

AGRIFOLD: AGRIficulture Federated learning for Optimized Leaf disease Detection

Francesco Piccialli , Ciro Della Bruna , Diletta Chiaro , Pian Qi , Martina Savoia

University of Naples Federico II, Department of Mathematics and Applications "R. Caccioppoli", M.O.D.A.L. - Mathematical mOdelling and Data Analysis research group and laboratory, Napoli, Italy

ARTICLE INFO

Keywords:
 Deep learning
 Agritech
 Federated learning
 Leaf disease detection
 Sustainable agriculture
 Precision agriculture

ABSTRACT

Efficient and accurate detection of plant leaf diseases is essential for protecting crop health and promoting sustainable and precision agriculture practices. However, the decentralized nature of agricultural data, combined with the inherent limitations of centralized Machine Learning (ML), presents significant challenges for developing scalable, privacy-preserving solutions. In this paper, we introduce AGRIFOLD, a Federated Learning (FL) framework designed to enable collaborative training of a lightweight Convolutional Neural Network (CNN) across diverse and distributed datasets while maintaining data privacy. By integrating an Efficient Channel Attention (ECA) mechanism into the VGG16 architecture, AGRIFOLD significantly improves classification accuracy and enhances interpretability through heatmaps that highlight regions affected by diseases. We evaluate the FL model using various aggregation methods, including FedAvg, FedProx, SCAFFOLD, FedBN, and FedDF, obtaining good accuracy levels for all tested aggregation strategies, with SCAFFOLD achieving the best overall performance. The model's lightweight design, optimized through ablation and pruning techniques, facilitates deployment on resource-constrained edge devices. Additionally, to further support farmers' decision-making, the framework incorporates a natural language processing-based recommender system that provides tailored treatment suggestions. Comprehensive experiments conducted on 12 heterogeneous datasets demonstrate high classification accuracy across 9 distinct leaf disease classes and healthy leaves, underscoring the practical potential of FL-based solutions for sustainable, real-world agricultural applications. The AGRIFOLD source code is available at <https://github.com/MODAL-UNINA/AGRIFOLD>.

1. Introduction

Managing crop diseases is essential in agriculture to improve yield and quality while minimizing the economic and visual damage caused by plant diseases. While ongoing research focuses on understanding the causes and developing effective treatments for crop diseases, early pathogen detection and continuous monitoring of plant health are vital to preventing disease spread and ensuring efficient management (Martinelli et al., 2015). Infections caused by pathogens such as fungi, bacteria, and viruses can compromise this vital function of the plant and lead to widespread crop failure. When a plant becomes infected, symptoms manifest on various parts of the plant, leading to a substantial agronomic impact. To enhance early detection and improve disease management strategies, advanced diagnostic techniques like remote sensing (RS) and spectroscopy are being increasingly adopted. Compared to traditional methods, they offer faster and more comprehensive disease detection across different spatial and temporal scales. When

calibrated with reliable reference data, these tools enable earlier identification of infections, supporting timely and effective interventions (López et al., 2003). Despite these advancements, large-scale disease detection remains a complex challenge. Detecting and safeguarding crops from pathogens at scale is a demanding and time-consuming task, making it nearly impossible for humans to inspect each plant individually. This challenge is further amplified by the vast size of agricultural fields, where the large number of plants makes manual monitoring impractical. Early detection is crucial, as even a single infected plant can rapidly spread the disease across the entire field. Moreover, many farmers lack the necessary expertise to accurately identify plant diseases and apply appropriate treatments. While hiring specialists could improve disease management, the associated costs can be prohibitive. As a result, farmers may resort to indiscriminate pesticide use, which not only fails to address the problem effectively but also harms the soil and surrounding ecosystem (Sankaran, Mishra, Ehsani, & Davis, 2010), (Gupta, Hans, & Chand, 2020). Over the years, several approaches have been proposed

* Corresponding author.

E-mail addresses: francesco.piccialli@unina.it (F. Piccialli), ciro.dellabruna@unina.it (C. Della Bruna), diletta.chiaro@unina.it (D. Chiaro), pian.qi@unina.it (P. Qi), martina.savoia@unina.it (M. Savoia).

for classifying plant diseases, ranging from traditional Machine Learning (ML) models such as Naïve Bayes (NB), k-Nearest Neighbors (k-NN), Support Vector Machines (SVM), and Random Forest (RF) to more advanced Deep Learning (DL) techniques. DL has led to significant progress in automating the detection and management of crop diseases. Convolutional Neural Networks (CNNs) have emerged as a widely used technique in DL and have drawn significant attention in numerous scientific disciplines, especially agriculture (Liu, Wang, Chen, Han, & Yang, 2018). However, CNNs have proven to be particularly well-suited for this task due to their ability to automatically extract relevant features from images. Unlike conventional models, which often require hand-crafted feature selection, CNNs can learn complex spatial hierarchies and patterns directly from image data, leading to improved classification performance. This capability makes them a preferred choice in the fields of computer vision and image processing, where accurate and reliable feature extraction is essential. Given their adaptability and effectiveness, CNN-based approaches continue to be at the forefront of plant disease detection research, providing a robust foundation for precision agriculture and automated crop monitoring (Demilie, 2024). Despite their success, many existing approaches rely on centralized ML models trained on single datasets, often in controlled conditions, limiting their generalization to real-world agricultural settings.

Current models, especially those trained on benchmark datasets such as PlantVillage, achieve high accuracy in ideal, laboratory-based conditions but may fail to account for the heterogeneous, non Independently and Identically Distributed (non-IID) nature of real-world agricultural data. These models require large amounts of centralized data, which presents privacy concerns and significant risks when dealing with sensitive agricultural information. In contrast, FL offers a decentralized alternative that preserves data privacy by keeping training data local, making it more suitable for practical agricultural applications (Li, Sahu, Talwalkar, & Smith, 2020a; Yang, Liu, Chen, & Tong, 2019; Zhang et al., 2021). FL inherently involves multiple conflicting objectives, particularly in scenarios with non-IID data distributions across clients. In such settings, optimizing solely for the global model's accuracy may result in poor or unfair performance on certain local datasets. To address these challenges, FL can be viewed as a multi-objective optimization (MOO) problem, aiming to balance overall global performance with fairness and consistency across individual clients (Shen et al., 2025). Recent advances in MOO have introduced robust and efficient algorithms designed to handle trade-offs across competing objectives. Methods such as the Multi-objective Geometric Mean Optimizer (MOGMO) (Pandya, Kalita, Jangir, Ghadai, & Abualigah, 2024a), Multi-objective Exponential Distribution Optimizer (MOEDO) (Kalita et al., 2024), and the Multi-objective RIME Algorithm (Pandya et al., 2024a) have shown remarkable capabilities in managing performance trade-offs in complex, heterogeneous environments. Most FL-based studies in plant disease detection have been limited in scope, often assuming homogeneous dataset, a small number of plant species and disease types, or a small number of clients. This simplification does not reflect the complexities of real agricultural scenarios, where data can vary significantly due to different crops, environmental conditions, and collection methods. Moreover, existing works frequently overlook the need for lightweight, resource-efficient models that can be deployed on edge devices used in agricultural environments.

To address these gaps, this study proposes AGRIFOLD, which stands for AGRIculture Federated learning for Optimized Leaf Disease detection, a novel FL framework that introduces several improvements compared to previous studies:

- **Use of multiple datasets:** Unlike prior studies that rely on limited datasets, this work aggregates 12 distinct datasets representing crops in both controlled and natural environments. Moreover, images differ in quality, quantity, size, acquisition device, and lighting conditions. This diversity ensures that model performance is evaluated across heterogeneous data sources.

- **Realistic FL simulation:** The FL environment mimics real-world conditions by assigning each dataset to a different client, simulating non-IID distributions and variable data amounts across clients, a critical improvement over simplified setups in previous research. All clients collaboratively train a unified classification model with 10 output classes, even though in practice, each local dataset may not contain samples from all classes. This setup reflects a real-world scenario in agriculture, where a client might only observe a subset of possible diseases.
- **Enhanced disease classification:** A system capable of accurately identifying diseases across multiple fruits, vegetables, and crops is proposed. By adapting VGG16 (Simonyan & Zisserman, 2014), a lightweight CNN model is developed to efficiently process images while maintaining high classification performance. The use of advanced structures, such as the Efficient Channel Attention (ECA) (Wang et al., 2020a) module, allows the model to effectively extract key features associated with shared disease characteristics in diverse crops.
- **eXplainable AI (XAI):** AGRIFOLD generates heatmaps that highlight diseased regions, enhancing interpretability and supporting better decision-making for farmers and researchers.
- **Personalized treatment recommender:** A recommendation system guides farmers by suggesting tailored treatments using web-scraped data and chatbot interactions, providing actionable advice for effective disease management.

By combining these innovations, AGRIFOLD bridges the gap between research and real-world applications by enabling scalable, privacy-preserving, and explainable plant disease detection. It offers a robust solution for managing diverse agricultural data and providing real-time, practical assistance to farmers.

2. Related work

This section presents a structured literature review. We begin examining works on Leaf Disease Detection (LDD) and classification using centralized approaches, followed by an overview of studies that apply FL to this domain. Lastly, we discuss the integration of explainability techniques and the development of lightweight models tailored for deployment in real-world agricultural settings. Table 1 provides a summarized overview of the referenced studies.

2.1. Leaf diseases detection

Plant diseases have a significant impact on the growth and health of their respective species, making early detection crucial. The area of DL research appears to have great potential in the detection and classification of plant diseases (Ashok et al., 2020; Jiang, Li, Yang, & Yu, 2020; Sholihati, Sulistijono, Risnumawan, & Kusumawati, 2020; Vishnoi, Kumar, Kumar, Mohan, & Khan, 2022). CNNs have become widely adopted for this task, due to their ability to effectively analyze and classify images. For example, in Zhong and Zhao (2020) a DenseNet-121 has been used to identify apple leaf diseases through regression, multi-label classification, and a focus loss function. Using 2462 images across six diseases, the used methods achieved test accuracies of 93.51 %, 93.31 %, and 93.71 %, respectively, demonstrating their effectiveness. In Rozaqi and Sunyoto (2020), a customized CNN is used to detect potato leaf diseases (early and late blight, and healthy leaves). With a 70:30 data split, the model achieved 97 % accuracy on the training and 92 % on the test, employing a batch size of 20 and 10 epochs. A CNN-based model for automatic plant disease identification is presented in Khobragade et al. (2022), classifying potato leaves into healthy, early blight, and late blight categories. Using a dataset of 5162 original and 82592 augmented images, the model achieved 98.07 % accuracy, demonstrating its ability to learn features directly from raw images. A DL-based CNN model (Singla et al., 2022) was developed to classify rice hispa images from

Table 1

Comparison of various works in LDD.

