely: their behaviour on the test set, their behaviour on new sets, and their computational cost. However, a well performed model on a source validation dataset can not be expected to perform the same on an external and a model's efficiency cannot be solely evaluated based on the number of parameters. Therefore, a multi-criterion evaluation could offer more information to choose CNN architecture for plant disease classification.

4.6. Limitations

There are a number of caveats to these findings. First, only 146 mapped PlantDoc images were used for the external evaluation, instead of the entire PlantDoc images, which corresponds to 17 mapped classes. The findings in this project are not generalisable to other farms/camera systems/other growing conditions/disease types due to the small number of samples and classes used in the project.

Second, the source data-set was split using a seeded random image level split – not a stratified or group based

split. Since the data is already enhanced, duplication and/or related images between training and validation sets should be explored. Their inclusion may have an impact on the validity of the internal validity estimates.

Third, a smaller set of classes (38) is used by the source models, while the external evaluation is performed with a smaller subset of classes (17) that is mapped from the original 38 classes. Predictions made by classes that are not in the mapped target vocabulary need to be dealt with in the same manner. The external metrics should therefore be re-calculated with the same set of saved checkpoints and a consistent class mapping method, with a clearly defined plan for handling predictions that aren't predicted. This is especially important as the experimental notebook provides MobileNetV2 macro-F1 that varies between multiple evaluation steps.

Fourth, it is clear from the current results that the observed performance degradation in cross-datasets cannot be blamed solely on specific items since it is a complex phenomenon. Experiments have excluded some factors, including differences in background, differences in the illumination, image quality, class composition and difference in annotations, but these are still potential explanations. It is necessary to do additional experiments to isolate the effects of each.

Finally, the computational-efficiency measurements were made in a CUDA environment, which does not record the exact model of the GPU, nor is it clear whether the timing was repeated using different instances of the same model. The measured latency and throughput therefore reflects the tested environment and not (directly) applies to mobile phones or embedded edge devices. Going forward, model evaluations should record the hardware used, repeat the benchmarking, and leverage other, independent field datasets to inform a robust cross-dataset stability and model selection for deployment use.

V. CONCLUSIONS

In the present study, the classification of plant diseases was studied and compared between three models, namely ResNet-50, MobileNetV2 and EfficientNet-B0, with a common protocol of transfer-learning the frozen features. The ResNet-50 model had the highest accuracy and macro F1 of 96.33% and 96.22%, respectively, against the internal validation data. But the accuracy obtained from the mapped PlantDoc evaluation was significantly lower for all three models. Overall, ResNet-50 performed best on the external accuracy of 21.23%, while MobileNetV2 was the best on the external macro-F1 score of 22.48%. MobileNetV2 also had the best measured computational profile with 2.27 million parameters, 4.97 ms of computation time to run an inference and 201.2 images per second. The results also indicate that model rankings are a function of the evaluation metric as well as internal validation on only one dataset does not provide an adequate characterization of performance on another.

There are a few limitations that should be taken into account when interpreting the findings. Because only 146 images were used for external evaluation across 17 classes mapped and with different sets of classes for the source and target macro-F1 scores, the numerical difference between the scores is not necessarily a measure of performance degradation. The predictions of the models did not differ significantly from each other in terms of prediction correctness (too few significant differences were found in the McNemar tests to conclude this). Furthermore, the computed measurements are given for a particular CUDA environment, from which the GPU model also was not identified. Future work should consider the use of other datasets acquired independently, revisit the source validation metrics with the same mapped classes, explore the potential for duplicate or near-duplicate images within the training and test sets, and perform hardware and model-specific memory benchmarking and analysis. These steps would establish better evidence to choose practical models for classification of plant diseases.

DATA AND CODE AVAILABILITY

The PyTorch training scripts for the evaluated models (ResNet-50, MobileNetV2, EfficientNet-B0) and the cross-dataset mapping dictionaries used in this study are publicly available on GitHub at: `https://github.com/nitink-22/AgriVision_Research_Project`. The New Plant Diseases Dataset and PlantDoc dataset are available in their respective public repositories as cited.

REFERENCES

[1] D. P. Hughes and M. Salathé, "An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing," *arXiv:1511.08060*, 2015.

[2] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, vol. 7, Art. no. 1419, 2016, doi: 10.3389/fpls.2016.01419.

[3] K. P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, vol. 145, pp. 311–318, 2018, doi: 10.1016/j.compag.2018.01.009.

[4] J. G. A. Barbedo, "A review on the main challenges in automatic plant disease identification based on visible range images," *Biosystems Engineering*, vol. 144, pp. 52–60, 2016, doi: 10.1016/j.biosystemseng.2016.01.017.

[5] D. Singh, N. Jain, P. Jain, P. Kayal, S. Kumawat, and N. Batra, "PlantDoc: A dataset for visual plant disease detection," in *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, 2020, pp. 249–253, doi: 10.1145/3371158.3371196.

[6] M. M. H. Shuvo, S. K. Islam, J. Cheng, and B. I. Morshed, "Efficient acceleration of deep learning inference on resource-constrained edge devices: A review," *Proceedings of the IEEE*, vol. 111, no. 1, pp. 42–91, 2023, doi: 10.1109/JPROC.2022.3226481.

[7] H. Benmeziane, K. El Maghraoui, H. Ouarnoughi, S. Niar, M. Wistuba, and N. Wang, "Hardware-aware neural architecture search: Survey and taxonomy," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021, pp. 4322–4329, doi: 10.24963/ijcai.2021/592.

