JOURNAL OF AGRICULTURAL DIGITALIZATION RESEARCH

LeafFed: Collaborative Plant Disease Intelligence Without Raw Data Sharing Based on Federated Learning

Ritesh Janga [1*], Rushit Dave [2], Mansi Bhavsar [3]
[1-3] Department of Computer Information Science, Minnesota State University, Mankato, MN 56001, USA

* Corresponding Author: **Ritesh Janga**

Article Info

P-ISSN: 3051-3421
E-ISSN: 3051-343X
Volume: 07
Issue: 02
Received: 23-04-2026
Accepted: 25-05-2026
Published: 27-06-2026
Page No: 09-18

Abstract
Background: Plant diseases cause an estimated 20–40 percent of annual crop yield losses worldwide, and most deep learning-based diagnostic systems depend on centralized data pipelines that require farmers to transmit raw crop images to remote servers, raising privacy, connectivity, and data-sovereignty concerns.
Objectives: This study aimed to develop and evaluate LeafFed, a federated learning framework for privacy-preserving, multi-node plant disease classification that avoids transferring raw images between clients.
Methods: LeafFed employs a RegularizedEfficientViT model (a frozen EfficientNet-B0 backbone with a regularized six-layer classifier head) trained on the 38-class PlantVillage color dataset, partitioned across three independent client nodes and optimized using the FedAvg algorithm over five communication rounds.
Key Findings: The resulting global model achieved 92.83 percent test accuracy, 92.50 percent precision, 92.60 percent recall, and a 92.55 percent F1-score, a 5.49 percentage-point improvement over a centralized baseline (88.00 percent accuracy). Test loss decreased monotonically from 2.79 to 0.94 across rounds, and per-client analysis revealed substantial cross-node knowledge transfer, with the weakest client rising from 37.24 percent to 85.31 percent accuracy. These results show that federated learning can match or exceed centralized performance while preserving farm data privacy and sovereignty.

DOI: https://doi.org/10.54660/JADR.2026.7.2.09-18

Keywords: crop disease detection, EfficientNet, federated learning, PlantVillage, precision agriculture

Introduction

Plant pathogens cause an estimated annual loss of US crop value equaling $220 billion [1]. Most of this loss occurs in low-income nations, where small-scale producers depend on local agronomists for plant disease diagnosis; however, due to resource constraints and geographical limitations, diagnostic procedures can take lengthy amounts of time and may be cost-prohibitive. The recent development of deep learning systems in automated plant pathology has provided new avenues for improving the efficiency of disease detection. Convolutional neural networks (CNNs) trained with images from PlantVillage have been reported to achieve accuracy rates of up to 99.2 percent in identifying plant diseases [3], and subsequent studies employing mobile architectures such as EfficientNet [4], MobileNet [5], and Swin Transformer [6] have enabled researchers to develop similar technologies for deployment on agricultural equipment.

A primary limitation of previous research employing CNNs is the reliance on a centralized training framework, which requires farmers to transmit raw images of their crops to remote servers. Remote transmission of raw crop images can violate the provisions outlined by the GDPR [7], may expose commercially sensitive crop intelligence to unauthorized third parties, and, most importantly, excludes participation by farmers in rural regions lacking adequate connectivity. To mitigate these deficiencies, McMahan *et al*. [8] developed Federated Learning (FL), a paradigm that enables multiple clients to jointly train the parameters of a single model while preventing the transfer of raw data between them. Rather than transmitting raw data to a central server, clients transmit only the updates that represent modifications made to their local copy of the model; a central server then combines these updates through a process known as FedAvg. Because none of the raw data leaves individual client machines, FL provides a means for secure, cooperative analysis of multiple distributed datasets. FL has advanced across domains including medical imaging [9], IoT security [10], and financial applications [11]; however, its application to visual plant-disease monitoring systems remains limited. Federated learning in agriculture is still an emerging area: Sarma *et al.* [24] applied FL alongside distributed IoT sensors for precision agriculture; Mamba Kabala *et al.* [25] found FL to be accurate and beneficial for image-based crop disease detection; and Aggarwal *et al.* [26] investigated rice leaf disease classification using Federated Transfer Learning across multiple client silos. Collectively, these studies indicate that FL is applicable to agriculture, but they have explored relatively few architectural variations, leaving a gap for systematic evaluation of federated performance using modern, compact backbone architectures such as EfficientNet.

This manuscript presents LeafFed, a federated learning system that utilizes a RegularizedEfficientViT architecture with a regularized classifier head, trained on the PlantVillage dataset partitioned among three client nodes over five communication rounds. The main contributions of this work are: (i) LeafFed, an end-to-end federated learning pipeline using transfer learning, dual dropout regularization, batch normalization, and label smoothing; (ii) a five-round federated experiment that achieves a diagnostic accuracy of 92.83 percent without exchanging any raw data between clients; (iii) a quantitative comparison showing a +5.49 percentage-point improvement over a centralized baseline; and (iv) an analysis demonstrating cross-node knowledge transfer during convergence. The remainder of this paper is organized as follows: Materials and Methods describes the dataset, preprocessing pipeline, model architecture, system architecture, and training protocol; Results presents the experimental findings; Discussion interprets these findings, situates them within the wider literature, and outlines limitations and future directions; and Conclusion summarizes the study.

Materials and Methods

Dataset

This study uses the PlantVillage color dataset [12], which is publicly available and widely regarded as the largest open-source collection of plant leaf images for diagnosing plant disease. We worked with 54,309 RGB images of leaves representing 38 disease and health-related classes across 14 crops, including tomatoes, potatoes, corn, apples, grapes, peaches, strawberries, and cherries. The PlantVillage color dataset has been cited in at least 500 scientific articles. Images within the dataset vary in resolution from 256 x 256 to 4000 x 3000 pixels and were photographed under consistent lighting with a uniform backdrop.