Work	Applied crops	Model	Dataset	Focus	Details
Zhong and Zhao (2020)	Apple	DenseNet121	Dataset derived from AI-Challenger-Plant-Disease-Recognition	Centralized	6 classes
Rozaiq and Sunyoto (2020)	Potato	Customized CNN	PlantVillage	Centralized	3 classes
Khobragade, Shriwas, Shinde, Mane, and Padole (2022)	Potato	Customized CNN	PlantVillage, Potato Disease Leaf Dataset	Centralized	2 classes
Singla, Sharma, Kukreja, and Bansal (2022)	Rice	Customized CNN	Self-acquired	Centralized	3 classes
Jung et al. (2023)	Bell Pepper, Potato, Tomato	ResNet50, AlexNet, GoogleNet, VGG19, EfficientNet	PlantVillage, AI-hub	Centralized	3 classes
Mac, Nguyen, Dang, Nguyen, and Nguyen (2024)	Tomato	VGG19, Inception-v3, DenseNet-201	PlantVillage	Centralized	10 classes
Prasad et al. (2024)	Grape	VGG16	Grapevine Disease Dataset	Centralized	4 classes
Rahman, Ahmad, Jon, Salam, and Rabbi (2024)	Tea	Customized CNN	Self-acquired	Centralized	4 classes
Kini, Prema, and Pai (2024)	Black Pepper	Inception V3, GoogleNet, SqueezeNet	Self-acquired	Centralized	6 classes
Vats, Kukreja, and Mehta (2024)	Jute	Customized CNN	Self-acquired	FL	4 classes FedAvg
Joshi, Aeri, Kukreja, and Mehta (2024)	Sunflower	Customized CNN	Not specified	FL	5 classes FedAvg
Suryavanshi, Kukreja, Upadhyay, Mehta, and Thapliyal (2024a)	Broccoli	Customized CNN	Data collected from five different customers	FL	5 classes Strategy not specified
Upadhyay, Verma, Kukreja, and Mehta (2024)	Aloe Vera	Customized CNN	A wide range of aloe vera leaf samples obtained from six clients	FL	4 classes FedAvg
Hari, Singh, and Singh (2024)	Apple, Maize, Tomato	H-CNN	PlantVillage	FL	7 classes Proposed FDL
Fahim-Ul-Islam et al. (2024)	Wheat	CoAtNets, Swin Transformer V2	Plant Disease Classification Merged Dataset, Wheat Nitrogen Deficiency and Leaf Rust Image	FL	4 classes FedMax
Antico, Moreira, and Moreira (2024)	Maize	AlexNey, SqueezeNet, ResNet-18, VGG11, ShuffleNet	PlantVillage	FL	4 classes FedAvg
Aggarwal, Khullar, Goyal, and Prola (2024)	Rice	EfficientNetB3	Agricultural database	FL	4 classes FedAvg
Suryavanshi, Kukreja, Malhotra, Mehta, and Choudhary (2024a)	Cauliflower	Customized CNN	Data collected from five different customers	FL	5 classes FedAvg
Singh, Chattopadhyay, and Singh (2024)	Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Pepper, Potato, Soybean, Raspberry, Squash, Strawberry, Tomato	MobileViT	PlantVillage	FL	PlantVillage classes FedLossAvg
Shrivastava et al. (2024)	Apple, Cron, Potato, Tomato, Grape, Peach, Strawberry, Blueberry	ViT	PlantVillage	FL	7 classes
Hari and Singh (2025)	Potato, Strawberry, Tomato, Banana, Coconut and Grapes	Parallel Multiscale CNN	2 datasets	FL	10-13 classes Proposed FDL-IWT
Yebasse, Shimelis, Warku, Ko, and Cheoi (2021)	Coffee	DenseNet121	RoCoLe	Centralized XAI	7 classes Grad-CAM, Grad-CAM + +, Score-CAM
Wang et al. (2021b)	Apple	CA-EfficientNet	Self-acquired	Centralized XAI	8 classes Coordination Attention
Mehedi et al. (2022)	Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Pepper, Potato, Soybean, Raspberry, Squash, Strawberry, Tomato	EfficientNetV2L, MobileNetV2, ResNet152V2	PlantVillage	Centralized XAI	PlantVillage classes LIME
Kiriella, Fernando, Sumathipala, and Udayakumara (2023)	Tomato	EfficientNetB0	PlantVillage	Centralized XAI	10 classes Score-CAM, LIME
Du, Li, and Qin (2024)	Apple, Cherry, Grape, Orange, Peach, Strawberry	FPEM-ResNet34	PlantVillage	Centralized XAI	10 classes ECA
Oad et al. (2024)	Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Pepper, Potato, Soybean, Raspberry, Squash, Strawberry, Tomato	VGG16, VGG19, ResNet101 V2, Inception V3	PlantVillage	Centralized XAI	PlantVillage classes LIME
Karim et al. (2024)	Grape	Customized MobileNetV3Large	Grapevine Disease Dataset	Centralized XAI	4 classes Grad-CAM

Table 1
(continued)

Work	Applied crops	Model	Dataset	Focus	Details
Nigar, Faisal, Umer, Oki, and Lukose (2024)	Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Pepper, Potato, Soybean, Raspberry, Squash, Strawberry, Tomato	CNN, MobileNetV2, EfficientNetB0, ResNet-50	New Plant Disease	Centralized XAI	PlantVillage classes LIME
Fathima and Booba (2024)	Apple, Potato, Tomato	MobileNet	PlantVillage	Centralized XAI	11 classes Grad-CAM
Patil, Pamali, Devagiri, Sushma, and Mirje (2024)	Corn, Potato, Rice, Wheat	Not specified	New Bangladeshi Crop Disease	Centralized XAI	14 classes LIME
Sarkar, Lifat, Hasan, and Hassan (2024)	Cotton	ResNet101V2	Cotton Leaf Dataset	Centralized XAI	2 classes Grad-CAM
Tang, Yang, Li, and Qi (2020)	Grape	ShuffleNet V1 and V2	PlantVillage	Centralized Quantization	3 classes Customized model
Wang et al. (2021a)	Grape	ECA-SNet	Self-acquired + not specified public dataset	Centralized XAI Quantization	6 classes ECA Ablation
Xiao, Shi, Zhu, Xiong, and Wu (2023)	Apple, Rice, Corn, Maize	SE-VRNet	PlantVillage + SelfData	Centralized Quantization	PlantVillage classes Customized model
Rakib, Rahman, Razi, and Hasan (2024)	Cherry, Corn, Peach, Potato, Strawberry, Tomato	Customized CNN	PlantVillage	Centralized Quantization	9 classes int8
Ji, Bao, Chen, and Wei (2024)	Corn	ICS-ResNet	PlantVillage, Crop Pest and Disease Dataset	Centralized XAI Quantization	5 classes Grad-CAM Ablation
Our work	Grape, Potato, Apple, Tomato, Peach, Cherry, Squash, Pepper bell, Blueberry, Corn, Raspberry, Strawberry, Soyabean, Sugarcane, Rice, Sweet potato	VGG16 + ECA	12 datasets	FL	10 classes FedAvg, FedProx FedBN, FedDF, SCAFFOLD
				XAI Quantization	ECA Ablation and Pruning

Punjab, India, into five severity levels. The model achieved 98.86 % accuracy for binary classification and 98.6 % for multi-class classification, demonstrating its effectiveness on the collected dataset. An automated system for crop disease detection is introduced in Jung et al. (2023). It used a stepwise classification model with images of diseased-healthy plant pairs and a CNN algorithm utilizing five pre-trained models by achieving an accuracy of 97.09 % on test. A fuzzy control algorithm and DL for automating greenhouse navigation and disease classification in tomato plants is adopted in Mac et al. (2024). A Deep Convolutional Generative Adversarial Network generated synthetic diseased leaf images, boosting ResNet-152's accuracy to 99.69 % on the augmented dataset, compared to 97.07 % on the original. A CNN and a VGG16-based Deep Convolutional Neural Network (DCNN) classifier have been employed in Prasad et al. (2024) to classify grape leaf diseases, leveraging a public dataset. The DCNN model incorporated additional layers to enhance performance. It achieves 99.18 % training accuracy and 99.06 % test accuracy, with systematic evaluation of metrics like F1-score. CNN-based model for detecting tea foliage diseases is proposed in Rahman et al. (2024), using a dataset of 3330 images from Sylhet, Bangladesh. The model classified leaves affected by red rust, brown blight, grey blight, and healthy conditions, achieving 96.65 % accuracy, validated through laboratory tests. Transfer learning with CNNs is employed in Kini et al. (2024) to predict diseases in black pepper leaves, using a field-collected, expert-annotated dataset. It classified anthracnose, slow wilt, early-stage phytophthora, phytophthora, and yellowing. The ResNet18 model outperformed others, achieving 99.67 % accuracy with a 0.001 learning rate. This study explored the use of digital image processing and DL to identify diseases in potato leaves, specifically early blight and late blight, alongside healthy leaves.

2.2. FL in crop disease detection

FL has emerged as a powerful approach for crop disease detection, enabling collaborative model training while preserving data privacy. Various studies have evaluated the performance of DL models in this

context, highlighting their effectiveness across different datasets and client distributions. An FL approach with a CNN was proposed for classifying jute leaf disease severity (Vats et al., 2024), using the Federated Averaging (FedAvg) (McMahan, Moore, Ramage, Hampson, & Arcas, 2017) algorithm to merge local updates while preserving data privacy. Evaluation yielded Macro and Weighted Average scores between 73.41–90.72 % and 73.12–90.73 %, showcasing strong performance. In Joshi et al. (2024), FL with five participants (xc_1 to xc_5) ensured data privacy, with model accuracy improving over time. Client xc_5 achieved 95.71 %, while xc_1 started at 88.76 %. Weighted averages ranged from 89.15 % (xc_1) to 95.72 % (xc_5), effectively handling data imbalances. An FL approach with CNNs for broccoli leaf disease classification is presented in Suryavanshi et al. (2024a), where five clients contributed data for five disease classes. The model achieved an average accuracy of 94.02 %, recall of 94.54 %, and F1-Score of 96.15 %, demonstrating its effectiveness in broccoli disease classification. FL with a CNN has been used in Upadhyay et al. (2024) for aloe vera leaf disease classification, where disease severity is categorized based on datasets from four clients. Client uj_4 achieved the best results with macro, weighted, and micro averages of 92.89 %, 92 %, and 92 %, respectively. A federated approach for Leaf Disease Detection (LDD) is proposed by Hari et al. (2024), using the PlantVillage dataset. The research proposed a lightweight Hierarchical CNN (H-CNN) with only 0.09 million parameters and a 0.35 MB model size. Experimental results showed that the proposed approach achieved 93 % accuracy, outperforming FedAdam (86.8 %) and FedAvg (87.7 %). An application of FL for wheat leaf disease classification is proposed in Fahim-Ul-Islam et al. (2024). Advanced vision transformer (ViT) models, CoAtNets and Swin Transformer V2, combined with a reinforced linear attention mechanism (LA) and weight pruning, enhanced feature representation while reducing inference time (624.249 ms and 644.899 ms) on low-power devices. The proposed pruned model (32M parameters) surpassed traditional transfer learning models, including ConvNeXt and EfficientNetV2, achieving up to 99 % accuracy, 98 % precision, 98 % recall, and 95 % F1-score, demonstrating FL's efficiency in wheat disease detection. In Antico et al. (2024), the application of FL for

maize LDD is explored. Five CNNs were evaluated under a distributed training paradigm, analyzing classification performance, training time, network traffic, and model complexity. Results demonstrated that FL effectively enhances data privacy in heterogeneous environments. Among the tested models, VGG-11 achieved the highest accuracy at 97.29%, followed by AlexNet at 96.87%, highlighting the potential of FL in agricultural disease classification. A resource-efficient FL approach for rice leaf disease classification has been employed in Aggarwal et al. (2024). It utilized federated transfer learning and federated feature extraction, evaluating their accuracy, loss, precision, recall, AUC, and resource usage (GPU memory, CPU/GPU utilization). The dataset consists of 5932 images of four rice leaf diseases: bacterial leaf blight, blast, brown spot, and tungro. EfficientNetB3, achieving 99% accuracy, is selected as the base model. The study finds that federated feature extraction reduces GPU memory and process usage, making it more suitable for IoT edge devices requiring lightweight execution. FL and CNNs have been employed in Suryavanshi et al. (2024a) to diagnose cauliflower leaf diseases, with five client populations (px_1 to px_5) using locally distributed data across five disease genera. The model demonstrated improved generalization, achieving macro averages from 91.83% (px_1) to 97.42% (px_5). A FL framework utilizing a ViT-based model and an enhanced aggregation method, FedLossAvgIn, is proposed in Singh et al. (2024). The research involved four clients, each handling distinct subsets of the Augmented PlantVillage dataset, to evaluate model performance under decentralized conditions. Both standard FedAvg and the proposed FedLossAvg were employed to compare their effectiveness in handling diverse data distributions. Experimental findings showed that the global model trained with FedLossAvg achieved high performance, reaching an F1-score of 95.62% and an accuracy of 99.89% on PlantVillage, demonstrating its capability to manage large and heterogeneous datasets effectively. In Shrivastava et al. (2024), it's proposed a privacy-preserving crop disease classification framework using FL, where individual farms classify their data locally with a ViT model and share only model weights with a central server. The ViT-based approach achieved an initial accuracy of 92.0% in disease detection, which improved to 97.0% when trained across 20 clients. A comparative analysis with other architectures, such as VGG-16 and VGG-19, demonstrated that ViT outperformed them, achieving higher accuracy (97% vs. 89.56% and 95.25%, respectively). Additionally, the model attained an AUC score of 0.98, highlighting its reliability in classification tasks. The study further evaluates model performance using F1-score, recall, and AUC metrics. An Intelligent Weight Transferring (IWT) method, which enhances model convergence by adjusting weight coefficients based on local model knowledge, is proposed in Hari and Singh (2025). The paper introduced the Parallel MultiScale CNN architecture (PMACNN) which is based on attention mechanism. Experimental results on two public datasets demonstrated that the proposed approach achieved 97.5% accuracy, surpassing seven state-of-the-art methods. Although FL has been widely adopted in crop disease classification tasks, most studies have utilized a single type of dataset, often without addressing the challenges posed by non-IID data. Additionally, these approaches typically focus on a limited number of disease classes, which may not fully represent the diversity of real-world agricultural scenarios. This highlights a gap in the current literature, where the generalizability and robustness of FL models across more complex, heterogeneous datasets could be further improved.

