IAES International Journal of Artificial Intelligence (IJ-AI)
Vol. 15, No. 6, December 2026, pp. 5176∼5189
ISSN: 2252-8938, DOI: 10.11591/ijai.v15.i6.pp5176-5189

Hybrid edge-cloud framework with generative models for automated phytopathology

Aman Bhardwaj, Jeet Bhardwaj, Sumit Dhariwal
Center for Artificial Intelligence, Madhav Institute of Technology and Science, Gwalior, India

Article Info

Article history:

Received Aug 16, 2025
Revised Aug 27, 2026
Accepted Sep 20, 2026

Keywords:

Cloud computing
Convolutional neural networks
Edge computing
Generative artificial intelligence
Large language models
Plant disease diagnosis

ABSTRACT

Diagnosing plant diseases accurately and on time is a major factor that keeps the agricultural sector productive and the world fed. Visual inspection by the traditional methods has low scalability and consistency as their main drawbacks. To address these, an edge-cloud diagnostic hybrid architecture is proposed in this paper, which includes the real-time inference on the device coupled with the high-precision analysis and treatment guidance in the cloud. To achieve this, a mobile network version 2 (MobileNetV2) classifier with slight computational overhead runs the classification very fast at the edge, while an efficient network variant B3 (EfficientNetB3) model that stands for the highest accuracy in the cloud environment. Both architectures are supported by the custom-designed classifier heads where the EfficientNetB3 uses batch normalization and dropout for regularization in the training phase. A proof-of-concept implementation integrates a large language model (LLM) for the generation of context-aware automatic treatment recommendations based on the classification output. Furthermore, along with the release of source codes and trained models for reproducibility, this study establishes a quantitative benchmark for the edge-cloud trade-off between computational efficiency and diagnostic fidelity. This architectural scheme is a platform that can be scaled up and deployed for precision agriculture.

This is an open access article under the CC BY-SA license.

Corresponding Author:

Sumit Dhariwal
Centre for Artificial Intelligence, Madhav Institute of Technology and Science
Gwalior, India
Email: sumitdhariwal22@gmail.com

1. INTRODUCTION

Diseases in plants can have such a massive impact on world agriculture that they can cause economic instability, create a situation where food security is at risk, and even wipe out crop varieties, which amount to big losses in yield [1], [2]. The traditional way of diagnosing diseases relies to a large extent on human inspection by experts. Therefore the diagnosis is rather slow and very subjective, and it cannot be scaled up to match the needs of modern agriculture. Farmers in remote areas suffer more under these conditions due to lack of experts, and what is worse is that this lack leads to the wrong treatment or the right one but very late, considering that there are some diseases that have very similar symptoms and can thus be confused. In addition, this condition hampers the application of precision farming, which is the utilization of resources at the best possible rates, providing inputs only in the required locations and times [3], [4]. Deep learning which is a key factor in the fast growth of artificial intelligence (AI) is creating very much reliable and robust solutions for image classification issues and consequently, another stage for automated

plant disease diagnosis is being opened [5], [6].

This study introduces an hybrid edge-cloud framework that is mainly designed to be fast and accurate in terms of the classification accuracy. The proposed method involves the use of two distinct convolutional neural networks (CNN) models. First, there is a light mobile network version 2 (MobileNetV2) [7] model designed for fast, on-device processing. At the same time, efficient network variant B3 (EfficientNetB3) [8] targets cloud deployment to infer with higher accuracy. Therefore, the field user might get quick diagnosis, while complicated or critical ones are automatically sent up for detailed evaluation. Also, within this study is a discussion of how an integration of large language model (LLM) using an application programming interface (API) by Google [9] can be utilized to convert the complicated diagnostics into treatment and action advice. To illustrate the whole process from image capture and automatic classification to intelligent and context-aware agricultural advisory, a proof of concept, Flask based interface [10] has been developed. Rather than introducing novel machine learning architectures, this study aims to provide a practical, system-level integration of these technologies. Ultimately, it aims to develop accurate, scalable, and accessible diagnostic systems for resource-poor agricultural settings by providing reusable design blueprints.

The primary contributions of this study are outlined as follows:

i) The proposed research work designs and assesses a hybrid edge-cloud system where MobileNetV2 will be used for on-device inference and EfficientNetB3 for the detection and classification of plant diseases in the cloud. This enables direct comparison of the performance characteristics of the two systems.
ii) This study quantifies the accuracy, precision, recall, and F1-score trade-offs between the edge-optimized model and the cloud-deployed model, respectively, using a publicly available PlantVillage dataset.
iii) Integration of generative AI on an experimental basis to show how the machine learning classification output can lead to stimulus-aware prevention-treatment recommendations and thus highlight a possible route towards intelligent agricultural advisory systems.
iv) A proof-of-concept that demonstrates the practicality of the idea, consisting of a web-application, trained models, and source code made available to promote reproducible research and support further research.

The paper is organized in this manner. Section 2 provides a concise overview of prior research; Section 3 describes the dataset, model architectures, and experimental setup. In section 4, the empirical results are presented and discussed, and directions for future research are proposed. Lastly, section 5 concludes the paper.

2. RELATED WORK

The plant disease detection methodology has undergone a substantial change. This shift moved from human observatory to computer vision automation. In this part, the movement from traditional machine learning to cutting-edge combined deep learning and generative AI techniques is evaluated.

2.1. Evolution from traditional learning to deep learning

Phytopathology algorithms based on classical approaches include support vector machine and random forest. While they were effective for small-sized data sets, they failed due to field variations [11]. Even though CNN-based algorithms could perform feature extraction automatically, very deep networks like visual geometry group 19 (VGG19) and residual network 50 layers (ResNet50) lacked sufficient computational resources, preventing their real-time applications [12]. That is why current studies mainly concentrate on lightweight neural networks.

These neural networks include MobileNetV2, which is lightweight for mobile devices [13], and EfficientNetB3, which relies on adaptive augmentation [14], thus preserving its high performance under the resource constraints. Numerous lightweight vision models and edge-AI solutions are used in the current studies. MobileNetV2 and EfficientNetB3 have been selected for the proposed solution due to their ecosystem support and hardware acceleration abilities.

2.2. Hybrid edge-cloud architectures

Latency-accuracy problem can be resolved through recent research in hybrid edge-cloud frameworks. In general, it means that light weight model will be used to carry out quick computations at the edge and uncertain samples will be computed using cloud models [15]. This usually leads to saving of computational power without lowering the quality of diagnostics, which is very important for precision agriculture. However, more detailed experimental study of specific models and the process of their implementation is required.

2.3. Generative artificial intelligence in agriculture

While CNN-based models perform exceptionally well in the tasks of image classification, they lack any reasoning capabilities in terms of providing the context and suggestions on the actions to be taken. The recent advent of the new generation of LLMs, such as generative pre-trained transformer (GPT) or Gemini, opens up many exciting avenues in the sphere of what is called "smart" agriculture. The existing solutions rely on the use of generative models to translate the output of classifiers into actionable agronomical recommendations [5]. Table 1 shows the limitations of current approaches and how the proposed method addresses them.