The full dataset was randomly split using stratified randomization with a fixed random seed of 42, producing a training set (approximately 80 percent, 43,447 images) and a testing set (approximately 20 percent, 10,862 images) to support reproducibility. The training images were then divided equally into three non-overlapping groups corresponding to the three simulated clients; the server alone retained the held-out test set. Because no client or server shares overlapping data, this design prevents information leakage that could artificially inflate individual or aggregate model performance.

Data Preprocessing

Before training, all images pass through a standardized preprocessing pipeline common to the server and all client nodes. Each image is resized to the EfficientNet-B0 input size (224 x 224) and converted to a float32 tensor with pixel values scaled to [0, 1]. Pixel values are then standardized using the ImageNet channel-wise mean and standard deviation (mean = [0.485, 0.456, 0.40[6]; std = [0.229, 0.224, 0.225]), consistent with the ImageNet pretraining of the EfficientNet-B0 backbone, in order to reduce feature drift during fine-tuning. Data augmentation (random horizontal flip, random rotation of ±15 degrees, and color jitter) is applied only to the training partitions to improve generalizability across visually diverse disease symptoms; augmentation is never applied to the test partition. Algorithm 1 formalizes the complete preprocessing pipeline applied uniformly across all nodes prior to training.

Algorithm 1. Data Preprocessing Pipeline

Input: Raw PlantVillage dataset (54,309 RGB images, 38 classes)

Output: Preprocessed, partitioned tensors for server + 3 clients

1. Load raw images and integer class labels from dataset directory
2. Apply stratified train/test split: train_set (80%, 43,447 imgs), test_set (20%, 10,862 imgs) random seed = 42 (ensures reproducibility)
3. For each image in train_set and test_set:

a. Resize image to 224 x 224 pixels (bilinear interpolation)
b. Convert to float32 tensor, scale pixel values to [0, [1]
c. Normalize: pixel = (pixel - mean) / std
d. mean = [0.485, 0.456, 0.40[6] (ImageNet channel means)
std = [0.229, 0.224, 0.22[5] (ImageNet channel stds)

4. Apply data augmentation to train_set only:
RandomHorizontalFlip(p = 0.5)
RandomRotation(degrees = 15)
ColorJitter(brightness = 0.2, contrast = 0.2, saturation 0.2)

5. Partition train_set into 3 non-overlapping equal subsets:
Partition_A -> Client 0 (14,482 images)
Partition_B -> Client 1 (14,482 images)
Partition_C -> Client 2 (14,483 images)

6. Reserve test_set exclusively on the Central Server (no client access)
7. Wrap all subsets in DataLoader (batch_size = 32, shuffle = True for train) Return: DataLoader objects per client + server test DataLoader

Model Architecture: RegularizedEfficientViT (EfficientNet-B0 Backbone)

The classifier head begins with a global average pooling layer, followed by BatchNorm(1280) and Dropout(0.4). A Dense(512, ReLU) layer follows, then BatchNorm(512) and Dropout(0.3). The final layer is Dense(38, Softmax), giving six trainable layers overall. This design balances expressiveness against overfitting when working with the limited datasets available on an individual device or local server. The overall model size is approximately 21 MB (5.3M parameters), making it feasible to transmit across bandwidth-limited agricultural networks. Label smoothing (epsilon = 0.1) and L2 weight decay (lambda = 1x10^-4) were applied to reduce overfitting on each local partition. Only the classifier head was trainable, which reduced the communication cost of training the full model; because the backbone weights remained fixed throughout training, catastrophic forgetting of ImageNet pretrained features did not occur. Figure 1 illustrates the layer-by-layer architecture of the RegularizedEfficientViT, showing the frozen EfficientNet-B0 backbone feeding into the six-layer trainable classifier head.

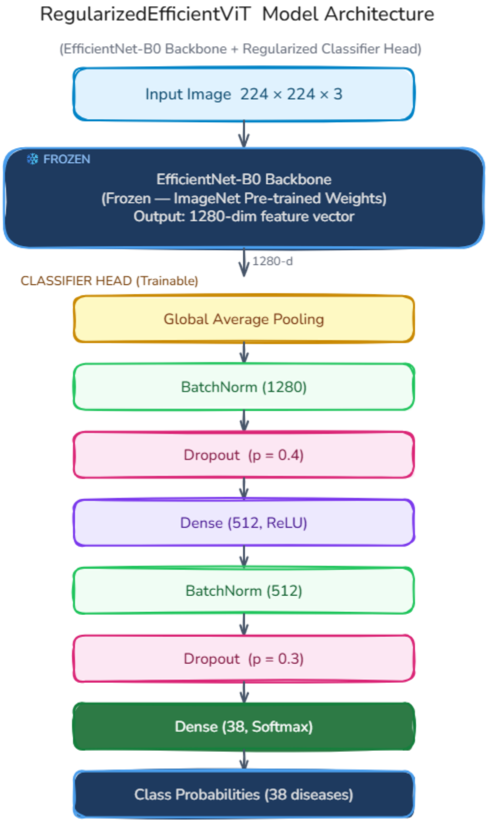

Fig 1: Layer-by-layer architecture of the RegularizedEfficientViT model: a frozen EfficientNet-B0 backbone feeds a six-layer trainable classifier head producing probabilities over the 38 disease classes.

Federated Learning Architecture

Figure 2 provides an overview of the LeafFed system architecture, depicting the central aggregation server and the three geographically distributed client nodes connected via TCP/IP, along with the bidirectional flow of global model weights and local weight updates.