2.3. Explainability in crop disease detection

In the context of LDD, leveraging discriminative features within disease-affected regions became a key focus, as this approach was expected to enhance performance metrics, which was indeed confirmed. In some cases, the contours and colors of the leaves may appear similar or even identical, making it essential to focus more closely on the disease-affected patches on the leaf surface. The integration of channel attention mechanisms into CNNs has played a pivotal role in

improving model performance. A method for visualizing coffee diseases is proposed in Yebasse et al. (2021), using Grad-CAM, Grad-CAM++, and Score-CAM to understand model learning. Visualization improved classification, with a guided approach achieving 98% accuracy on the Robusta coffee leaf dataset, surpassing the naïve method's 77%. In Wang et al. (2021b), the introduction of the CAB attention module in CA-ENet enhanced its ability to focus on lesion areas. Compared to ResNet-152, DenseNet-264, and ResNeXt-101, CA-ENet accurately identified lesions in both complex and simple backgrounds, avoiding positional errors that compromise robustness and accuracy in other models. A framework detecting 38 leaf diseases across 14 plant types is proposed in Mehedi et al. (2022), comparing three pre-trained models. EfficientNetV2L outperformed others with 99.63% accuracy. Additionally, the LIME XAI technique was applied to interpret the EfficientNetV2L model's predictions. A tomato disease classification model leveraging EfficientNetB0 was implemented in Kiriella et al. (2023), using pre-trained weights of ImageNet and generated explanations for the predictions. The analysis revealed challenges in accurately locating disease symptoms when they covered the entire plant leaf. The ResNet34-based model in Du et al. (2024) improved classification by incorporating the ECA mechanism and meta-Acon activation function, capturing disease-specific features and reducing errors. The use of the Feature Pyramid Network enhanced feature fusion. The model achieved 98.46% accuracy on a Mosaic-augmented dataset. In Oad et al. (2024), an ensemble classifier combining VGG16, VGG19, ResNet101 V2, and Inception V3 detected plant diseases, achieving over 90% accuracy. The model used LIME for explanation, providing transparency by highlighting key pixels influencing predictions. A modified MobileNetV3Large model is adopted in Karim et al. (2024) for real-time grape leaf disease monitoring on an edge device. The model achieved 99.66% train and 99.42% test accuracy, with precision, recall, and F1 scores around 99.42%. Deployed on an Nvidia Jetson Nano, it predicted from both saved and real-time data. Grad-CAM visualization highlighted key image areas influencing CNN decisions. An XAI-based plant disease classification system is proposed in Nigar et al. (2024) to identify 38 distinct plant diseases with accuracy, precision, and recall of 99.69%, 98.27%, and 98.26%, respectively. Using LIME, the system generated visual explanations aligned with prior knowledge and best practices in interpretability. MobileNet, a lightweight CNN, is combined with Local Binary Pattern (LBP) in Fathima and Booba (2024) to enhance feature extraction for disease detection. LBP improved the model's ability to capture texture details, while Grad-CAM visualization highlights critical image regions influencing predictions, aiding validation and transparency. The approach achieved 96% accuracy, 90% precision, 89% recall, and 89% F1 score on a comprehensive dataset. Visualization techniques have been employed in Patil et al. (2024) to analyze model decision-making processes. The XAI visualization method achieved superior performance with 89.75% accuracy, outperforming both the heat map model and the LIME explanation method in disease identification. ResNets with Grad-CAM XAI heatmaps are analyzed in Sarkar et al. (2024) for cotton LDD. Among six ResNet models, ResNet101V2 performed best (96.9%) due to its depth and architecture. Grad-CAM enhanced transparency and offers insights into the model's decisions. These works emphasize the importance of attention mechanisms in enhancing the performance of LLD models, enabling more accurate and robust feature extraction. At the same time, the integration of explainability techniques plays a crucial role in improving model interpretability, helping to validate predictions in automated plant disease diagnosis.

2.4. Model quantization

Several studies have focused on optimizing DL models for LDD by reducing their complexity, often with only a slight loss in accuracy. Techniques such as quantization, ablation, and lightweight architectures have been widely adopted to enable deployment on resource-constrained devices. Below, we discuss some of the most relevant

approaches in the literature that leverage these strategies. The lightweight CNN model for diagnosing grape diseases, proposed in Tang et al. (2020), utilized channel-wise attention, ShuffleNet, and SE blocks. Validated on 4062 grape leaf images, the model achieved 99.14% accuracy and reduced the model size from 227.5 MB to 4.2 MB, making it suitable for mobile devices. The ECA-SNet model, proposed in Wang et al. (2021a), utilized a cross-channel interactive attention mechanism for efficient leaf disease recognition. With 6867 images of five common leaf diseases, image augmentation techniques were used for training. The ShuffleNet-v2 backbone is enhanced with a channel attention strategy for improved fine-grained lesion extraction. ECA-SNet achieved 98.86% accuracy, outperforming ShuffleNet-v2 with a 3.66% improvement, while using only 24.6% of the parameters and 37.4 M FLOPs. A new lightweight model, SE-VRNet, was proposed in Xiao et al. (2023) to improve the region of interest and lesion extraction. The model combined a deep variant residual network (VRNet) with a squeeze-and-excitation (SE) module and attention mechanism to improve feature extraction for dispersed leaf diseases. It achieved a top-1 accuracy of 99.73%, and a top-3 accuracy of 99.98% on NewData, and top-1 accuracy of 95.71%, and a top-3 accuracy of 99.89% on SelfData. Q-CNN has been proposed in Rakib et al. (2024) for recognizing 9 plant diseases using a computationally efficient model deployed on the ESP32-CAM IoT device. By applying model quantization, the approach reduced memory overhead while maintaining optimal performance. The model, with int8 quantization, achieved a size of just 28 KB, making it suitable for low-computation edge devices. ICS-ResNet, a lightweight CNN based on ResNet50, is employed in Ji et al. (2024). The network integrated improved channel and spatial attention modules, a depth-separable residual structure, and cosine annealing to enhance performance. The proposed model achieved 98.87% accuracy, outperforming eight popular networks by up to 5.03%, while reducing parameters by 69.21% and computations by 54.88%.

Despite the significant advancements in FL-based plant disease detection, existing studies often rely on limited datasets, small-scale data, and simplified FL scenarios that do not fully capture real-world challenges. In contrast, our work introduces a more comprehensive approach by utilizing 12 diverse datasets, each representing different scenarios, including controlled laboratory settings with uniform backgrounds and real-life conditions where leaves appear in natural environments. Overall, the 12 datasets comprise 16 various plants, 9 disease classes and one healthy class. This ensures broader generalization and robustness. Moreover, we design an FL framework that closely mimics real-world settings, with each dataset assigned to a single client, effectively addressing data heterogeneity and distribution variability. This realistic setup enhances the applicability of FL for large-scale agricultural disease detection. Following the classification, XAI techniques allow us to highlight the areas where the model has identified the disease. Through a recommendation process that leverages both online resources and a personalized chatbot experience, the system provides customized recommendations regarding the plant, the detected disease, and the appropriate treatment.

3. Background

3.1. Federated learning

FL is an ML paradigm where multiple clients or workers collaboratively train a shared global model while keeping their data decentralized. This approach enables the development of a more robust and efficient model that can be distributed back to the participating clients. FL consists of a central server and a set of k clients, denoted as $\{C_1, \dots, C_k\}$. Each client C_i , $1 \leq i \leq k$, maintains its own private dataset D_i , ensuring that sensitive data remains confined to the client's local device, a critical feature for preserving privacy. Each client begins by receiving the initial global model $W^{(0)}$ from the central server. Using this model, clients perform local training on their individual datasets. Once local

training is completed, the clients send updates containing their trained model parameters back to the central server. Throughout this process, all training data remains private and securely stored on the client's devices, ensuring that no raw data is ever shared. The central server then aggregates the parameters received from all clients to create an updated global model.

To refine the global model, the server aggregates updates from a subset of clients S_t , selected according to a specific strategy. A widely adopted aggregation approach, FedAvg (McMahan et al., 2017), weights each client's contribution proportionally to the size of its local dataset. At the t -th communication round, the global model $W^{(t)}$ is updated as follows:

$$W^{(t)} = \sum_{i \in S_t} \frac{n_i}{n} W_i^{(t)} \quad (1)$$

where $i \in S_t$ denotes the clients selected for aggregation in round t , n_i represents the size of the local dataset held by client C_i , n is the total size of data from all selected clients, $W_i^{(t)}$ denotes the locally updated model parameters of client i at the t -th communication round, and $W^{(t)}$ is the aggregated global model at round t . The updated global model is then distributed back to the clients, where it is evaluated using each client's test dataset. The training process proceeds iteratively over multiple communication rounds between the clients and the central server until a specified performance criterion, such as the desired accuracy or a predefined number of rounds, is met. At the end of the training, the final global model is shared with all clients, enabling them to perform inferences with the improved and consolidated model.

To further enhance the robustness and efficiency of FL, several extensions of FedAvg have been developed to address challenges such as statistical heterogeneity, client-drift, and model aggregation constraints. Among these, FedProx, SCAFFOLD, FedBN, and FedDF introduce key modifications to improve stability, convergence, and adaptability in non-IID settings. FedProx (Li et al., 2020b) is an extension and re-parametrization of FedAvg, specifically designed to address the challenges posed by data and system heterogeneity in FL. Unlike FedAvg, which struggles with statistical variations across clients and the presence of stragglers, FedProx introduces a proximal term in the optimization objective to enhance stability and convergence. This term serves as a regulatory mechanism, reducing the negative impact of partial updates from straggling clients and ensuring a more consistent learning process. By incorporating this adjustment, FedProx effectively mitigates the trade-offs between system-level and statistical heterogeneity, leading to improved performance in FL settings. SCAFFOLD (Karimireddy et al., 2020) (Stochastic Controlled Averaging for Federated Learning) is an FL optimization algorithm designed to address the issue of client-drift, which arises due to data heterogeneity across different clients. To mitigate this problem, SCAFFOLD introduces control variates, a variance reduction technique, to correct local updates and align them more effectively with the global model. The algorithm estimates the update direction of the central server model and compares it with the update direction of each client. The difference between these two values represents an estimation of the client drift, which is then used to adjust the local updates. FedBN (Li, Jiang, Zhang, Kamp, & Dou, 2021) is a learning strategy designed for non-IID scenarios. Like FedAvg, it updates local models and performs model averaging. However, FedBN specifically applies to models with Batch Normalization (BN) layers and excludes BN-related parameters from the aggregation process. FedBN operates independently of communication and aggregation strategies, making it highly flexible. In practice, it can be seamlessly integrated with various optimization algorithms, communication protocols, and aggregation methods. Lastly, FedDF (Lin, Kong, Stich, & Jaggi, 2020) (federated learning using ensemble distillation for robust model fusion) introduces ensemble distillation for model fusion, on the server side, training the central classifier using unlabeled data based on client model outputs. This knowledge distillation approach maintains the same level of privacy protection and computational efficiency as standard FL

algorithms while enabling flexible aggregation of heterogeneous client models, which may vary in size, numerical precision, or architecture.