Table 1. Comparison with prior related work and novelty of proposed work

Prior work	Limitations	Novelties in this study
Upadhyay *et al.* [11]	Focused on standard CNN benchmarking; lacked a hybrid deployment strategy.	Proposes and evaluates a dual-model edge-cloud workflow to optimize both latency and accuracy.
Lu *et al.* [13]	Optimized MobileNetV2 for mobile devices but did not integrate cloud-based verification for ambiguous cases.	Utilizes MobileNetV2 as a fast edge filter, backed by EfficientNetB3 as a simulated cloud node for high-precision fallback.
Adnan *et al.* [14]	Utilized EfficientNetB3 for high accuracy but did not address the computational latency for real-time field use.	Quantifies the trade-off between the edge model (speed) and the cloud model (fidelity) in a unified system.
Standard CNNs [11], [12]	Outstanding in visual classification but lack contextual reasoning or the ability to suggest agronomic actions.	Integrates a generative AI (Gemini API) component to provide automated, context-aware treatment recommendations.

3. METHOD

This section provides a thorough description of the data acquisition procedures. It outlines the relevant preprocessing methods and all model architecture specifications. It also describes the training specifications and evaluation criteria.

3.1. Dataset collection and preprocessing

Drawing upon the external PlantVillage dataset [16], a curated subset is used in this research comprising 32,962 images of color leaves. These images belong to 27 classes, which include either the healthy state or various diseases of six types of crops, namely apple, corn, grape, potato, strawberry, and tomato. A diagrammatic representation of the image distribution across the various classes is provided in Figure 1. The figure clearly demonstrates the class imbalance present within the dataset, where the bar colors indicate the respective plant species as detailed in the legend. For effective training and validation of the model, the data set was properly divided into the training set, the validation set, and the testing set. The exact data splitting process is described in detail in Table 2.

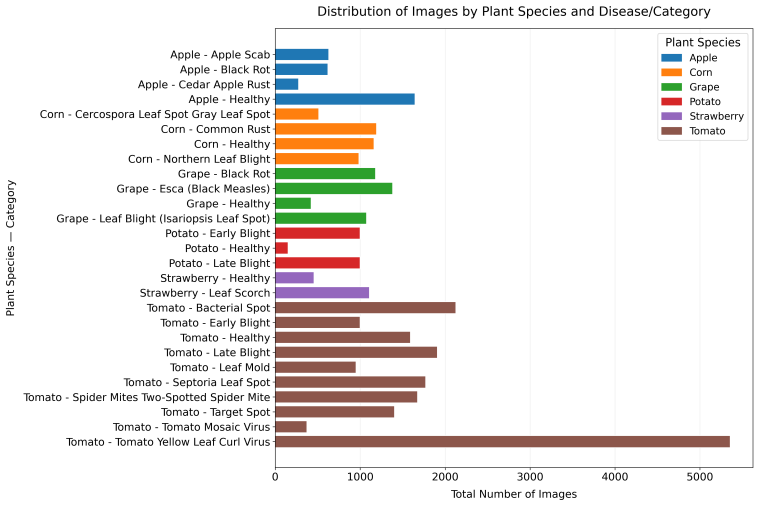


Figure 1. Class distribution of the PlantVillage dataset subset used in this study across 27 classes

Table 2. Data split for training, validation, and testing

Set	Images	Percentage (%)
Training	21,095	64
Validation	5,274	16
Testing	6,593	20
Total	32,962	100

Image preprocessing procedures were tailored to each model's input and optimization techniques:

i) MobileNetV2: images were rescaled to 224×224 pixels and normalized to have intensities from $[0, 1]$.

ii) EfficientNetB3: images were rescaled to 300×300 pixels and normalized using the model's preprocessing method, where specific mean and standard deviation calculated during the pre-training of the model on image network (ImageNet) are applied.

iii) Data augmentation (training set only): in order to avoid overfitting and encourage generalization, data augmentation techniques were applied to the training dataset. It consisted of randomly rotating, shifting images in width and height, applying shear transformation, zooming, and horizontal flipping. Geometric transformations are extremely important in plant pathology, since they ensure invariance to the angle of camera and orientation of leaves in field conditions [17]. MobileNetV2 used stronger augmentation, while EfficientNetB3 used milder augmentation with class weighting.

3.2. Hybrid edge-cloud architecture

The overarching system architecture, visually represented in Figure 2, is engineered as an offline-first solution, which prioritizes offline edge inference. Cloud analysis is triggered based on network availability and user demand for advanced recommendations. The core routing logic demonstrates how ambiguous edge predictions seamlessly trigger cloud validation and LLM advisory handshakes. The primary diagnostic process occurs locally on the edge device to ensure functionality in remote areas with poor internet connectivity. The cloud infrastructure serves as an on-demand extension for high-precision validation and generative AI advisory.

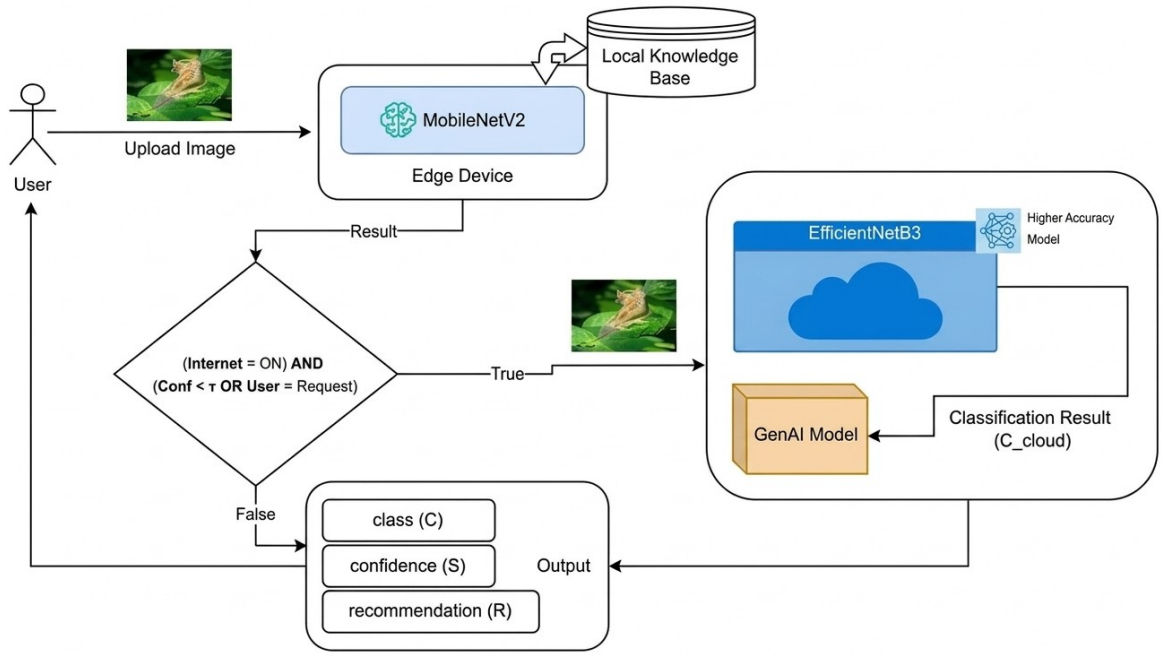


Figure 2. The proposed workflow

3.2.1. Edge node: MobileNetV2 (offline)

MobileNetV2 [7] was chosen as the lightweight architecture designed to perform classification on edge devices such as smartphones. This work used a pre-trained model for the ImageNet backbone that was frozen initially with a custom head, consisting of the global average pooling layer, the dense layer (512 neurons with rectified linear unit (ReLU activation)), the dropout layer (dropout rate 0.5), and the softmax layer

(27 classes). The architecture of the model is shown in Figure 3, observe the lightweight custom head designed to minimize computational overhead on mobile CPUs. Modified from [7].