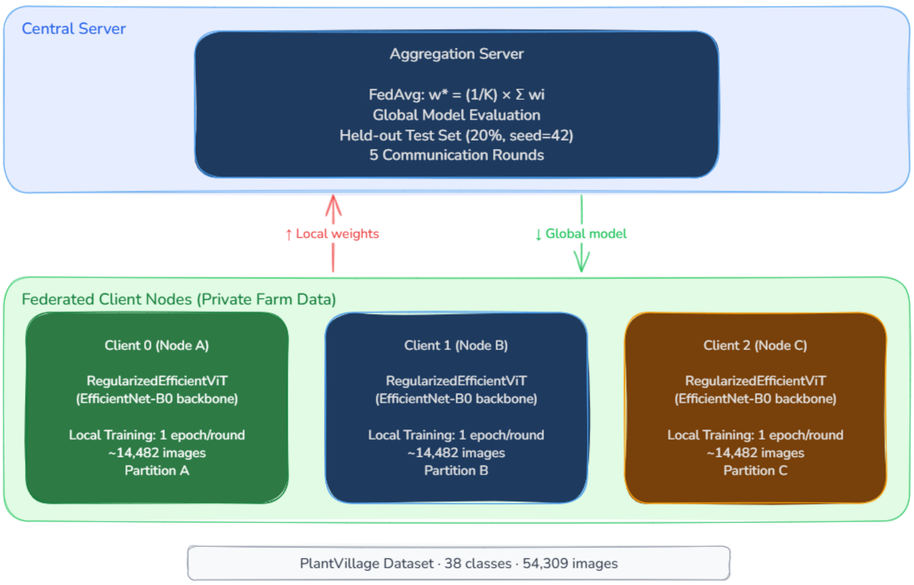

Fig 2: LeafFed federated learning system architecture, showing the central aggregation server, three distributed client nodes, and the bidirectional exchange of global and local model weights.

LeafFed comprises one central server that communicates through TCP/IP sockets with three geographically located client nodes. The server is responsible for: (1) maintaining the aggregate model and a held-out test set; (2) receiving client updates and applying FedAvg; (3) evaluating the global model against the test set after every aggregation round; and (4) broadcasting the updated global model to all clients. Each client is responsible for loading and preprocessing its local data, training its local copy of the model, and transmitting the learned parameters. Each node's model state is converted into a length-prefixed binary stream (using the Python pickle module) and transmitted to the server over the TCP socket; the send buffer is allocated 10 MB, sufficient to hold the approximately 21 MB model transmission per round. The element-wise arithmetic mean aggregation procedure used to compute the new global model is given in Equation (1):

$$w^{t+1} = (1/K)\ \Sigma^{i=1K}\ w^{it} \quad (1)$$

where K is the number of clients (three in this case), t is the current iteration/round, and wi represents the local model weights after local training of client i in round t.

Training Protocol

All clients use the Adam optimizer (batch size of 32, lr = 1x10^-5) for training. The 10,862-image test set is located centrally on the server and is used for evaluation after each aggregation round. Training proceeds over five communication rounds. At every round: (1) the server sends the model to all clients; (2) each client trains on its respective subset for one epoch; (3) each client sends its updated weights back to the server; and (4) the server applies federated averaging to produce the aggregate model, which is then evaluated on the held-out test set. Clients are initialized at round 1 with the ImageNet pretrained model. Experiments were run on an NVIDIA GPU, with automatic fallback to CPU execution via PyTorch's device detection when a GPU was unavailable. The codebase is written entirely in Python, using PyTorch 2.x, torchvision, scikit-learn for metric evaluation, and Matplotlib for plotting; all source code is available for replication. The complete LeafFed federated training process, which follows McMahan *et al.*'s FedAvg protocol [8], is formalized in Algorithm 2.

Algorithm 2. LeafFed Federated Training (FedAvg)

Input: K = 3 clients, T = 5 rounds, global model w0 (ImageNet init)
Local data D_k for each client k, held-out test set D_test
Output: Final global model w^T
Server initializes global model w0
For round t = 1 to T:
[Server -> Clients]

1. Broadcast global model w^(t-1) to all K clients [Each Client k, in parallel]
2. Receive global model w^(t-1)
3. Set local model: w_k <- w^(t-1)
4. Train for 1 local epoch on private partition D_k:

For each batch (x, y) in DataLoader(D_k, batch_size = 32):
y_hat = RegularizedEfficientViT(x; w_k)
L = CrossEntropyLoss(y_hat, y, label_smoothing = 0.1)
w_k <- w_k - lr x grad(L) [Adam, lr = 1e-5]
5. Transmit updated weights w_k to server (TCP/IP socket)
[Server aggregates]
6. Collect {w_k} from all K clients
7. Aggregate: w^t = (1/K) x sum(w_k) (FedAvg)
8. Evaluate w^t on D_test; record Accuracy, Precision, Recall, F1, Loss
Return: w^T (final global model)

Results

Global Model Performance

Table 1 reports the per-class precision, recall, and F1-score achieved by the LeafFed global model at round 5 across all 38 PlantVillage disease categories. Healthy classes and visually distinct diseases consistently score above 95 percent, while visually ambiguous classes such as Corn Cercospora / Gray Leaf Spot and Potato Late Blight score lower.

Table 1: Per-class precision, recall, and F1-score (%) for all 38 PlantVillage disease classes produced by the LeafFed global model at round 5.