3.2. VGG Model for classification

VGG16 (Simonyan & Zisserman, 2014) is a DL model belonging to the family of CNNs which is recognized as one of the most influential architectures in the field of computer vision. The network is built with 13 convolutional layers, 5 pooling layers, and 3 fully connected layers, amounting to a total of 21 layers. However, only 16 of these contain learnable parameters, from which the name "VGG16". The model takes as input RGB images of size 224×224 pixels and outputs a vector of dimension 1000, representing the probabilities associated with each class.

The pooling layers employ 2×2 filters with a stride of 2 to progressively reduce the spatial dimensions of feature maps, while the convolutional layers utilize 3×3 filters with a stride of 1. These filters are consistently paired with padding to maintain the spatial resolution of the input across layers. The fully connected portion of the architecture consists of three dense layers: the first two are composed of 4096 neurons each, while the final layer has 1000 units to represent class probabilities. Softmax activation is applied in the output layer to facilitate multi-class classification, while the ReLU activation function is used throughout the hidden layers to introduce non-linearity, improving the network's ability to model complex patterns.

3.3. Explainability with attention layers

XAI enhances model transparency by providing insights into decision-making processes, making DL models more interpretable. In the context of feature extraction and attention mechanisms, various techniques have been developed to improve model efficiency and interpretability. The ECA mechanism (Wang et al., 2020b) is an improvement over the Squeeze-and-Excitation Network (SENet) (Hu, Shen, & Sun, 2018). A SENet is constructed by stacking multiple SE blocks. In the SE block's core structure, the input feature maps $\mathbf{U} \in \mathbb{R}^{H \times W \times C}$ first undergo a squeeze operation. This step aggregates spatial information across $H \times W$ dimensions creating a channel descriptor that captures the global distribution of feature responses across channels. This descriptor allows the network's lower layers to utilize global receptive field information. Next comes the excitation operation, where a self-gating mechanism learns activation weights for each channel based on channel dependencies. These weights dynamically adjust the importance of each channel. Finally, the feature maps \mathbf{U} are reweighted according to these activations. Moreover, SENet utilizes a fully connected layer for feature compression, which can negatively affect the network's predictive accuracy. Additionally, this process introduces greater model complexity and increases computational overhead. In contrast, ECANet refines SENet by integrating the ECA module, which focuses on enhancing

local cross-channel interactions. Unlike SENet, ECANet employs global average pooling without reducing the dimensionality of the channels. The ECA module is specifically designed to extract local cross-channel interaction information by considering each channel and its k nearest neighboring channels. A structural diagram of the ECA module is depicted in Fig. 1.

The initial step involves aggregating the features through global average pooling, which is used to capture the global information of each channel. The global average pooling operation can be expressed using the following formula:

$$y = \frac{1}{H \times W} \sum_a^H \sum_b^W x_i(a, b) \quad (2)$$

Here, x_i denotes the i -th feature map with dimensions $H \times W$, and y represents the resulting global features. In the next step, the number of cross-channels k is adaptively calculated based on the total number of channels C . The adaptive function is defined as follows:

$$k = \psi(C) = \left\lceil \frac{\log_2(C)}{\gamma} + \frac{b}{\gamma} \right\rceil_{odd} \quad (3)$$

where $\lceil t \rceil_{odd}$ refers to the nearest odd integer to t , C is the total number of channels, and b and γ are constants, with $b = 1$ and $\gamma = 2$.

$$w = \text{sigmoid}(C_1 D_k(y)) \quad (4)$$

Let $C_1 D$ be a 1D convolution operation, y the result of the global average pooling and k the size of the convolution kernel, then w represents the channel weights. Finally, the channel attention-enhanced features are obtained by performing a dot product between the original input features and the channel weights.

Moreover, in traditional CNN architectures, all features are processed with equal importance, as there is no inherent mechanism to differentiate the significance of individual features. The ECA module addresses this limitation by enhancing the model's ability to represent features. This mechanism assigns greater weights to specific regions associated with disease-affected areas.

3.4. Quantization for lightweight models

To enable the deployment of the model on resource-constrained edge devices, it is essential to reduce its computational and memory requirements. This is achieved by applying model optimization techniques such as ablation and pruning. These methods aim to significantly reduce the model's memory while only slightly compromising its performance, ensuring efficient use of hardware resources and facilitating real-time disease classification and visualization on devices with limited computational power.

Ablation refers to the process of systematically removing components or layers from a model's architecture to evaluate their impact

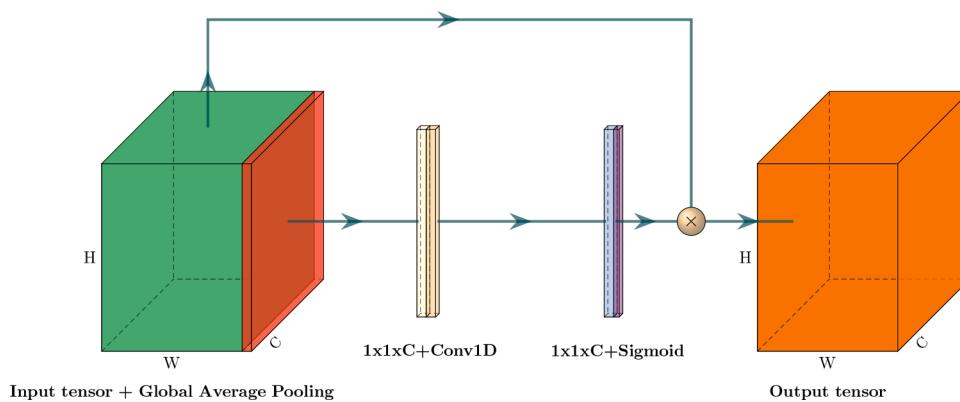


Fig. 1. ECA layers.

on overall performance. This technique helps identify redundant or less critical parts of the model, which can be excluded without significantly compromising its accuracy. By eliminating these components, the model's computational requirements and memory can be effectively reduced.

Pruning involves removing less important weights, filters, or channels from a neural network based on a defined importance criterion. For channel pruning, an importance score is calculated for each channel to identify which ones contribute least to the model's performance (Fang, Ma, Song, Mi, & Wang, 2023). The approach involves the following steps:

- **Importance criterion:** A fixed norm of grouped weights is used as the importance criterion to determine which channels to remove. This allowed identifying the least important parameters in the model for pruning. For example, in the case of L_2 norm, the Importance is:

$$\text{Importance}(W) = \|W\|_2 \quad (5)$$

where W represents the channel weights.

- **Pruner initialization:** A pruner is initialized using the model and the importance criterion, with specific layers, such as the final classifier, excluded from pruning.
- **Pruning operation:** The pruning ratio indicates the proportion of channels removed during the pruning process. If C_{original} represents the total number of channels before pruning and C_{pruned} is the number of channels retained after pruning, the pruning ratio can be expressed as:

$$\text{Pruning ratio} = 1 - \frac{C_{\text{pruned}}}{C_{\text{original}}} \quad (6)$$

- **Evaluation:** The model's performance, in terms of multiply-accumulate operations (MACs) and the number of parameters, is evaluated before and after pruning. This step is crucial in quantifying the computational benefits gained from pruning.

3.5. Large language models for personalized recommendations

A recommender system is an ML-based approach that provides personalized suggestions to users by analyzing their previous interactions, preferences, and behavioral patterns. It falls under the category of information filtering systems, employing algorithms to suggest relevant items based on user interests and activity. In traditional recommender systems, techniques such as clustering, content-based filtering, and collaborative filtering are commonly used to connect users with relevant content or with other similar users. These methods analyze user interactions, preferences, and similarities to generate recommendations.

An advanced approach to improving recommender systems involves leveraging Large Language Models (LLMs). LLMs have the potential to transform the way recommendations are generated, especially when dealing with unstructured textual data. By understanding context, extracting meaningful insights, and processing vast amounts of information, LLM-based recommender systems can provide accurate and personalized suggestions, enhancing user experience and decision-making processes. LLMs process natural language by transforming raw text into numerical representations that the model can understand. This begins with tokenization, where text is broken into smaller units, such as words or subwords. These tokens are then converted into dense numerical vectors which are fed into a model based on Transformer architecture.

Notable examples of LLMs include GPT-3, introduced by Brown et al. (2020), which marked a significant advancement in scale and capability with 175 billion parameters, enabling high-quality text generation with minimal fine-tuning. Building on this success, OpenAI released GPT-4 in 2023, further improving performance and expanding the potential applications of LLMs (Achiam et al., 2023).

4. Framework overview

Our work proposes AGRIFOLD, a comprehensive framework (Fig. 2) for plant disease detection and appropriate treatment recommendations. AGRIFOLD consists of two main phases: the first involves clients participating in an FL process to collaboratively train a global model for LDD and classification. Specifically, after the training, AGRIFOLD enables real-time, on-site analysis of plant leaf conditions using images captured with various devices. The second phase is the recommendation phase where clients receive treatment suggestions for the identified diseases through web scraping and interaction with a chatbot.

4.1. Leaf disease detection and classification

The supervised classification model is trained in an FL setting, utilizing datasets collected from multiple geographic regions. These datasets include images taken under varying lighting conditions, backgrounds, and environmental factors, ensuring robustness and generalization across different scenarios. FL allows individual users, such as farmers and researchers, to collaboratively train a global model while keeping their data private, preserving confidentiality and data security. Our model is based on the VGG16 architecture, enhanced with ECA layers, which improve feature extraction and classification performance. This design enables precise identification of plant diseases across multiple crop types. Furthermore, following the classification predicted, if our model detects a disease, integration with the website information (see 4.2) and the Google Gemini API (see 4.3) allows users to obtain detailed information about the identified disease including symptoms, causes, and recommended treatments.

4.2. Web scraping

To enhance the decision-making process and provide actionable insights, we collected relevant treatment strategies through web scraping from selected websites. For general plant disease knowledge, Britannica (????) provides comprehensive descriptions of diseases such as late blight, powdery mildew, and rust, covering their causes, symptoms, and impact. Similarly, Wikipedia (2025) offers a broad overview of fungal diseases. The University of California Integrated Pest Management platform (UCANR , 2025) features extensive guides on diseases like early and late blight, bacterial spot, black rot, mosaic, yellow, and powdery mildew. The Michigan State University (CARN , 2025) offers information about rust, bacterial spot, black rot, and powdery mildew. Moreover, Bayer Crop Science (2025), Koppert (2025), and Plantix (2025) are just a few of the many platforms and websites from which we have extracted information. By aggregating data from these diverse sources, AGRIFOLD ensures that users receive up-to-date, practical disease management recommendations from the web, tailored to their specific plant health concerns. Furthermore, the database containing the sources can be continuously updated and modified to incorporate new references, including those suggested by clients, thereby broadening the scope of diseases for which the system can provide information.