Importantly, the proposed edge device would be complemented by a lightweight database with the standard agriculture treatment methods, which takes up little memory. Right after the classification, the application receives pre-verified recommendations, corresponding to the detected class. This approach provide an immediate though standard decision-making support to farmers in the no-connectivity zones.

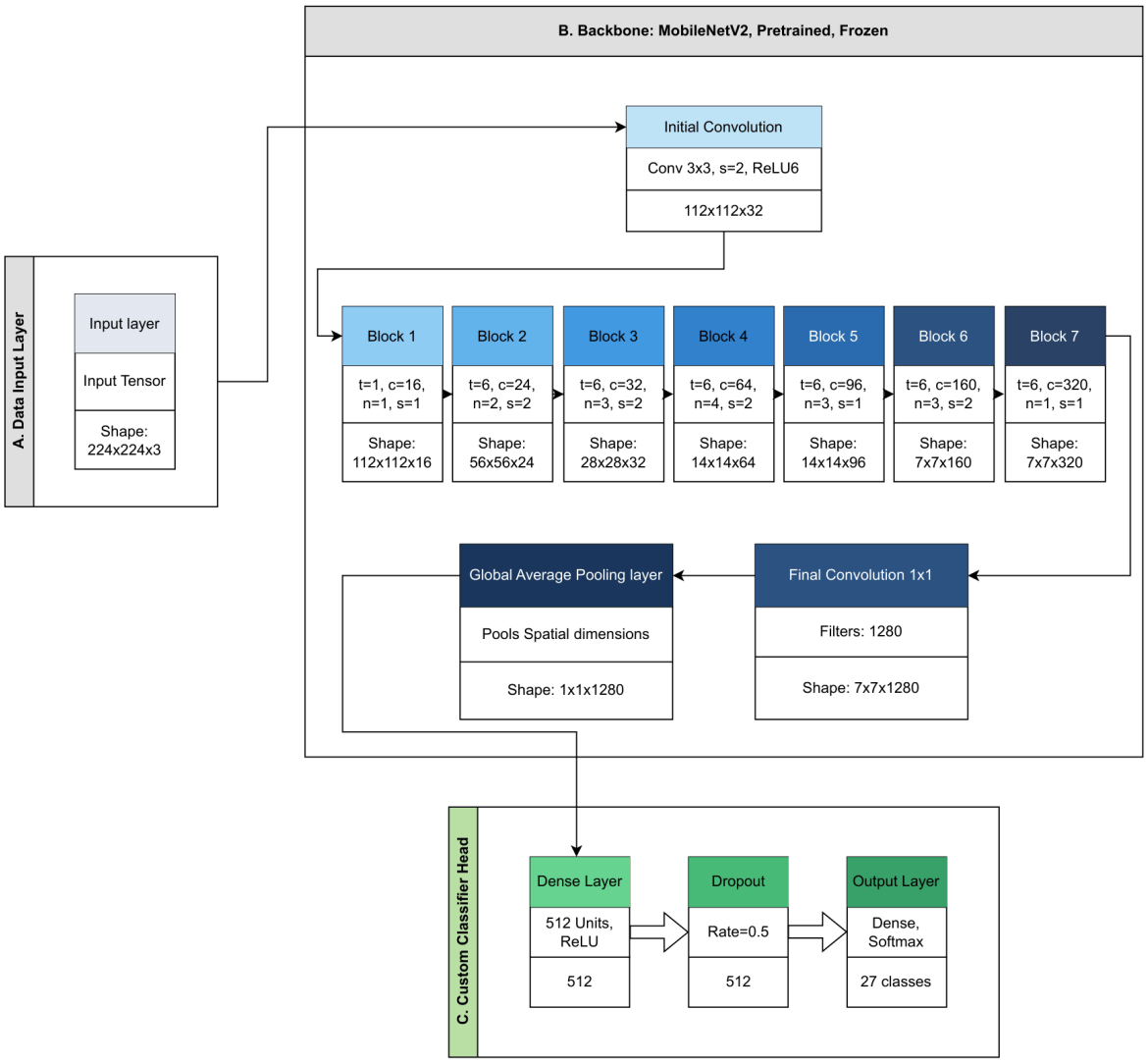


Figure 3. Architecture of MobileNetV2-based model

3.2.2. Cloud node: EfficientNetB3 (on-demand)

The cloud node utilizes the EfficientNetB3 [8] architecture which is tuned for accuracy rather than efficiency. It consists of compound scaled backbone connected to custom head with two dense layers (1,024 and 512 nodes each) normalized by batch normalization and dropout (0.5). The architecture is shown in Figure 4, the expanded dense blocks and regularization layers ensure high-precision validation for complex or ambiguous cases forwarded from the edge, modified from [8].

This node does not activate automatically. It's called up if i) internet is accessible and ii) either the confidence score of the model on the edge drops under a threshold τ or the user asks for a "second opinion". Upon which, EfficientNetB3 process the image and ask the LLM about the best course of action.

3.3. Training and optimization

The experiments have been performed with TensorFlow [18] running on NVIDIA Tesla P100 GPU. The training process included the adaptive moment estimation (Adam optimizer) [19] with the starting learning rate of 1×10^{-4} and the loss function being the categorical cross-entropy. In order to tackle the problem of imbalance in the dataset [20], while training the EfficientNetB3 network, the class weights which have been calculated by the formula $w_c = N/(C \cdot n_c)$ were used, where N stands for the number of training samples, C stands for the number of classes in total (27), and n_c is the number of occurrences of class c. Training schedule, MobileNetV2 model was trained for one complete cycle, spanning 50 epochs with a batch size of 32. Conversely, EfficientNetB3 had a training strategy based on a two-phase transfer learning: i) custom head training with frozen back-bone; despite being scheduled for 50 epochs, training stopped in epoch 25 with early stopping (patience = 5) in order to avoid overfitting [21]; and ii) top 30 layers were unfrozen and fine-tuned for 10 epochs with a lowered learning rate (1×10^{-5}). Learning rate scheduler (ReduceLROnPlateau, factor = 0.1) was active throughout training process [22].