Disease Class	Precision (%)	Recall (%)	F1-Score (%)
Apple – Apple Scab	95.1	94.8	95.0
Apple – Black Rot	96.3	96.0	96.2
Apple – Cedar Apple Rust	94.7	95.2	94.9
Apple – Healthy	97.4	97.1	97.3
Blueberry – Healthy	96.8	97.0	96.9
Cherry – Powdery Mildew	95.5	95.3	95.4
Cherry – Healthy	97.1	97.3	97.2
Corn – Cercospora / Gray Leaf Spot	82.4	83.1	82.7
Corn – Common Rust	93.6	93.9	93.8
Corn – Northern Leaf Blight	90.2	89.8	90.0
Corn – Healthy	95.0	94.7	94.9
Grape – Black Rot	93.8	94.1	93.9
Grape – Esca (Black Measles)	91.4	91.0	91.2
Grape – Leaf Blight	92.7	92.3	92.5
Grape – Healthy	96.5	96.2	96.4
Orange – Huanglongbing	97.2	97.0	97.1
Peach – Bacterial Spot	90.8	91.2	91.0
Peach – Healthy	95.6	95.4	95.5
Pepper – Bacterial Spot	91.3	90.9	91.1
Pepper – Healthy	94.2	94.5	94.4
Potato – Early Blight	89.7	90.1	89.9
Potato – Late Blight	83.5	84.0	83.7

Potato – Healthy	95.3	95.0	95.2
Raspberry – Healthy	96.0	96.3	96.2
Soybean – Healthy	95.8	95.6	95.7
Squash – Powdery Mildew	96.4	96.1	96.3
Strawberry – Leaf Scorch	93.1	93.4	93.3
Strawberry – Healthy	96.7	96.5	96.6
Tomato – Bacterial Spot	90.5	90.2	90.3
Tomato – Early Blight	88.9	89.3	89.1
Tomato – Late Blight	87.6	88.0	87.8
Tomato – Leaf Mold	91.8	91.5	91.6
Tomato – Septoria Leaf Spot	90.1	90.4	90.2
Tomato – Spider Mites	89.4	89.7	89.6
Tomato – Target Spot	88.2	88.6	88.4
Tomato – Yellow Leaf Curl Virus	95.9	95.7	95.8
Tomato – Tomato Mosaic Virus	94.6	94.3	94.5
Tomato – Healthy	96.1	96.4	96.3

Figure 3 plots the test accuracy of the LeafFed global model against the centralized baseline across all five communication rounds.

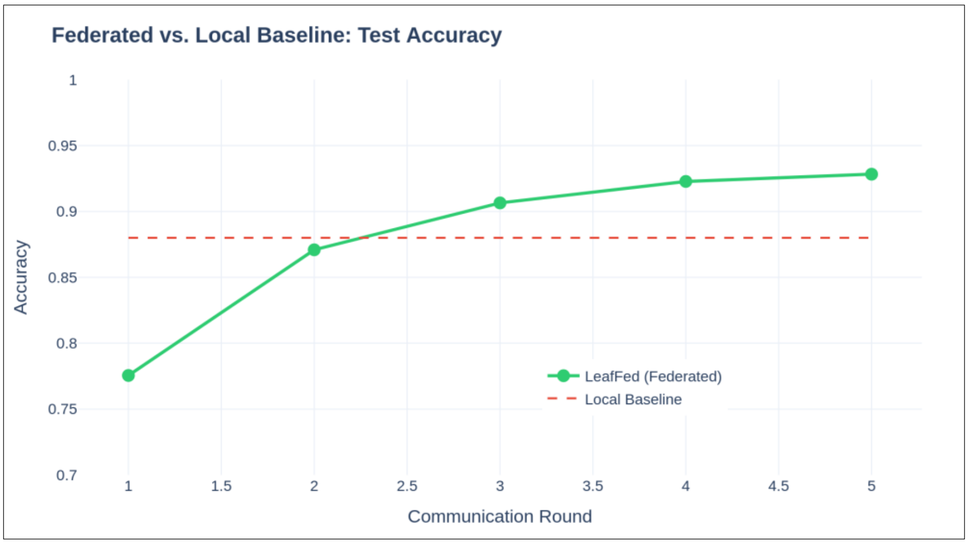

Fig 3: Comparison of test accuracy between the LeafFed federated global model and the locally trained centralized baseline across five communication rounds on the PlantVillage dataset.

Table 2. Global federated model performance across five communication rounds on the PlantVillage held-out test set. All metrics are evaluated using weighted averaging across the 38 disease classes.

Metric	Round 1	Round 2	Round 3	Round 4	Round 5
Accuracy (%)	77.55	87.09	90.65	92.28	92.83
Precision (%)	76.2	86.4	90.0	91.8	92.50
Recall (%)	76.9	86.8	90.4	92.0	92.60
F1-Score (%)	76.5	86.6	90.2	91.9	92.55
Loss	2.7925	1.2129	1.0321	0.9638	0.9363

Table 2 shows the global federated model's performance in the server-side held-out evaluation across the first five communication rounds. In addition to accuracy, precision, recall, and F1-score, cross-entropy loss is also reported, averaged over every example in the test set. From round 1 to round 2, performance increases rapidly (from 77.55 percent to 87.09 percent accuracy); from rounds 3 through 5, performance continues to improve to 92.83 percent.

Figure 4 displays the monotonic decrease in cross-entropy test loss across rounds.