4.3. Chatbot

In AGRIFOLD, a dynamic recommendation system is implemented to suggest tailored treatment solutions for plant leaf diseases. After classifying the uploaded or user-captured photos of leaf diseases directly on-site, the system triggers an interactive chatbot dialogue powered by the Google Gemini API to provide the corresponding treatment information. Specifically, a general prompt template is used to initiate the conversation, such as: "My plant is {plant_name}, it is affected by {disease_class}, how can I treat it?" The dialogue then evolves according to the user's input, allowing the chatbot to generate personalized treatment recommendations and care plans based on the specific circumstances provided.

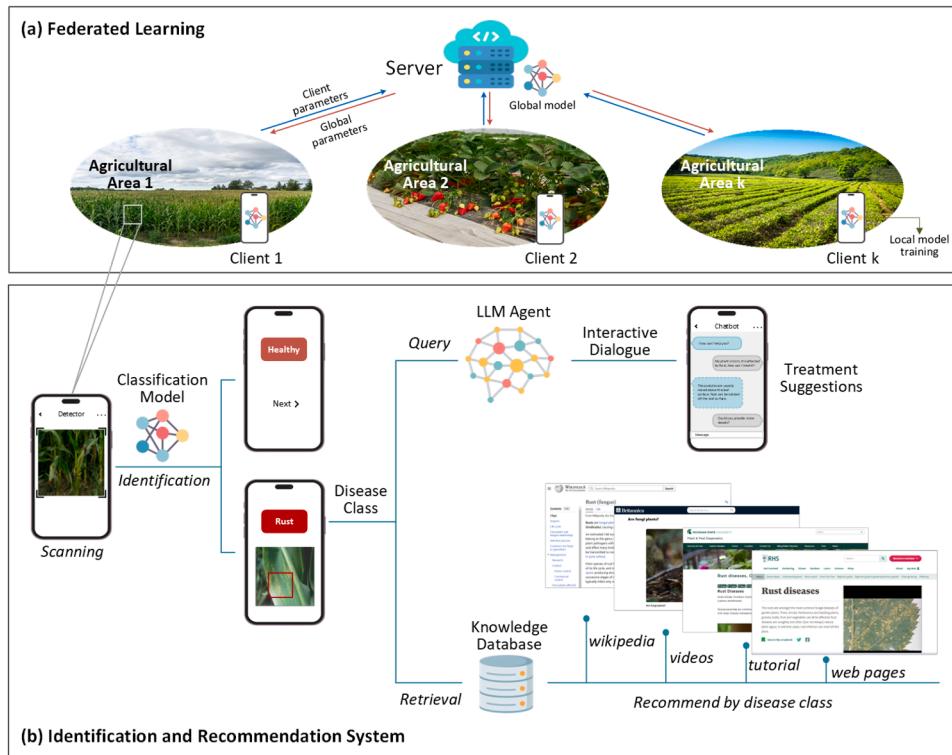


Fig. 2. AGRIFOLD framework. **a)** Clients train the global model locally and send updates to the server, which aggregates them to create an updated global model, and then redistributes it. **b)** Each client can utilize the trained classification model to detect plant diseases by analyzing leaf images, including real-time images captured directly in the field. If a disease is detected, AGRIFOLD provides a collection of resources for further information about the plant, the disease, and possible treatments. Additionally, a chatbot is available to offer a personalized experience and assist in identifying the most suitable treatment for the specific case.

5. Experimental setup

Here, we present our experimental setup, detailing the datasets used, the chosen FL configuration, and the hardware infrastructure employed to simulate an FL environment. Additionally, we describe the framework used to manage communication between the server and clients, along with the selected hyperparameters. Finally, we discuss the quantization study conducted on the model, including both ablation and pruning experiments.

5.1. Dataset description

Image datasets used in this study are from various free available sources (see Section Data and Resources). To create a realistic simulation of the FL scenario, we selected datasets representing various plant diseases from different regions worldwide. With this approach, each client in the federation has plants with various diseases, collected with different devices. Accordingly, we incorporated the following datasets and diseases (Fig. 3):

- Apple Tree Leaf Disease Segmentation Dataset ([Feng & Chao, 2022](#)): apple leaf disease images, collected in four different zones in Northwest China, using a Glory V10 mobile phone. Images with different degrees of disease were taken in the laboratory and in real cultivation fields, under different weather conditions and at different times of the day. Selected classes from this dataset are brown spot, rust, and healthy.
- Corn Leaf Disease ([Indika & Saputra, 2023](#)): corn leaf images collected in Indonesia. The image data was captured using a mobile phone camera with a resolution of 16 MegaPixels, placing white paper behind the corn leaves. The images were taken 5 times for each corn leaf, during daytime (12 PM - 2 PM WIB) to ensure good

lighting conditions. There are 4 classes in the data: healthy, leaf blight, leaf spot, and leaf rust. Each class contains 1000 images. We selected healthy and leaf rust classes.

- Dataset of Tomato Leaves ([Huang, Chang, & Ya-Han, 2020](#)): this dataset is divided into two datasets for tomato leaf images according to different image sources. The tomato leaf images of the first dataset are selected from the PlantVillage dataset. Each image is composed of a single leaf and a single background, for a total of 14531 images. The second dataset is composed of images of Taiwanese tomato leaves, with six classes (five disease classes and one health). It consists of a single leaf, multiple leaves, a single background, and a complex background. From the second dataset, we selected bacterial spot, early blight, late blight, powdery mildew, and the healthy classes.
- Indigenous Dataset for Apple Leaf Disease Detection and Classification ([Yatoo & Sharma, 2024](#)): apple leaf images collected in the Jammu and Kashmir region. The dataset contains 7569 images belonging to healthy, alternaria, and mosaic classes. We selected only the healthy and mosaic classes.
- JMUBEN 3 ([Jepkoech, Mugo, Kenduiywo, & Too, 2023](#)): potato leaf images collected in Kenya using a digital camera and with the help of plant pathologists for easy disease identification. The dataset contains 1383 images affected by rust.
- PlantDoc ([Singh et al., 2020](#)): 2598 images across 13 plant species and up to 17 classes of diseases. We select the following diseases, which do not belong to all plant species: Bacterial spot, black rot, early blight, late blight, mosaic, powdery mildew, rust, yellow, and healthy.
- PlantVillage ([J, Pandian, & Gopal, 2019](#)): 61486 plant leaf and background images belonging to 39 different classes. We selected the following 8 classes: Bacterial spot, Black rot, early blight, late blight, mosaic, powdery mildew, yellow, and healthy.

Apple Tree Leaf Disease Segmentation Dataset			Corn Leaf Disease		
Healthy	Brown spot	Rust	Healthy	Rust	
					
Dataset of Tomato Leaves			Indigenous Dataset for Apple Leaf Disease Detection and Classification		
Bacterial spot	Healthy	Late blight	Powdery mildew	Healthy Mosaic	
					
PlantDoc					
Healthy					
					
Apple	Blueberry	Cherry	Grape	Peach	
					
Tomato	Grape	Potato	Potato	Squash	
PlantVillage					
Healthy					
					
Apple	Blueberry	Cherry	Corn	Grape	
					
Peach	Pepper bell	Tomato	Apple	Grape	
					
Healthy	Early blight	Late blight	Potato	Potato	
					
Potato Disease Leaf Dataset (PLD)			Potato Leaf (Healthy and Late)		
Healthy	Early blight	Late blight	Healthy	Late blight	
					
Potato	Potato	Potato	Potato	Potato	
Sugarcane Leaf Dataset			Sugarcane Leaf Images		
Healthy	Mosaic	Rust	Brown spot	Healthy	
					
Sugarcane	Sugarcane	Sugarcane	Sugarcane	Sugarcane	
					

Fig. 3. Example images from each dataset, illustrating the intra- and inter-dataset variability of plant species and disease classes.

- Potato Disease Leaf Dataset (PLD) (Rizwan, Saeed, & Rashid, 2021): 4062 images collected from the Central Punjab region of Pakistan. Disease classes in this dataset are early blight, late blight, and healthy.
- Potato Leaf (Healthy and Late Blight) (Tilahun, 2020): potato leaf images collected in Ethiopia from two plant pathologists, in various noisy environments, using a high-resolution digital camera and smartphone. Late blight class comprises 63 images and Healthy class comprises 363 images.
- Rice Leaf Disease Image (Sethy, Barpanda, Rath, & Behera, 2020): 5932 images encompassing four types of rice leaf diseases: bacterial blight, blast, brown spot, and tungro. We only select the brown spot class.
- Sugarcane Leaf Disease (Daphal & Koli, 2022): sugarcane leaf images captured with various smartphones in India. It contains 2569 images

including 5 disease classes: healthy, mosaic, redrot, rust, and yellow. We selected all classes except the redrot class.

- Sugarcane Leaf Images (Thite, Suryawanshi, Yogesh, Patil, & Kailas, 2023): 7134 high-resolution images of sugarcane leaves, categorized into 12 distinct classes, including 10 disease classes, a healthy leaves class, and a dried leaves class. These images were collected capturing different angles and stages of the disease. We selected brown spot, yellow, and healthy classes.

Hence, in the federated configuration, there are 12 datasets, each assigned to one client, and a total of 10 classes: healthy, bacterial spot, black rot, brown spot, early blight, late blight, mosaic, powdery mildew, rust and yellow. As reported in Fig. 4, each client has a different number of images, representing various diseases on leaves. The PlantDoc and PlantVillage datasets are the ones with the highest variety of diseases,

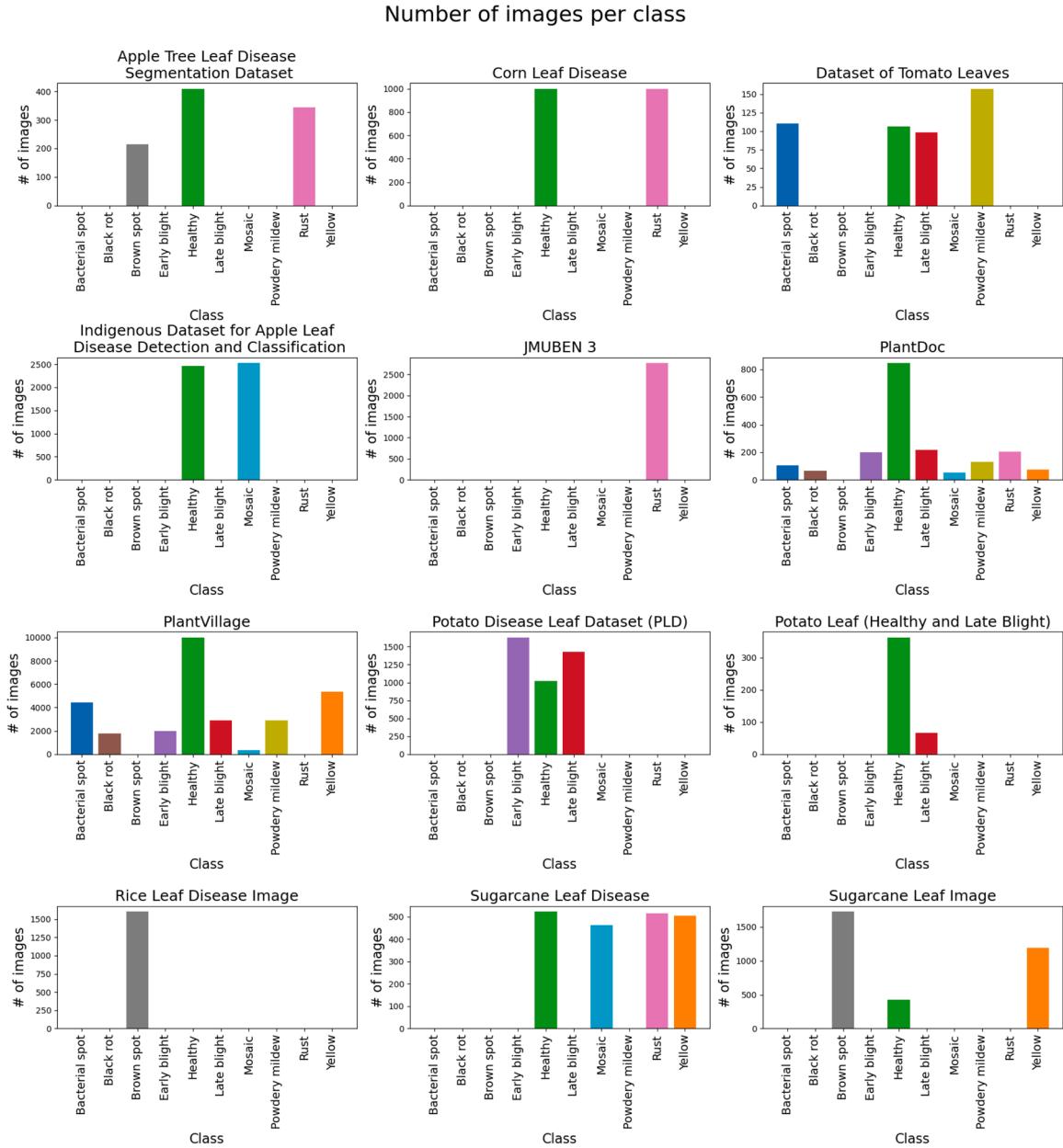


Fig. 4. Number of images per class, for each dataset. The plots highlight the significant variability in class representation across clients, confirming the presence of a highly non-IID data distribution scenario, which reflects realistic conditions in agricultural applications.