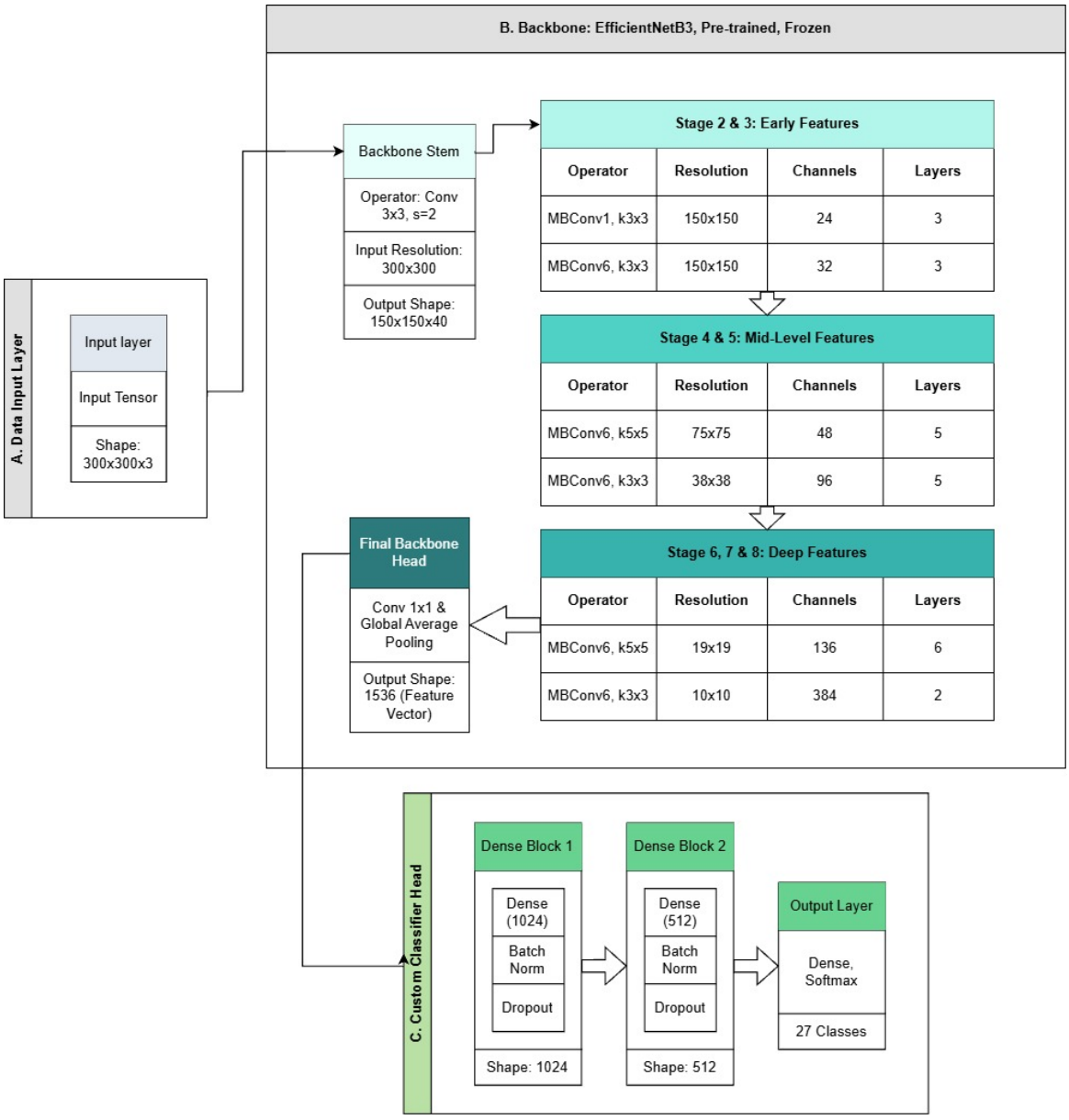


Figure 4. EfficientNetB3-based cloud model architecture

3.4. System workflow

The diagnostic pipeline specified in Algorithm 1 is designed for operational resilience. The default mode is an offline mode that compares the classification by the edge model with a local database of standard treatment techniques that have been previously indexed. The high-latency path of accessing the cloud with EfficientNetB3 and Gemini API usage will be enabled only in certain cases to ensure that only ambiguous cases or when a patient requests the advanced AI-powered analysis will consume bandwidth.

The confidence threshold τ is the main criterion that separates edge and cloud tiers. High τ means that more images will be sent to the cloud, thus increasing the accuracy of the diagnosis but increasing the latency and bandwidth consumption too, whereas low τ means more reliance on the edge model. In this study, τ is used as a constant reference parameter to compare models individually; however, in a real operational environment, this parameter could be constantly adjusted based on network conditions.

Algorithm 1 Offline-first hybrid diagnosis with on-demand cloud support

Input: Leaf image I, threshold τ, connectivity Net, user request Req
Output: Disease class C, confidence score S, recommendation R

// Phase 1: Offline edge inference (always executed)
Preprocess I for MobileNetV2: resize to 224×224
$(C_{\text{edge}}, S_{\text{edge}}) \leftarrow \text{Classify}(M_{\text{edge}}, I)$
$R_{\text{local}} \leftarrow \text{QueryLocalDB}(C_{\text{edge}})$
$C \leftarrow C_{\text{edge}}, S \leftarrow S_{\text{edge}}, R \leftarrow R_{\text{local}}$
// Phase 2: Conditional cloud offloading
if Net is **active and** ($S_{\text{edge}} < \tau$ **or** Req is **true**) **then**
    Upload I to the cloud server
    Preprocess I for EfficientNetB3: resize to 300×300
    $(C_{\text{cloud}}, S_{\text{cloud}}) \leftarrow \text{Classify}(M_{\text{cloud}}, I)$
    $Prompt \leftarrow$ ``Provide agronomic advice for`` $+ C_{\text{cloud}}$
    $R_{\text{gen}} \leftarrow \text{GeminiAPI}(Prompt)$
    $C \leftarrow C_{\text{cloud}}, S \leftarrow S_{\text{cloud}}, R \leftarrow R_{\text{gen}}$
end if
return (C, S, R)

3.5. Evaluation metrics

To perform an evaluation of the diagnostic capability, the independent test set ($N = 6,593$) was employed for assessing the performance of the models. Two main measures used for this analysis were overall accuracy and categorical cross-entropy loss. Since there was imbalance in the dataset, it was necessary to assign higher weights to the results and compute the corresponding weighted averages for precision, recall and F1-score in order to avoid any bias from the majority classes. In addition, the confusion matrices were generated in order to illustrate the misclassification patterns among phenotypically similar disease classes.

3.6. Proof-of-concept implementation

To show the effectiveness of the hybrid architecture, a prototype utilizing Flask [10] was developed. Although the theoretical design aims at a native mobile application for offline operation, this web-based implementation serves to test and check the model inference pipeline as well as the generative AI integration. The prototype user interface as shown in Figure 5 takes leaf images uploaded by users and embodies the decision-making process:

i) Edge inference simulation: users are able to classify images using the lightweight MobileNetV2 model to view the quick, baseline classification that is meant to occur on-device.
ii) Cloud validation testing: there is a unique module that allows for image classification by making use of the powerful model known as EfficientNetB3. In a practical scenario, images whose confidence score falls under the value of τ would be assigned to this category for manual verification.
iii) Recommendation production: once classified, the disease label obtained is inputted into the built-in Gemini API which will generate a formatted prevention and cure method as shown in Figure 6.