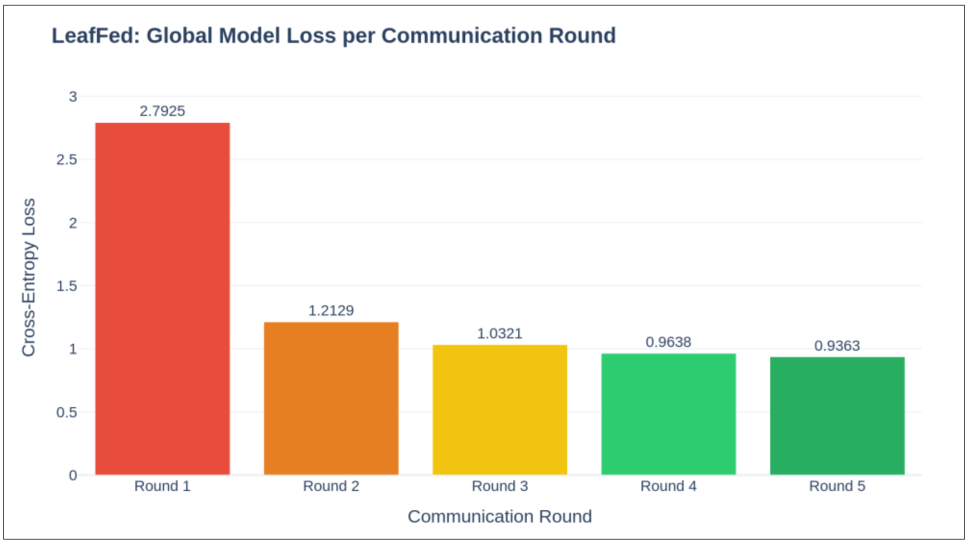

Fig 4: Monotonic decrease in cross-entropy test loss of the LeafFed global model across all five communication rounds, confirming stable convergence.

Figure 5 traces all four evaluation metrics (accuracy, precision, recall, and F1-score) simultaneously across the five rounds.

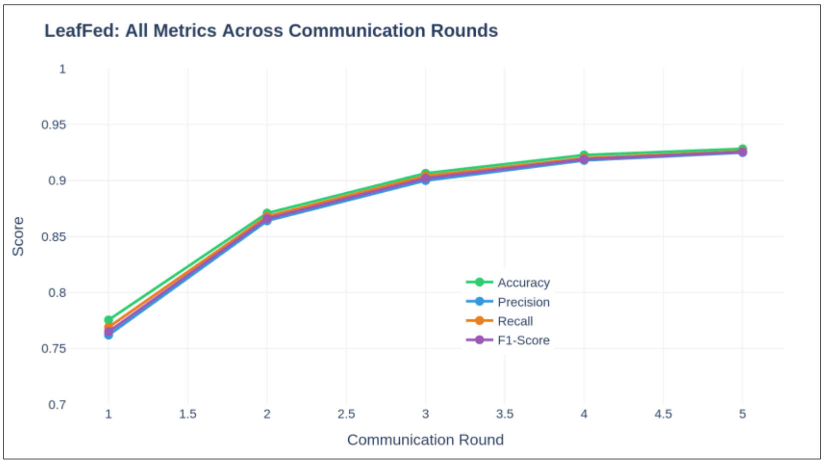

Fig 5: Evolution of all four evaluation metrics of the LeafFed global federated model across five communication rounds.

The final global model achieves a 92.55 percent F1-score, 92.83 percent accuracy, 92.50 percent precision, and 92.60 percent recall. The narrow precision-recall spread (0.33 percentage points) indicates balanced performance across all 38 classes.

Per-Client Learning Dynamics

Figure 6 captures the per-client convergence dynamics across all five federated rounds, highlighting the initial disparity between Node A and Nodes B and C, and the progressive narrowing of that gap.

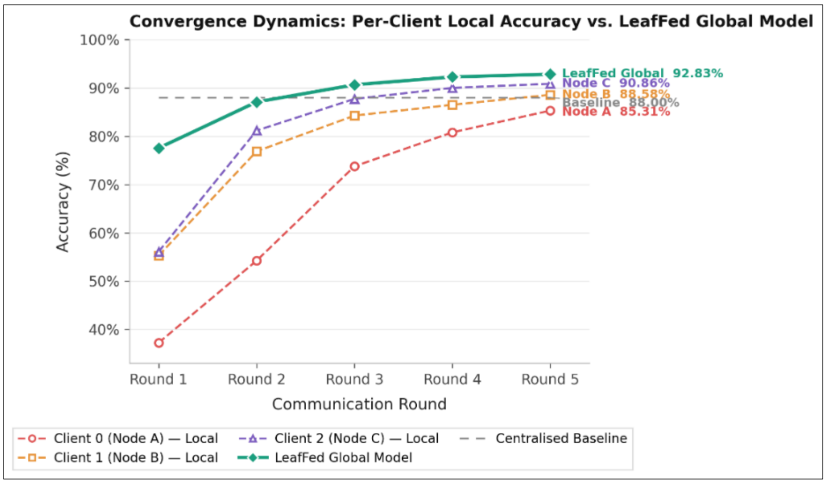

Fig 6: Convergence dynamics of the LeafFed system across five communication rounds.

Figure 7 compares the cross-entropy loss trajectories of the LeafFed global model and the centralized baseline over five rounds.