containing 8 and 7 disease classes, respectively, along with the healthy class. PlantVillage is also the dataset with the largest number of images. Some datasets, such as JMUBEN 3 and Rice Leaf Disease Image, contain only one class and do not include the “healthy” class. Potato Leaf (Healthy and Late Blight) and Dataset of Tomato Leaves are the datasets with the smallest number of images. This highlights how clients have varying amounts and types of data as well as different distributions, confirming that our setting is highly non-IID and effectively simulates a real-world scenario.

5.2. Preprocessing

To ensure uniformity and enhance the quality of input images, we applied a structured preprocessing pipeline. First, all images were read and resized to a fixed resolution of 224×224 pixels while maintaining the aspect ratio by adding white padding where necessary. This step ensured consistency across different datasets and acquisition conditions. Subsequently, pixel intensity values were normalized within the range

$[0, 1]$ using min-max normalization for each color channel (RGB) separately. The preprocessed images were then stored in a compressed format to optimize storage and retrieval efficiency. This preprocessing step aimed to standardize the dataset, reducing variations due to differences in image dimensions, acquisition devices, and lighting conditions.

5.3. Flower framework

To create an FL environment, we employed the Flower framework (Beutel & et al., 2022), which facilitates the coordination between the central server and multiple participating clients. Flower was chosen due to its flexibility in integrating new training algorithms and its adaptability to various client environments, such as mobile and wireless devices, making it particularly suitable for agricultural applications. Its modular design allowed us to easily implement and manage FL simulations using a variety of datasets and configurations.

However, the current version of Flower supports only synchronous aggregation, meaning that the central server must wait for all clients to

complete their local training before proceeding to the next communication round. This approach introduces waiting times for clients with smaller datasets, as they finish their training faster compared to those handling larger data volumes. While this synchronous strategy ensures that all client updates are included in the global model, it can lead to inefficiencies in scenarios with significant data imbalance. To mitigate this, we optimized local training settings, ensuring that clients with larger datasets efficiently utilized their computational resources while minimizing idle time for smaller clients.

5.4. Hardware infrastructure

The experiments were conducted on a multi-node server setup utilizing the Infrastructure for Big Data and Scientific Computing (I.B.I.S.CO) at the S.Co.P.E. Data Center, University of Naples Federico II (Barone et al., 2022). This infrastructure comprises 36 nodes, of which 32 are allocated for computation and 4 are designated for storage, providing the necessary resources to simulate an FL environment. Each computational node is powered by a DELL C4140 server, equipped with 4 NVIDIA Tesla V100 GPUs, resulting in a total of 128 GPUs across the entire system. The choice of this HPC system was driven by its ability to allocate dedicated GPUs to each client, ensuring parallel and efficient local training. Additionally, the system's high-bandwidth interconnections minimized communication delays between clients and the central server, a crucial factor in FL simulations. Its flexibility allowed the emulation of heterogeneous data distributions and non-IID conditions, closely mimicking real-world agricultural scenarios where data volumes and device capabilities vary significantly. This setup enabled large-scale simulations while maintaining computational efficiency and scalability.

5.5. Model and hyperparameters

In the implemented FL configuration, a distinct dataset was associated with each client, leading to a total of 12 clients participating in the training process. The server and clients are managed using the Flower framework. For experimentation, the 12 clients are each assigned to a different GPU. An additional GPU is used by the server, for a total of 13 GPUs utilized across various nodes of the I.B.I.S.CO server, depending on availability. This approach allows for the simulation of a real-world scenario where clients have different plants, affected by various diseases, and in varying quantities. The clients participate in FL to collaboratively train the same model and share knowledge among themselves. Our framework uses 5 aggregation strategies to combine the contributions from each client and update the global model. A total of 100

communication rounds are chosen. The training process adopts a stochastic gradient descent algorithm (SGD) as optimizer, while cross-entropy loss is employed as the criterion for model optimization. Moreover, each client locally trains the model for 5 epochs per round. The backbone of the shared model is the VGG16 architecture, and we employed its pretrained version on the ImageNet dataset. The choice of VGG16 was motivated by its proven effectiveness in various visual recognition tasks, particularly in the domain of plant and LDD. It has been widely adopted in prior studies, consistently demonstrating high accuracy and robust performance. This strong track record made it a reliable and well-suited choice for our FL framework. The model is organized into five blocks, where each block consists of several convolutional layers followed by ReLU activations. After each block of VGG16, an ECA module was inserted, as shown in Fig. 5, following findings from related works that have demonstrated the effectiveness of adding attention mechanisms, such as ECA, after each convolutional block to improve the model's feature extraction and overall performance. In the proposed DCNN classifier, the original 1000 classes of the ImageNet dataset are adapted to classify images into 10 categories of plant diseases affecting fruits, vegetables, and crops. In our training using the ECA module, we fixed all parameters except for the kernel size k , which was the only hyperparameter fine-tuned. The tuning was performed in the FL setting using the FedAvg strategy and a batch size of 128. Based on the comparable results obtained by changing the value of k , we selected $k = 3$.

To enhance the understanding and interpretability of our model's predictions in the multi-class classification task, we employed heatmaps. Heatmaps provide a visual representation of the regions in the image that the model considers most relevant for classification, enabling the identification of specific areas of the leaves that most influence the predictions.

5.6. Ablation study for optimizing model complexity

The ablation technique was applied exclusively to the final classifier of the VGG16 model, as its fully connected layers constituted a significant portion of the memory of the entire architecture. For the purposes of experimentation, three alternative configurations of the classifier were designed and evaluated.

As shown in Fig. 5, the classifier in the first configuration (a) was designed as a reduced architecture with the aim of balancing complexity and performance. It consisted of two fully connected layers: the first transformed the input from $512 \times 7 \times 7$ to 2048 dimensions, followed by a ReLU activation function and a dropout layer with a probability of

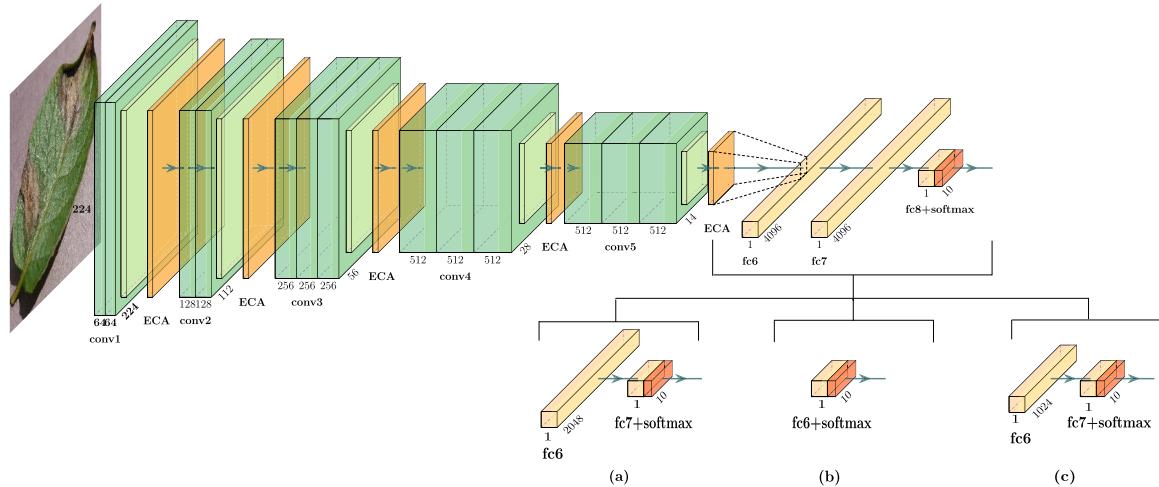


Fig. 5. VGG16 architecture with ECA layers and ablation configurations applied on classifier composed of: (a) two fully connected layers, (b) a single fully connected layer, (c) a fully connected layer, followed by a ReLU activation function, a 50% dropout and a linear layer.

50 %. The second layer reduced the dimensionality to match the number of output classes, effectively serving as the final prediction layer.

The second configuration (b), was the simplest and lightest among the tested setups. The classifier in this case consisted of a single fully connected layer that directly mapped the input features $512 \times 7 \times 7$ to the output classes. This minimalistic design greatly reduced the number of parameters and computational complexity, providing a baseline for assessing the impact of a highly simplified classifier.

The third configuration, labeled as (c), represented an intermediate design, striking a balance between the simplicity of (b) and the complexity of (a). In this setup, the input features $512 \times 7 \times 7$ was first transformed into a vector of size 1024 by a fully connected layer, followed by a ReLU activation function and a 50 % dropout. A second linear layer then reduced the dimensionality to match the number of classes.

5.7. Pruning

A structured pruning technique was applied to reduce model size, leading to a slight decrease in performance. The pruning process was based on channel importance, measured using the L_2 norm of grouped weights.

Pruning was performed at different levels using the formula in (6), with pruning ratios of 0.3, 0.5, 0.7, and 0.9. In particular, the experiments were conducted using FedAvg as the aggregation method, setting the batch size to 256. This resulted in model sizes of 356.27 MB, 246.53 MB, 141.82 MB, and 45.47 MB. The final classifier layer was excluded from pruning. The pruned models were then evaluated in terms of accuracy, multiply-accumulate operations (MACs), and parameter count to assess changes in computational complexity and model size.

6. Results and discussion

In this section, we present the results obtained from the FL framework using 12 datasets, each assigned to a client. Table 2 shows the association between datasets and clients. This setup brings with it all the challenges associated with non-IID data in FL, such as divergent local updates that can favor clients with larger datasets and lead to reduced accuracy for clients with rare diseases. Specifically, we compared the performance of five FL methods, including FedAvg, FedProx, SCAFFOLD, FedBN, and FedDF. When using FedProx, we set the coefficient $\mu = 0.9$ while the implementation of FedBN was achieved by integrating a BN layer after each convolutional layer within the ECA module. The accuracies and the weighted average metrics achieved after 100 rounds are reported, respectively, in Tables 3 and 4. It is worth noting that, due to memory constraints, all FL methods except FedAvg were executed using a batch size of 128. In contrast, FedAvg was the only method that could be successfully trained with a larger batch size of 256. Therefore, we report results for both batch sizes (128 and 256) to provide a comprehensive comparison. As observed from the reported results, SCAFFOLD consistently achieved the best performance across most clients

Table 2
Association between datasets and clients.