[8] E. Moupojou, A. Tagne, F. Retraint, A. Tadonkemwa, D. Wilfried, H. Tapamo, and M. Nkenlifack, "FieldPlant: A dataset of field plant images for plant disease detection and classification with deep learning," *IEEE Access*, vol. 11, pp. 35398–35410, 2023, doi: 10.1109/ACCESS.2023.3263042.

[9] A. Ahmad, A. El Gamal, and D. Saraswat, "Toward generalization of deep learning-based plant disease identification under controlled and field conditions," *IEEE Access*, vol. 11, pp. 9042–9057, 2023, doi: 10.1109/ACCESS.2023.3240100.

[10] X. Wu, X. Fan, P. Luo, S. D. Choudhury, T. Tjahjadi, and C. Hu, "From laboratory to field: Unsupervised domain adaptation for plant disease recognition in the wild," *Plant Phenomics*, vol. 5, Art. no. 0038, 2023, doi: 10.34133/plantphenomics.0038.

[11] S. Yang, Q. Feng, J. Zhang, W. Yang, W. Zhou, and W. Yan, "From laboratory to field: Cross-domain few-shot learning for crop disease identification in the field," *Frontiers in Plant Science*, vol. 15, Art. no. 1434222, 2024, doi: 10.3389/fpls.2024.1434222.

[12] M. T. Mallick, S. Banerjee, N. Thakur, H. N. Saha, and A. Chakrabarti, "Evaluation of state-of-the-art deep learning techniques for plant disease and pest detection," *Computers, Materials & Continua*, vol. 85, no. 1, pp. 121–180, 2025, doi: 10.32604/cmc.2025.065250.

[13] A. Upadhyay, N. S. Chandel, K. P. Singh, S. K. Chakraborty, B. M. Nandede, M. Kumar, A. Subeesh, K. Upendar, A. Salem, and A. Elbeltagi, "Deep learning and computer vision in plant disease detection: A comprehensive review of techniques, models, and trends in precision agriculture," *Artificial Intelligence Review*, vol. 58, Art. no. 92, 2025, doi: 10.1007/s10462-024-11100-x.

[14] R. A. Charisma, R. Halim, A. Maulina, K. A. Minor, and A. Nathen, "Paddy leaf image classification using EfficientNetB5 with realistic augmentation," in *Proceedings of the 8th International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, 2025, pp. 152–157, doi: 10.1109/ISRITI68345.2025.11393422.

[15] P. Kaushik and S. Khurana, "Jute leaf disease detection using ResNet50: A deep learning approach for precision agriculture," in *Proceedings of the International Conference on Electronics and Renewable Systems (ICEARS)*, 2025, pp. 1901–1906, doi: 10.1109/ICEARS64219.2025.10940960.

[16] H. Joshi, "Edge-AI for agriculture: Lightweight vision models for disease detection in resource-limited settings," *arXiv preprint arXiv:2412.18635*, 2024.

[17] A. Junaidi, S. Z. Mohd Hashim, M. S. B. Othman, M. M. B. Mohamad, H. Alhussian, S. J. Abdulkadir, M. Nasser, and Y. A. Bena, "Deep learning and edge computing in agriculture: A comprehensive review of recent trends and innovations," *IEEE Access*, vol. 13, pp. 137464–137490, 2025, doi: 10.1109/ACCESS.2025.3595621.

[18] A. V. Karkikatti, R. H. Goudar, V. N. Rathod, A. Kulkarni, and S. Sangani, "A systematic review of artificial intelligence and internet of things applications in precision agriculture," *Discover Artificial Intelligence*, vol. 6, Art. no. 867, 2026, doi: 10.1007/s44163-026-01955-0.

[19] P. Mahapatra, M. Panda, S. K. Dash, and U. K. Sahu, "Advancing plant disease classification using an attention-based CNN for intra-dataset and cross-dataset training," *Scientific Reports*, vol. 16, Art. no. 10925, 2026, doi: 10.1038/s41598-026-45464-7.

[20] A. Davila, J. Colan, and Y. Hasegawa, "Comparison of fine-tuning strategies for transfer learning in medical image classification," *Image and Vision Computing*, vol. 146, Art. no. 105012, 2024, doi: 10.1016/j.imavis.2024.105012.

[21] A. Lynch, G.-J.-S. Dovonon, J. Kaddour, and R. Silva, "Spawrious: A benchmark for fine control of spurious correlation biases," *arXiv:2303.05470*, 2023.

[22] P. Izmailov, P. Kirichenko, N. Gruver, and A. G. Wilson, "On feature learning in the presence of spurious correlations," in *Advances in Neural Information Processing Systems*, 2022.

[23] S. Wang, R. Veldhuis, C. Brune, and N. Strisciuglio, "What do neural networks learn in image classification? A frequency shortcut perspective," *arXiv:2307.09829*, 2023.

[24] S. Schneider, E. Rusak, L. Eck, O. Bringmann, W. Brendel, and M. Bethge, "Improving robustness against common corruptions by covariate shift adaptation," in *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[25] H. Benmeziane, K. El Maghraoui, H. Ouarnoughi, S. Niar, M. Wistuba, and N. Wang, "A comprehensive survey on hardware-aware neural architecture search," arXiv:2101.09336, 2021.

[26] H. Benmeziane, K. El Maghraoui, H. Ouarnoughi, S. Niar, M. Wistuba, and N. Wang, "Hardware-aware neural architecture search: Survey and taxonomy," in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, 2021.

[27] H. Lim, B. Kim, J. Choo, and S. Choi, "TTN: A domain-shift aware batch normalization in test-time adaptation," in *Proceedings of the Eleventh International Conference on Learning Representations (ICLR)*, 2023.