Figure 7 shows the test environment data flow, demonstrating the practicality of the "offline-first" logic. This figure highlighting the API handshakes between the inference engine and the generative model.

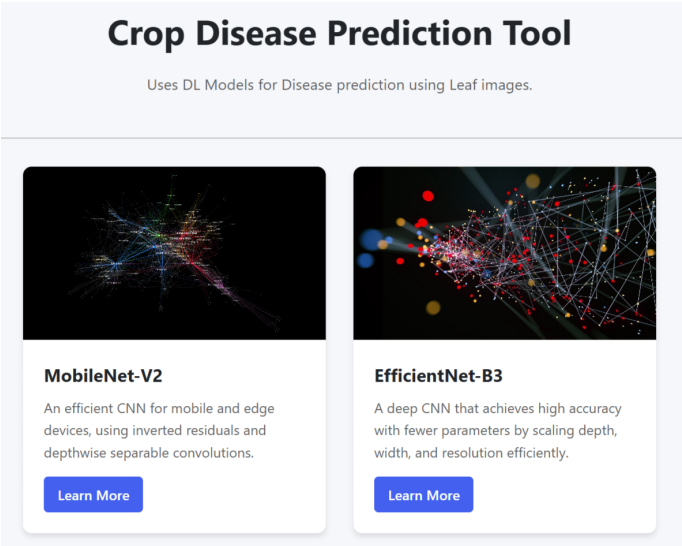


Figure 5. The user interface of the proof-of-concept prototype

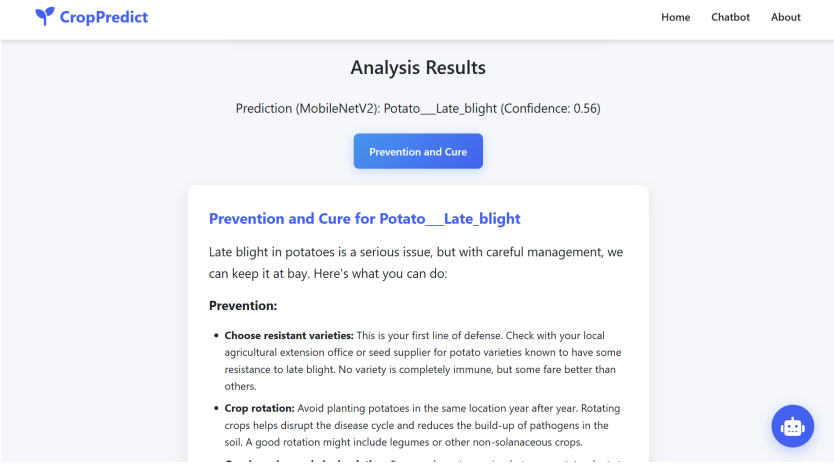


Figure 6. Sample output showing AI-generated treatment recommendations fetched via Gemini API

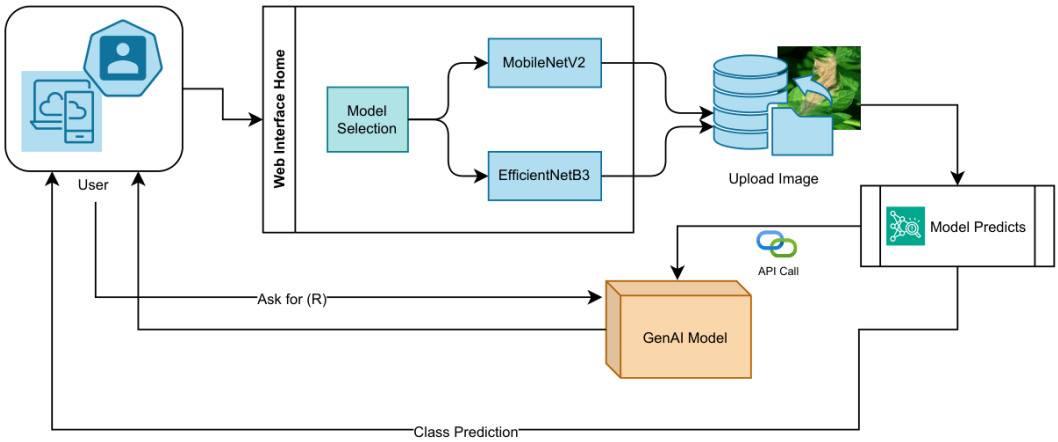


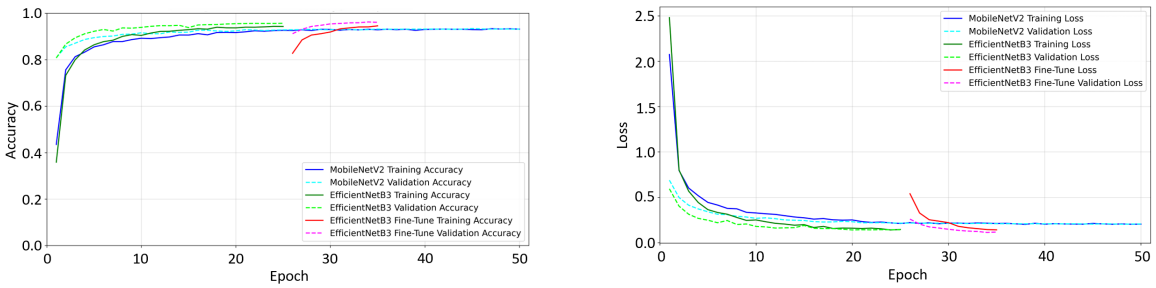
Figure 7. Detailed component interaction diagram for the proof-of-concept implementation

4. RESULTS AND DISCUSSION

The hybrid models' empirical performance is demonstrated in this section, focusing on the significant trade-offs between diagnostic accuracy, computational latency, and architectural complexity within the suggested edge-cloud framework. The analysis examines the behavior of the two models during the training and validation phase, followed by their ability to classify data on the independent test dataset. It also analyze the latency of inference and the effect of the confidence threshold on the distribution of loads on edge and cloud nodes. Finally, the confusion matrix analysis highlights classification challenges involving visually similar plant diseases and provides further insight into the strengths and limitations of the two models.

4.1. Model training and validation performance

Training dynamics, as shown in Figure 8, presents a steady convergence of both models without any issues associated with overfitting. In particular, Figure 8(a) shows the graphs of accuracy curves comparing the edge and cloud models, highlighting the cloud model's higher peak performance, and it can be seen that the maximum validation accuracy of EfficientNetB3 (cloud node) is 96.25%, while for MobileNetV2 is 93.42%. Moreover, Figure 8(b) shows the graphs of loss curves indicating superior generalization and lower error for the EfficientNetB3 model over MobileNetV2, which confirm that both models converge steadily without significant overfitting. The explanation of the different results is in the bigger size of EfficientNet model and in more effective usage of the fine-tuning strategy, when the frozen stage is especially important.