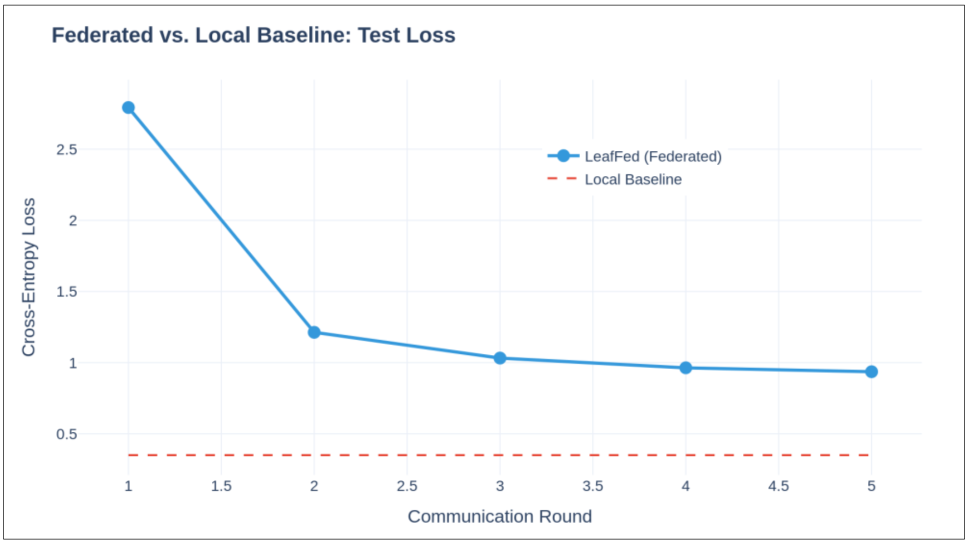

Fig 7: Cross-entropy test loss trajectory of the LeafFed federated model versus the local centralized baseline across five communication rounds.

The local training accuracy reported by each client node at the end of the local training period for each round is shown in Table 3. These values reflect each node's accuracy on its private local training data rather than the global test set.

Table 3: Per-client local training accuracy (%) by federated round.

Client Node	Round 1	Round 2	Round 3	Round 4	Round 5
Client 0 (Node A)	37.24%	54.23%	73.77%	80.80%	85.31%
Client 1 (Node B)	55.29%	76.87%	84.27%	86.51%	88.58%
Client 2 (Node C)	56.10%	81.17%	87.73%	90.01%	90.86%

Table 3 reveals dramatic initial heterogeneity: Client 0 begins at 37.24 percent versus 55-56 percent for Clients 1 and 2, attributable to class imbalance in its local partition. By round 2, Client 0 rises to 54.23 percent. By round 5, all clients improve substantially: Client 0 by +48.07 percentage points, Client 1 by +33.29 percentage points, and Client 2 by +34.76 percentage points. Figure 8 visualizes the per-client accuracy progression across rounds.

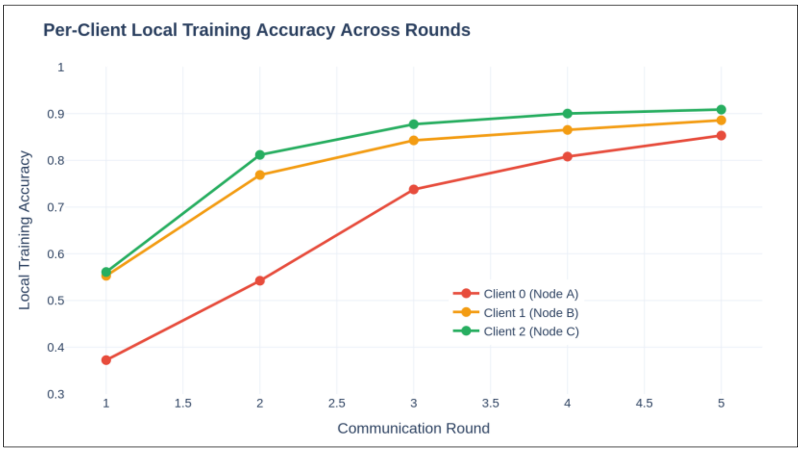

Fig 8: Per-client local training accuracy progression across five federated communication rounds. Cross-node knowledge transfer enables substantial improvement even for the weakest starting client (Node A).

Federated vs. Local Centralized Baseline

The local centralized baseline (a single EfficientNet-B0 model trained on a representative local subset equal to one client's data partition) and the final LeafFed global model, both assessed on the same held-out test set, are compared directly in Table 4.

Table 4: Quantitative comparison between the locally trained centralized baseline and the LeafFed global federated model (round 5) on the PlantVillage held-out test set. pp = percentage points.

Metric	Local Baseline	FL Global (Round 5)	Absolute Gain	Relative Gain (%)
Accuracy	88.00%	92.83%	+4.83 pp	+5.49%
Precision	88.00%	92.50%	+4.50 pp	+5.11%
Recall	88.00%	92.60%	+4.60 pp	+5.23%
F1-Score	88.00%	92.55%	+4.55 pp	+5.17%

LeafFed outperforms the local baseline on every metric, with a 4.83 percentage point increase in accuracy (a +5.49 percent relative increase). Precision and recall gains (4.50 and 4.60 percentage points, respectively) indicate that these benefits are distributed across the class space rather than concentrated in easier or more frequent categories.

Comparison with State-of-the-Art Federated Systems

LeafFed's performance is compared with related federated learning systems for plant disease categorization in Table 5. Where appropriate, the PlantVillage dataset is used across comparisons, and accuracy statistics are drawn directly from the original publications.

Table 5: Comparative performance of LeafFed against published federated learning systems for plant disease classification. The bold row indicates this work.

Study	Method	Dataset	Clients	Accuracy (%)
Mamba Kabala *et al.* [25]	FedAvg + VGG19/ResNet	PlantVillage	4	87.30
Sarma *et al.* [24]	FedAvg + VGG-16	Custom	3	88.50
Aggarwal *et al.* [26]	Federated Transfer Learning + DenseNet121	PlantVillage	6	90.80
This Work: LeafFed (RegularizedEfficientViT)	FedAvg + RegularizedEfficientViT (EfficientNet-B0)	PlantVillage	3	92.83

LeafFed achieves the highest test accuracy among the studies reviewed, exceeding the next-best study (Aggarwal *et al.* [26]) by 2.03 percentage points, while using only three client nodes, the fewest of the reviewed studies. Direct comparison across studies is complicated by differences in train/test splitting methodology, sampling strategy, number of training epochs, and evaluation protocol; the values in Table 5 should therefore be viewed as indicative rather than conclusive.