Client	Dataset
1	PlantVillage
2	Potato Leaf (Healthy and Late Blight)
3	Potato Disease Leaf Dataset
4	Dataset of Tomato Leaves
5	PlantDoc
6	Apple Tree Leaf Disease Segmentation Dataset
7	Corn Leaf Disease
8	Indigenous Dataset for Apple Leaf Disease Detection and Classification
9	JMUBEN 3
10	Rice Leaf Disease Image Samples
11	Sugarcane Leaf Image Dataset
12	Sugarcane Leaf Disease Dataset

Table 3
Accuracies of different FL techniques.

Client	Accuracy (%)					
	FedAvg (batch size = 256)	FedAvg (batch size = 128)	FedProx	SCAFFOLD	FedBN	FedDF
1	98.39	98.79	91.80	99.21	98.99	96.60
2	86.92	94.62	89.23	98.46	79.23	97.69
3	86.51	88.80	78.90	97.55	89.62	93.54
4	24.48	25.17	28.67	72.03	23.78	62.24
5	49.91	49.05	44.39	72.71	48.01	74.27
6	59.25	59.60	57.19	97.60	64.73	95.21
7	83.50	90.50	82.83	99.00	93.67	95.17
8	97.53	96.72	93.65	99.67	97.66	93.98
9	97.22	92.77	73.61	99.76	95.42	99.52
10	55.21	60.21	59.17	99.59	76.88	95.63
11	78.00	82.69	69.85	99.5	82.19	91.34
12	50.58	54.23	37.81	87.73	58.54	80.76

and metrics. This suggests its superior ability to handle the heterogeneity inherent in non-IID data distributions, likely due to its control variates mechanism, which effectively mitigates client drift and improves alignment between local and global model updates. In contrast, FedProx yielded the lowest overall performance, indicating its limited capacity to generalize across clients with distinct datasets. Globally, the accuracies are notably high, especially for datasets like PlantVillage, which contains a large number of images captured in good conditions with neutral backgrounds. Datasets such as PlantDoc, which include images with natural backgrounds, show lower accuracy. Among all datasets, the Dataset of Tomato Leaves exhibits the lowest performances. This can be attributed to multiple factors, including the presence of complex natural backgrounds, which introduce visual noise and make feature extraction more challenging for the model. Additionally, the dataset suffers from low image quality, with issues such as low resolution and different levels of zoom. These artifacts not only degrade the overall image clarity but also obscure or alter the visual characteristics of plant diseases, making it difficult for the model to correctly recognize disease patterns. As a result, the model may misinterpret or fail to detect key disease traits, leading to a significant drop in classification accuracy.

Fig. 6 illustrates examples of heatmaps for each dataset on a correctly classified test image, obtained by leveraging the weights and feature maps produced by the ECA layers. The overlay of the heatmap highlights the regions of the leaf exhibiting symptoms.

Moreover, we analyzed the variation in performance as the model size changed, using ablation and pruning techniques under the FedAvg setting, with batch size equal to 256. By reducing the final part of the model, we obtained smaller models with sizes equal to 264.48 MB, 161.68 MB, and 59.87 MB. Correspondingly, there is a lower GPU memory consumption during the training phase, as shown in Fig. 7a), when the model size decreases. The accuracies for each client, relative to the ablated models, are also reported in Fig. 7b). Clients 6, 7, 10, 11, and 12 achieve higher accuracy with the 246.53 MB model, indicating that a more complex model leads to performance degradation. This decline is likely due to the limited size of their datasets, which avoids effective learning from larger models. For clients 1, 4, 5, 8, and 9, accuracy remains relatively stable across different model sizes, suggesting that even a smaller model could be effectively used for accurate classification. This is particularly beneficial for deployment on tiny devices, which require lightweight models due to hardware constraints. Particularly, clients 1, 8, and 9, achieve higher accuracy compared to the other clients.

Using the pruning technique, we modified the model and analyzed its performance for each client. Several previous works have also employed the L_2 norm in pruning strategies (Agarwal, Mathew, Patel, Tripathi, & Swami, 2024; Gaikwad & El-Sharkawy, 2019; Geng, Gao, Zhang, & Xu, 2024; Liu et al., 2024; Lovenia, Jemima, Raghul, & Christopher, 2021) motivating our adoption of this criterion in the federated context. Specifically, we adopted the L_2 norm as the importance criterion

Table 4
Weighted average of different FL techniques.

Weighted average													
Client		1	2	3	4	5	6	7	8	9	10	11	12
FedAvg (batch size = 256)	Precision	0.98	0.90	0.92	0.68	0.54	0.79	0.93	0.99	1.00	1.00	0.77	0.62
	Recall	0.98	0.78	0.90	0.29	0.48	0.62	0.90	0.98	0.94	0.67	0.70	0.48
	F1-score	0.98	0.82	0.91	0.29	0.48	0.64	0.91	0.98	0.97	0.80	0.71	0.51
FedAvg (batch size = 128)	Precision	0.99	0.97	0.93	0.39	0.60	0.83	0.96	0.99	1.00	1.00	0.88	0.80
	Recall	0.99	0.95	0.89	0.25	0.52	0.60	0.91	0.97	0.93	0.60	0.83	0.54
	F1-score	0.99	0.96	0.90	0.25	0.52	0.60	0.93	0.98	0.96	0.75	0.84	0.57
FedProx	Precision	0.92	0.96	0.87	0.58	0.58	0.84	0.94	0.99	1.00	1.00	0.82	0.83
	Recall	0.92	0.89	0.79	0.29	0.44	0.57	0.83	0.94	0.74	0.59	0.70	0.38
	F1-score	0.92	0.92	0.81	0.36	0.49	0.60	0.88	0.96	0.85	0.74	0.75	0.46
SCAFFOLD	Precision	0.99	0.98	0.98	0.77	0.72	0.98	0.99	0.99	0.99	0.99	0.99	0.88
	Recall	0.99	0.98	0.98	0.72	0.73	0.98	0.99	0.99	1.00	0.99	0.99	0.88
	F1-score	0.99	0.98	0.98	0.74	0.72	0.98	0.99	0.99	1.00	0.99	0.99	0.88
FedBN	Precision	0.24	0.87	0.40	0.33	0.27	0.23	0.60	0.54	1.00	1.00	0.62	0.51
	Recall	0.25	0.46	0.28	0.22	0.26	0.24	0.37	0.35	0.68	0.01	0.30	0.45
	F1-score	0.24	0.60	0.31	0.23	0.26	0.24	0.41	0.39	0.81	0.01	0.32	0.46
FedDF	Precision	1.00	0.99	1.00	0.89	0.96	1.00	1.00	1.00	1.00	1.00	1.00	1.00
	Recall	0.96	0.98	0.93	0.62	0.74	0.95	0.95	0.94	0.99	0.96	0.91	0.81
	F1-score	0.97	0.98	0.97	0.71	0.81	0.97	0.97	0.96	1.00	0.98	0.95	0.88

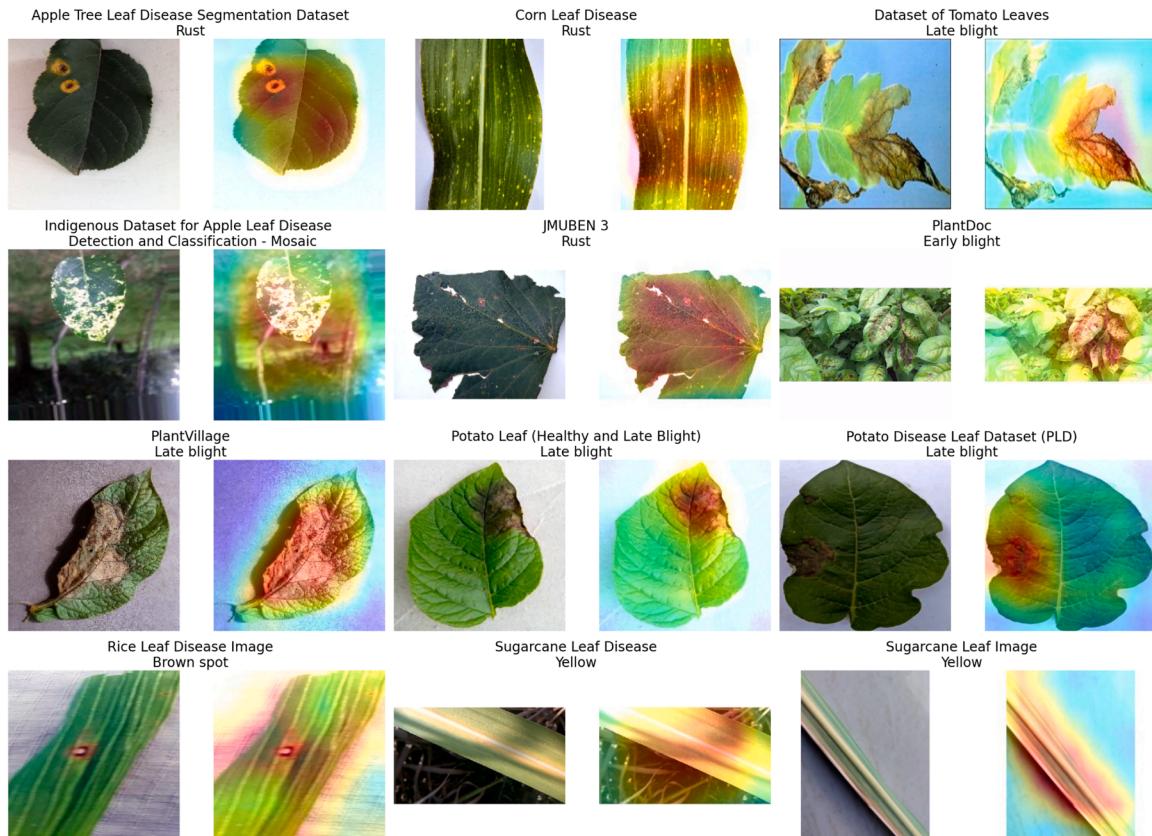


Fig. 6. Heatmap for each client generated on a correctly classified test image: the original image is displayed on the left, with the corresponding heatmap overlaid on the right.

for channel pruning due to its ability to better handle data heterogeneity across clients. Unlike more aggressive methods such as L_1 norm-based pruning, which may remove channels critical for certain clients, L_2 pruning gradually reduces the importance of less relevant channels without entirely eliminating them, ensuring that useful information is retained across all clients. This not only helps preserve both shared and client-specific features but also supports better generalization in federated settings, where image data vary in resolution, lighting, and acquisition conditions. Moreover, L_2 pruning may mitigate the issue of "client drift" in FL. If individual clients prune their models too aggressively

using L_1 norm-based pruning, they may remove critical features that are essential for other clients, leading to divergence in local models and impairing overall model convergence. By promoting a more balanced weight distribution, L_2 pruning helps maintain consistency across client models, reducing the risk of excessive deviation. Finally, to validate our selection of the L_2 norm as the pruning criterion, we conducted an empirical comparison of model accuracies resulting from L_1 and L_2 norm-based pruning across the selected pruning ratios (0.3, 0.5, 0.7, 0.9). The results, detailed in Table 5, demonstrate that the accuracies of the models pruned using the L_2 norm outperformed those obtained by pruning

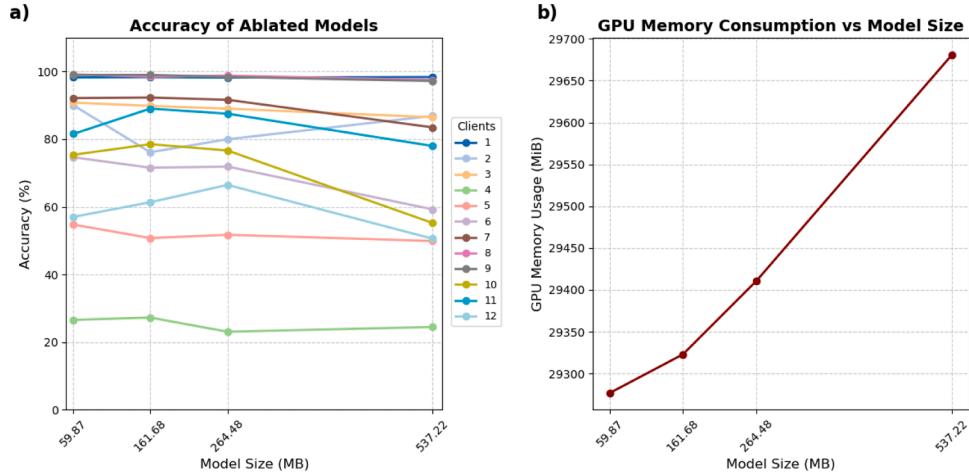


Fig. 7. a) Accuracies varying model size, for each client b) each GPU usage during training for the employed model and each ablated model. All experiments were conducted using the FedAvg aggregation strategy.