(a) (b)

Figure 8. Training and validation metrics over epochs for both models of (a) accuracy and (b) loss

4.2. Test set performance and latency analysis

For the independent test set ($N = 6,593$), the EfficientNetB3 demonstrated not only consistent results with the highest metric scores but also overall accuracy of 96.56% and weighted F1-score of 0.9659 for all aggregate categories. In its turn, MobileNetV2 also gained decent accuracy of 93.54% (Table 3) and, therefore, can be considered a reliable first-pass filter. The 95% confidence intervals for these final accuracies are $96.56\% \pm 0.44\%$ for EfficientNetB3 and $93.54\% \pm 0.59\%$ for MobileNetV2.

4.2.1. Inference latency evaluation

In order to evaluate the model deployment feasibility on devices with low computational resources, the latency benchmark was conducted using the CPU setting (TensorFlow 2.19.0, single thread) on Kaggle as a simulation of edge devices that do not have AI accelerators. The procedure of the inference time evaluation included a warm-up phase followed by 100 iterations of single image inference. The disparity of a significant magnitude in the computational cost was spotlighted by the results. MobileNetV2 attained an average inference latency of 286.06 ms ($\sigma = 27.05$), which made it appropriate for real-time interaction on mobile CPUs. On the other hand, EfficientNetB3 the average 1,117.41 ms ($\sigma = 75.06$)—taking more than a second per image. This represents a $3.91\times$ speedup in favor of MobileNetV2.

It was also concluded that running models like EfficientNetB3 (or larger model) locally on standard smartphones is not worth it owing to the latency. While the EfficientNetB3 model can take advantage of GPUs and achieve a much lower latency ($\sim$ 68 ms, as on P100 GPU), accessing such infrastructure for every query could be quite costly and it would also require a lot of bandwidth. In particular, the process involves

the transfer of a compressed 300×300 size image (approx. 30 – 50 KB) and results in additional network delay of 100 – 300 ms. Thus, the hybrid architecture that was proposed is now backed by empirical evidence: MobileNetV2 will take care of the major part of the traffic at the edge, whereas GPU power is only to be used when the cases are really ambiguous, thereby making the cost of GPU acceleration and EfficientNetB3 use in the cloud justifiable.

Table 3. Performance metrics of MobileNetV2 and EfficientNetB3 on the test set

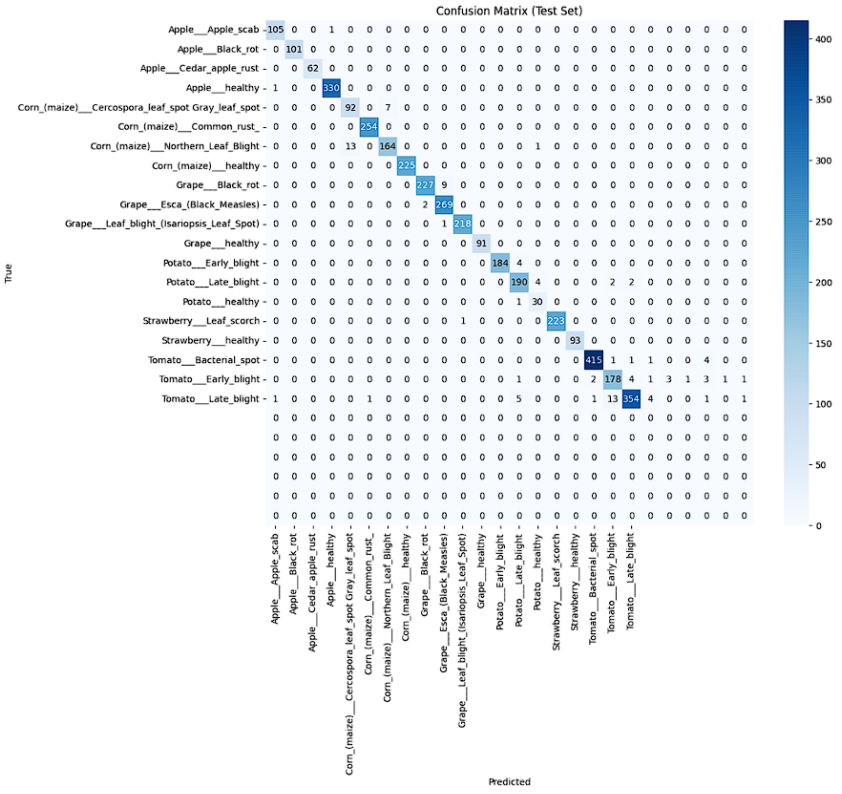
Class	MobileNetV2 (Edge)			EfficientNetB3 (Cloud)		
	Precision	Recall	F1-score	Precision	Recall	F1-score
Apple scab	0.97	0.92	0.95	0.98	0.99	0.99
Apple black rot	0.98	1.00	0.99	1.00	1.00	1.00
Apple cedar apple rust	1.00	0.98	0.99	1.00	1.00	1.00
Apple healthy	0.97	0.99	0.98	1.00	1.00	1.00
Corn common rust	0.97	1.00	0.98	1.00	1.00	1.00
Corn gray leaf spot	0.80	0.79	0.80	0.88	0.93	0.90
Corn healthy	1.00	1.00	1.00	1.00	1.00	1.00
Corn northern leaf blight	0.90	0.89	0.89	0.96	0.92	0.94
Grape black rot	0.97	0.94	0.96	0.99	0.96	0.98
Grape esca black measles	0.95	0.98	0.97	0.96	0.99	0.98
Grape healthy	1.00	1.00	1.00	1.00	1.00	1.00
Grape leaf blight	0.99	0.99	0.99	1.00	1.00	1.00
Potato early blight	0.97	0.94	0.95	0.99	0.98	0.99
Potato healthy	0.84	0.87	0.86	0.83	0.97	0.90
Potato late blight	0.94	0.90	0.92	0.95	0.96	0.95
Strawberry healthy	1.00	0.98	0.99	1.00	1.00	1.00
Strawberry leaf scorch	1.00	0.98	0.99	1.00	1.00	1.00
Tomato bacterial spot	0.91	0.95	0.93	0.95	0.98	0.97
Tomato early blight	0.82	0.75	0.79	0.88	0.91	0.90
Tomato healthy	0.93	0.96	0.94	0.99	0.95	0.97
Tomato late blight	0.90	0.93	0.91	0.97	0.93	0.95
Tomato leaf mold	0.83	0.87	0.85	0.92	0.98	0.95
Tomato septoria leaf spot	0.89	0.87	0.88	0.99	0.91	0.95
Tomato spider mites	0.93	0.86	0.90	0.89	0.97	0.93
Tomato target spot	0.77	0.86	0.81	0.87	0.95	0.91
Tomato mosaic virus	0.93	0.90	0.91	0.93	0.96	0.94
Tomato yellow leaf curl virus	0.99	0.97	0.98	1.00	0.96	0.98