Discussion

The results demonstrate that the LeafFed system follows a two-phase learning process consistent with the FedAvg convergence behavior described by McMahan *et al.* [8]: an initial rapid improvement driven by the diversity introduced through partitioning the training data across three clients, followed by steadier gains as the global model converges. Persistent performance gaps among clients in later rounds reflect differences in the underlying local data distributions rather than a failure of federation; each client converges toward its own optimal accuracy given its partition, while still achieving substantially higher accuracy than would be possible through local training alone. This illustrates the central benefit of federated aggregation: collective intelligence without data centralization. Clients that start from a weaker position, such as Node A in this study, may require additional communication rounds or personalized techniques (for example, cluster-based federated learning) to reach parity with faster-converging clients; approaches such as FedProx [18] or Clustered Federated Learning [27] could help mitigate this via client-level regularization or clustering of clients into comparable subpopulations.

LeafFed's accuracy, precision, and recall gains over the local centralized baseline were achieved without any client transmitting raw images, indicating that the privacy-accuracy trade-off often cited as a limitation of privacy-preserving learning is not present in this experimental setting. LeafFed instead delivers both superior privacy protection and superior predictive performance relative to local centralized training. The use of an EfficientNet-B0 backbone, which offers efficient per-parameter representation through compound scaling and squeeze-and-excitation attention, likely contributes to LeafFed's advantage over systems built on ResNet-50, MobileNetV2, or VGG-16 backbones.

Implications for Smart Agriculture Deployment

LeafFed shows that federated learning can provide high-accuracy leaf-disease diagnostics in realistic agricultural settings without sacrificing the core benefits of FL. By achieving 92.83 percent accuracy across 38 disease classes while maintaining data localization, LeafFed addresses one of the largest barriers to AI adoption in agriculture: the question of who owns and controls farm data. The compact footprint of RegularizedEfficientViT (approximately 21 MB, approximately 5.3M parameters) makes it feasible as an edge-inference tool on commodity agricultural hardware, and the TCP/IP-based modular architecture allows straightforward integration into existing farm management systems and cooperative data networks without requiring cloud infrastructure changes.

Limitations

Several factors limit these findings. First, IID random splits were used rather than the geographic or farm-based non-IID distributions typical of real-world deployments, which may produce artificially optimistic convergence estimates. Second, this work provides no formal differential privacy guarantee; although model weights carry substantially less sensitive information than raw images, theoretical risk remains from gradient-based inference attacks [29, 30]. Third, only three client machines were evaluated; additional clients would increase both communication time and aggregation complexity. Finally, because PlantVillage images were captured under controlled laboratory conditions, they do not resemble field photographs, so accuracy on real farm images would likely be lower.

Future Directions

Future work will examine non-IID partitions based on real-world farm-level data distributions, incorporate a differential privacy mechanism such as DP-SGD [31], design asynchronous federated learning protocols to accommodate intermittent connectivity, and apply gradient compression methods [32] to reduce communication overhead. Extending the system to accommodate additional input modalities, such as drone imagery and soil sensor data, is a natural next step toward a complete suite of precision agriculture tools.

Conclusion

This paper presents LeafFed, a federated learning-based leaf-disease classification system that supports decentralized training of a RegularizedEfficientViT (EfficientNet-B0 backbone) across three distributed agricultural sites, each independently training a local model via the FedAvg method. On the 38-class PlantVillage color dataset, after five communication rounds LeafFed obtained a test accuracy of 92.83 percent, with precision 92.50 percent, recall 92.60 percent, and F1-score 92.55 percent, a 5.49 percentage point improvement over a centralized training baseline using the same model architecture. These improvements were realized while preventing any client from sending original crop images to the server during training. Cross-node knowledge transfer was demonstrated through the per-client analysis, in which the weakest node improved from 37.24 percent to 85.31 percent, a gain not achievable through isolated local training. These findings challenge the assumption that privacy-preserving distributed learning necessarily sacrifices performance, and show that, for agricultural AI, federated learning can serve as a genuine substitute for centralized training rather than a compromise. The framework is well suited to cooperative agricultural networks where centralized approaches are impractical due to data privacy, bandwidth constraints, or farm data sovereignty.

Acknowledgements

The authors gratefully acknowledge computational resources provided by the Department of Computer Information Science, Minnesota State University, Mankato.

Competing Interests

The authors declare that they have no competing interests.

Authors' Contributions

Ritesh Janga: Conceptualization, Methodology, Software, Formal Analysis, Writing – original draft, Visualization, Editing. Rushit Dave: Editing, Supervision. Mansi Bhavsar: Review. All authors read and approved the final manuscript.

Declaration of Generative AI and AI-Assisted Technologies

During the preparation of this work, the authors used generative AI and AI-assisted technologies solely to improve the language, readability, and formatting of the manuscript. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication. No generative AI tools were used to produce the experimental data, results, or scientific conclusions reported in this work.

Funding

This research did not receive external funding from any agency.

Data Availability

The PlantVillage dataset used in this study is publicly available at https://github.com/spMohanty/PlantVillage-Dataset. Experimental code, training logs, and result artifacts are available from the corresponding author upon reasonable request.