4.2.2. Impact of the confidence threshold (τ)

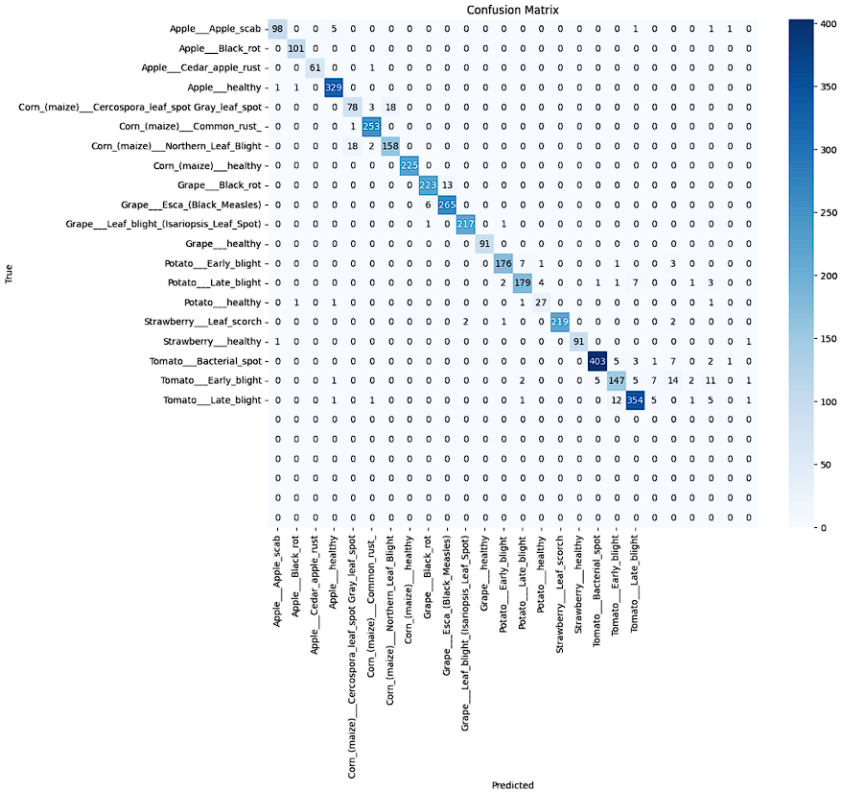
Even though τ was kept fixed in order to capture the base performance level, the dynamic change of such a threshold in actual deployment becomes the direct determinant of the edge-cloud workload distribution. Lower τ leads to less offloading and lower latency with tight bandwidth constraints. On the other hand, larger τ focuses on diagnostic accuracy with more ambiguity handled in the cloud provided that the bandwidth is sufficient. In theory, the sensitivity range of the proposed system extends from the lowest ($\tau \rightarrow 0$, processing at the edge resulting in 93.54% accuracy) to the highest ($\tau \rightarrow 1$, offloading to the cloud resulting in 96.56% accuracy).

4.3. Confusion matrix analysis

The models' confusion matrices shown in Figure 9 illustrate challenges faced by fine-grained classification. In particular, Figure 9(a) presents the confusion matrix of the cloud-based EfficientNetB3 model, demonstrating a pronounced diagonal dominance that underscores its high classification accuracy across the majority of classes. On the other hand, Figure 9(b) illustrates the confusion matrix of MobileNetV2, where the analysis reveals that errors are not randomly distributed but rather clustered around diseases with shared characteristics, thereby indicating higher confusion rates among phenotypically similar classes. For instance, an example of the two models failing to distinguish between tomato_early_blight and tomato_target_spot due to their concentric ring-like patterns However, EfficientNetB3 was quite the contrary in terms of solving the class under-representation challenge. Specifically, it was able to reach F1-score of 0.90 on potato_healthy, while MobileNetV2 was only able to obtain 0.86.



(a)



(b)

Figure 9. Confusion matrices on the test set for the first 20 classes: (a) EfficientNetB3 and (b) MobileNetV2

4.4. Future directions

The present study provides a proof-of-concept, but improvements can be made in the future, including:

i) Expanded field validation: the PlantVillage dataset, which was used for training, provides mainly images in laboratory-like setting. The next step after this work can be taken by recognizing real-world field datasets containing various factors that need to be dealt with, such as different lighting conditions, background noise, and environmental noise, and so on, and determining whether the system can adapt to such datasets, thus, considerably boosting the practicality of the product.

ii) Alignment of recommendations with knowledge and expert review: the generated recommendations have practical applicability to agriculture-related decision-making. However, the precision and validity of the recommendations could be significantly enhanced by applying expert agricultural knowledge. This could be achieved through the training or fine-tuning of LLMs using extremely specialized data sets on agriculture, which include crop management guidelines, soil science information, taxonomy of pests and diseases, and adaptive farming practices [23]. Also, the application of the retrieval-augmented generation (RAG) models in conjunction with extremely reputable and credible sources of agricultural knowledge such as extension literature, peer-reviewed papers on agronomy, and field test results would guarantee the accuracy of the recommendations and prevent model hallucinations instead of logical probability [24]. In order to increase quality assurance, high-quality LLM-as-a-Judge systems could be used for the automatic evaluation of the recommendations in terms of accuracy, validity in agriculture, and compliance with best practices [25]. Human-in-the-loop approaches could also be applied by having certified agronomists review the recommendations on a case-by-case basis [26].

5. CONCLUSION

In this paper, a hybrid approach to phytopathology that integrates edge and cloud technologies has been introduced and verified, addressing the challenging problem of balancing between accessibility and diagnostic accuracy in smart agriculture. In particular, the usage of lightweight MobileNetV2 for efficient edge inference along with a powerful EfficientNetB3 for cloud-based analysis results in a useful solution. The edge-based inference is capable of reducing the latency during mobile operation, while the cloud-based solution takes care of the problem of ambiguous results and ensures high accuracy of the diagnosis. Finally, the use of generative AI helps to bridge the gap between classification of images and practical recommendations in agriculture.

FUNDING INFORMATION

Authors state no funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Aman Bhardwaj		✓	✓			✓	✓		✓	✓	✓			
Jeet Bhardwaj		✓		✓	✓		✓	✓	✓	✓		✓		
Sumit Dhariwal	✓								✓	✓		✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal Analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject Administration
Fu : **Fu**nding Acquisition

CONFLICT OF INTEREST STATEMENT

The authors declare no conflicts of interest related to this work.

INFORMED CONSENT

This section is not applicable. This study did not involve human participants, human data, or any personally identifiable information. All data used were either publicly available, fully anonymized, or derived from non-human sources, and therefore no informed consent was required from individuals.

ETHICAL APPROVAL

This section is not applicable. This research did not involve human subjects, human biological materials, or experimental procedures on animals. The work was conducted solely on computational models, publicly available datasets, or non-sensitive data that did not require intervention with living organisms. Therefore, ethical approval from an institutional review board or animal ethics committee was not necessary for this study.

DATA AVAILABILITY

The source code, trained model weights, and experimental logs are available in GitHub at https://github.com/GitHub-AmanBhardwaj/Crop-Disease-detection. The image dataset employed is a part of the openly accessible PlantVillage dataset.

REFERENCES

[1] N. Wang *et al.*, "Key challenges in plant pathology in the next decade," *Phytopathology®*, vol. 114, no. 5, pp. 837–842, May 2024, doi: 10.1094/PHYTO-04-24-0137-KC.

[2] S. Wang *et al.*, "Advances in deep learning applications for plant disease and pest detection: a review," *Remote Sensing*, vol. 17, no. 4, Feb. 2025, doi: 10.3390/rs17040698.

[3] V. S. Dhaka, N. Kundu, G. Rani, E. Zumpano, and E. Vocaturo, "Role of internet of things and deep learning techniques in plant disease detection and classification: a focused review," *Sensors*, vol. 23, no. 18, Sep. 2023, doi: 10.3390/s23187877.

[4] V. Riyanto, S. Nurdiati, M. Marimin, M. Syukur, and S. N. Neyman, "Deep learning approaches for plant disease diagnosis systems: a review and future research agendas," *Journal of Applied Agricultural Science and Technology*, vol. 9, no. 2, pp. 185–204, May 2025, doi: 10.55043/jaast.v9i2.308.

[5] C. Krupitzer, "Generative artificial intelligence in the agri-food value chain - overview, potential, and research challenges," *Frontiers in Food Science and Technology*, vol. 4, Sep. 2024, doi: 10.3389/frfst.2024.1473357.

[6] N. Singh *et al.*, "Farmer.Chat: scaling AI-powered agricultural services for smallholder farmers," Oct. 2024, *arXiv: 2409.08916*.

[7] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: inverted residuals and linear bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 4510–4520, doi: 10.1109/CVPR.2018.00474.

[8] M. Tan and Q. Le, "EfficientNet: rethinking model scaling for convolutional neural networks," in *36th International Conference on Machine Learning-PMLR 97*, 2019, pp. 6105–6114.

[9] S. Sai, S. Kumar, A. Gaur, S. Goyal, V. Chamola, and A. Hussain, "Unleashing the power of generative AI in Agriculture 4.0 for smart and sustainable farming," *Cognitive Computation*, vol. 17, no. 1, Feb. 2025, doi: 10.1007/s12559-025-10420-6.

[10] M. Grinberg, *Flask web development*. Sebastopol, United States: O'Reilly Media, Inc., 2018.

[11] A. Upadhyay *et al.*, "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, no. 3, Jan. 2025, doi: 10.1007/s10462-024-11100-x.

[12] A. H. Ali, A. Youssef, M. Abdelal, and M. A. Raja, "An ensemble of deep learning architectures for accurate plant disease classification," *Ecological Informatics*, vol. 81, Jul. 2024, doi: 10.1016/j.ecoinf.2024.102618.

[13] J. Lu, X. Liu, X. Ma, J. Tong, and J. Peng, "Improved MobileNetV2 crop disease identification model for intelligent agriculture," *PeerJ Computer Science*, vol. 9, Sep. 2023, doi: 10.7717/peerj-cs.1595.

[14] F. Adnan, M. J. Awan, A. Mahmoud, H. Nobanee, A. Yasin, and A. M. Zain, "EfficientNetB3-adaptive augmented deep learning (AADL) for multi-class plant disease classification," *IEEE Access*, vol. 11, pp. 85426–85440, 2023, doi: 10.1109/ACCESS.2023.3303131.

[15] A. U. Rehman, Y. Alamoudi, H. M. Khalid, A. Morchid, S. M. Muyeen, and A. Y. Abdelaziz, "Smart agriculture technology: an integrated framework of renewable energy resources, IoT-based energy management, and precision robotics," *Cleaner Energy Systems*, vol. 9, Dec. 2024, doi: 10.1016/j.cles.2024.100132.

[16] D. P. Hughes and M. Salathe, "An open access repository of images on plant health to enable the development of mobile disease diagnostics," Apr. 2016, *arXiv: 1511.08060*.

[17] K. Maharana, S. Mondal, and B. Nemade, "A review: data pre-processing and data augmentation techniques," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/j.gltp.2022.04.020.

[18] Y. Özçevik and F. Sönmez, "An embedded TensorFlow lite model for classification of chip images with respect to chip morphology depending on varying feed," *Journal of Intelligent Manufacturing*, vol. 36, no. 3, pp. 1623–1635, Mar. 2025, doi: 10.1007/s10845-023-02320-z.

[19] X. Liu, H. Qi, S. Jia, Y. Guo, and Y. Liu, "Recent advances in optimization methods for machine learning: a systematic review," *Mathematics*, vol. 13, no. 13, Jul. 2025, doi: 10.3390/math13132210.

[20] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, pp. 9260–9269, doi: 10.1109/CVPR.2019.00949.

[21] L. Prechelt, "Early stopping - but when?," in *Neural Networks: Tricks of the Trade*, Berlin, Germany: Springer, 1998, pp. 55–69, doi: 10.1007/3-540-49430-8_3.
[22] P. Johri *et al.*, "Advanced deep transfer learning techniques for efficient detection of cotton plant diseases," *Frontiers in Plant Science*, vol. 15, Dec. 2024, doi: 10.3389/fpls.2024.1441117.
[23] S. Wolfert, L. Ge, C. Verdouw, and M.-J. Bogaardt, "Big data in smart farming – a review," *Agricultural Systems*, vol. 153, pp. 69–80, May 2017, doi: 10.1016/j.agsy.2017.01.023.
[24] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *NIPS'20: Proceedings of the 34th International Conference on Neural Information Processing Systems*, 2020, pp. 9459–9474.
[25] L. Zheng *et al.*, "Judging LLM-as-a-Judge with MT-Bench and chatbot arena," in *Advances in Neural Information Processing Systems 36*, 2023, pp. 46595–46623, doi: 10.52202/075280-2020.
[26] A. Holzinger, "Interactive machine learning for health informatics: when do we need the human-in-the-loop?," *Brain Informatics*, vol. 3, no. 2, pp. 119–131, Jun. 2016, doi: 10.1007/s40708-016-0042-6.

BIOGRAPHIES OF AUTHORS



Aman Bhardwaj is a student at the Center for Artificial Intelligence, Madhav Institute of Technology and Science (MITS-DU), Gwalior, India, pursuing his studies in Artificial Intelligence and Data Science. He works in the fields of deep learning, computer vision, natural language processing, large language models, and retrieval-augmented generation. His main areas of interest are developing and testing intelligent systems using modern artificial intelligence tools and applying them to practical knowledge. His work is aimed at making AI technologies more accurate and robust while building a bridge between methodology and practical implementation He can be contacted at email: whyamanbhardwaj@gmail.com.



Jeet Bhardwaj is a student at the Center for Artificial Intelligence, Madhav Institute of Technology and Science (MITS-DU), Gwalior, India. His research interests are in machine learning, deep learning, and the creation of reliable and robust artificial intelligence systems. His research concentrates on the creation and evaluation of AI models, in which he tries to improve their correctness, reliability, robustness, and efficiency. He is especially interested in creating rigorous methodologies for developing such systems. He can be contacted at email: thejeetbhardwaj@gmail.com.



Sumit Dhariwal is a faculty member of the Center for Artificial Intelligence, Madhav Institute of Technology and Science (MITS-DU), Gwalior, India. He completed his Ph.D. from the Malaysia University of Science and Technology in association with the Singapore Research Center. The areas of research include artificial intelligence, machine learning, image processing, computer vision, crop disease identification, cybersecurity, and internet of things (IoT). He has published over 30 research papers in international journals and conferences where his research concentrates on practical applications of AI. He can be contacted at email: sumitdhariwal22@gmail.com.