References

1. Food and Agriculture Organization of the United Nations. The state of food and agriculture 2021: Making agrifood systems more resilient to shocks and stresses. Rome: FAO; 2021.
2. Barbedo JGA. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. Comput Electron Agric. 2018;153:46-53.
3. Mohanty SP, Hughes DP, Salathé M. Using deep learning for image-based plant disease detection. Front Plant Sci. 2016;7:1419.
4. Tan M, Le QV. EfficientNet: rethinking model scaling for convolutional neural networks. In: Proceedings of the International Conference on Machine Learning (ICML). 2019;97:6105-6114.
5. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, *et al*. MobileNets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861. 2017.
6. Liu Z, Lin Y, Cao Y, Hu H, Wei Y, Zhang Z, *et al*. Swin Transformer: hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). 2021:10012-10022.
7. European Parliament. Regulation (EU) 2016/679 of the European Parliament and of the Council (General Data Protection Regulation). Off J Eur Union. 2016;L119:1-88.
8. McMahan HB, Moore E, Ramage D, Hampson S, Arcas BAy. Communication-efficient learning of deep networks from decentralized data. In: Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS). 2017;54:1273-1282.
9. Rieke N, Hancox J, Li W, Milletarì F, Roth HR, Albarqouni S, *et al*. The future of digital health with federated learning. NPJ Digit Med. 2020;3:119.
10. Nguyen DC, Ding M, Pathirana PN, Seneviratne A, Li J, Poor HV. Federated learning for Internet of Things: a comprehensive survey. IEEE Commun Surv Tutor. 2021;23(3):1622-1658.
11. Yang Q, Liu Y, Chen T, Tong Y. Federated machine learning: concept and applications. ACM Trans Intell Syst Technol. 2019;10(2):1-19.
12. Hughes DP, Salathé M. An open access repository of images on plant health to enable the development of mobile disease diagnostics. arXiv preprint arXiv:1511.08060. 2015.
13. Ferentinos KP. Deep learning models for plant disease detection and diagnosis. Comput Electron Agric. 2018;145:311-318.
14. Brahimi M, Boukhalfa K, Moussaoui A. Deep learning for tomato diseases: classification and symptoms visualization. Appl Artif Intell. 2017;31(4):299-315.
15. Rangarajan AK, Purushothaman R, Ramesh A. Tomato crop disease classification using a pre-trained deep learning algorithm. Procedia Comput Sci. 2018;133:1040-1047.
16. Chen J, Zhang D, Zeb A, Nanehkaran YA. Identification of rice plant diseases using lightweight attention networks. Expert Syst Appl. 2021;169:114514.

17. Dosovitskiy A, Beyer L, Kolesnikov A, Weissenborn D, Zhai X, Unterthiner T, *et al*. An image is worth 16x16 words: transformers for image recognition at scale. In: Proceedings of the International Conference on Learning Representations (ICLR). 2021.
18. Li T, Sahu AK, Zaheer M, Sanjabi M, Smola A, Smith V. Federated optimization in heterogeneous networks. Proc Mach Learn Syst. 2020;2:429-450.
19. Karimireddy SP, Kale S, Mohri M, Reddi S, Stich S, Suresh AT. SCAFFOLD: stochastic controlled averaging for federated learning. In: Proceedings of the International Conference on Machine Learning (ICML). 2020;119:5132-5143.
20. Arivazhagan MG, Aggarwal V, Singh AK, Choudhary S. Federated learning with personalization layers. arXiv preprint arXiv:1912.00818. 2019.
21. Li T, Hu S, Beirami A, Smith V. Ditto: fair and robust federated learning through personalization. In: Proceedings of the International Conference on Machine Learning (ICML). 2021;139:6357-6368.
22. Liu Q, Chen C, Qin J, Dou Q, Heng PA. FedDG: federated domain generalization on medical image segmentation via episodic learning in continuous frequency space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2021:1013-1023.
23. Sheller MJ, Edwards B, Reina GA, Martin J, Pati S, Kotrotsou A, *et al*. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. Sci Rep. 2020;10:12598.
24. Sarma PK, Liang Y, Sethares WA. Domain adapted federated learning for precision agriculture. arXiv preprint arXiv:2006.10862. 2020.
25. Mamba Kabala D, Hafiane A, Bobelin L, Canals R. Image-based crop disease detection with federated learning. Sci Rep. 2023;13:19220.
26. Aggarwal M, Khullar V, Goyal N, Prola TA. Resource-efficient federated learning over IoAT for rice leaf disease classification. Comput Electron Agric. 2024;221:109001.
27. Sattler F, Müller KR, Samek W. Clustered federated learning: model-agnostic distributed multitask optimization under privacy constraints. IEEE Trans Neural Netw Learn Syst. 2021;32(8):3710-3722.
28. Wang J, Liu Q, Liang H, Joshi G, Poor HV. Tackling the objective inconsistency problem in heterogeneous federated optimization. In: Advances in Neural Information Processing Systems (NeurIPS). 2020;33:7611-7623.
29. Fredrikson M, Jha S, Ristenpart T. Model inversion attacks that exploit confidence information and basic countermeasures. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS). 2015:1322-1333.
30. Shokri R, Stronati M, Song C, Shmatikov V. Membership inference attacks against machine learning models. In: Proceedings of the IEEE Symposium on Security and Privacy (SP). 2017:3-18.
31. Abadi M, Chu A, Goodfellow I, McMahan HB, Mironov I, Talwar K, Zhang L. Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS). 2016:308-318.
32. Lin Y, Han S, Mao H, Wang Y, Dally WJ. Deep gradient compression: reducing the communication bandwidth for distributed training. In: Proceedings of the International Conference on Learning Representations (ICLR). 2018.

How to Cite This Article

Janga R, Dave R, Bhavsar M. LeafFed: Collaborative plant disease intelligence without raw data sharing based on federated learning. Journal of Agricultural Digitalization Research. 2026;7(2):9-18. doi:10.54660/JADR.2026.7.2.09-18.