Table 5
Comparison of model accuracies resulting from L_1 and L_2 norm-based pruning across selected pruning ratios.

Pruning ratio	L1 norm				L2 norm			
	0.3	0.5	0.7	0.9	0.3	0.5	0.7	0.9
Accuracy (%)	96.5	76.72	49.37	25.54	97.63	88.24	57.86	28.64
	84.46	72.31	45.46	4.20	86.92	69.77	50	5.38
	68.27	45.63	19.54	19.09	74.81	80.70	27.47	19.3
	16.78	6.99	10.49	13.67	17.48	13.99	14.69	18.18
	45.94	26.42	23.83	10.15	48.53	23.66	24.01	12.26
	38.71	31.51	39.38	3.12	41.78	47.95	38.01	4.45
	76.5	66.17	37.17	4.5	82	70.5	42	4.5
	90.17	42.34	40.71	8.34	95.05	78.06	44.62	7.09
	91.93	69.40	0.12	0	89.03	88.67	0.12	0
	35.79	6.25	2.5	0	36.25	26.01	0	0
Accuracy (%)	62.39	31.84	15.22	2.79	61.79	33.33	30.45	3.28
	37.81	33.33	26.7	9.22	38.14	34.32	30.35	13.27

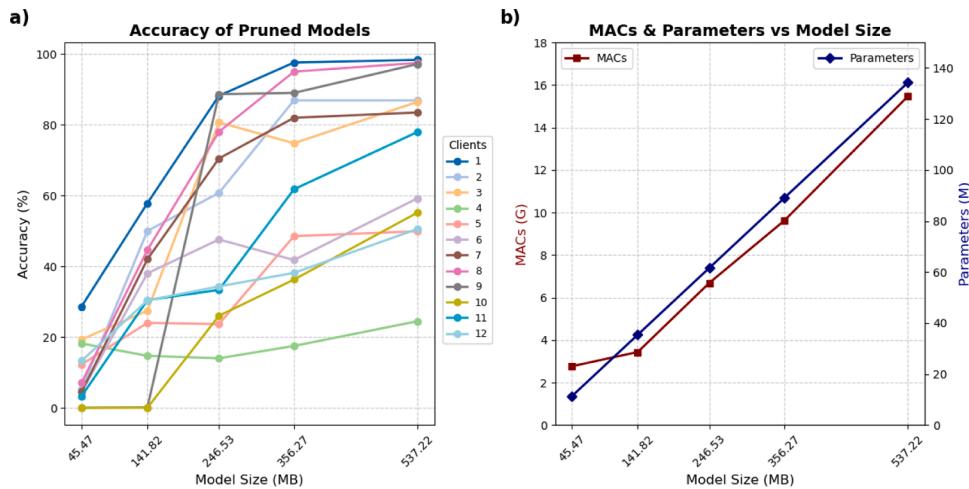


Fig. 8. a) Pruned model accuracies for each client, varying the model size and b) comparison between model size, MACs, and Parameters for each pruned model. All experiments were conducted using the FedAvg aggregation strategy with a batch size of 256.

with the L_1 norm. In light of these empirical results, we have decided to adopt the L_2 norm as our exclusive pruning criterion in subsequent experiments.

The accuracies obtained for the models, pruned using the L_2 norm and with sizes of 356.27 MB, 246.53 MB, 141.82 MB, 45.47 MB are shown in Fig. 8a). As shown, excessive model reduction leads to a

general drop in performance across all clients. Notably, for clients 9 and 10, performance drops to 0% when using models of size 45.47 MB and 141.82 MB. Overall, the smallest model does not perform well for all clients. As the model size increases, the MACs and the number of parameters also increase, as illustrated in Fig. 8b).

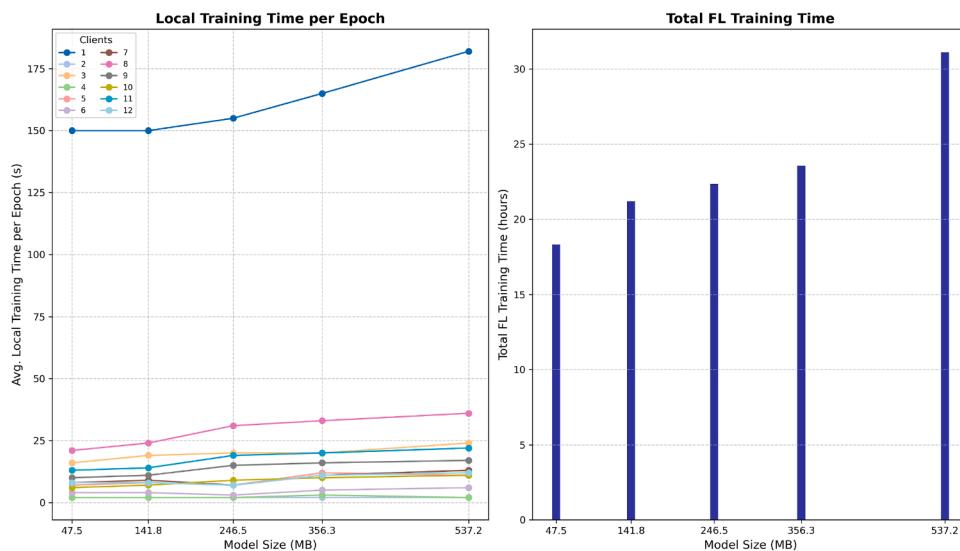


Fig. 9. a) Average local training time per epoch for each client, showing that clients with larger datasets require more time to complete an epoch, for different models. b) Total FL training time over 100 rounds, highlighting the influence of model size on overall training duration in a synchronous FL setting. All experiments were conducted using the FedAvg aggregation strategy with a batch size of 256.

FL with Flower is performed synchronously, meaning the global model is aggregated only after all clients have completed their local training for a given round. This approach introduces waiting times for clients with smaller datasets, as they complete their training faster than those with larger datasets. Among all datasets, PlantVillage contains the largest amount of data and requires the longest time to complete a single local epoch. The local training time per epoch is shown in Fig. 9a), comparing models of various sizes obtained after pruning with the initial model. With the same models, the total FL training time over 100 rounds is presented in Fig. 9b). As shown, time increases as the model size increase.

Additionally, the recommender not only provides a list of websites with information about the identified plant and disease but also offers

personalized recommendations using the Google Gemini API. This personalization is tailored to the specific plant and disease detected. As shown in Fig. 10, which presents a simulated conversation between the user and the chatbot, the API delivers a comprehensive response regarding the treatment of the disease, "Powdery Mildew" in this case, along with guidance on disease prevention, occurrence frequency, and severity.

To further contextualize the performance of our FL approach, we compared it with a state-of-the-art centralized model, EfficientNetB7, trained on the aggregated dataset. The centralized model achieved a high overall accuracy of 98.75 %, confirming its strong capability in leaf disease classification when data from all sources are available in a centralized manner. The detailed performance obtained at varying the

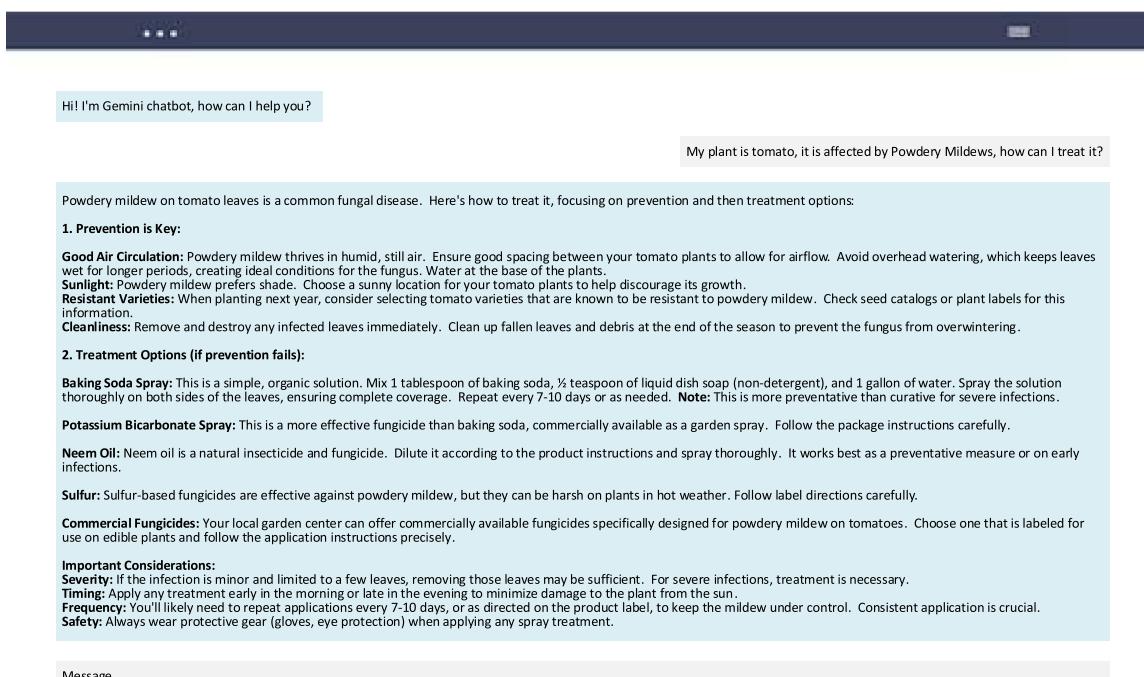


Fig. 10. Chatbot dialogue process powered by the Google Gemini API for plant disease treatment recommendations. The system initiates the conversation with a general prompt based on the identified plant disease and dynamically generates personalized treatment plans based on user input.

Table 6
EfficientNetB7 results.

Class	Precision	Recall	F1-score
Healthy	0.99	0.99	0.99
Late blight	0.97	0.98	0.98
Early blight	0.98	0.97	0.97
Bacterial spot	0.99	0.98	0.98
Powdery mildew	0.98	0.99	0.98
Black rot	0.99	0.98	0.99
Rust	0.99	0.99	0.99
Brown spot	1.00	1.00	1.00
Yellow	0.99	1.00	0.99
Mosaic	0.98	0.98	0.98

disease classes is reported in [Table 6](#). However, in real-world agricultural settings, centralized training approaches are often unuseful. This is primarily due to the decentralized nature of data collection in agriculture: plant disease images are typically captured in geographically dispersed locations such as individual farms or research institutions which may have limited connectivity in sharing data due to privacy, proprietary, or regulatory concerns. In contrast, FL offers a privacy-preserving and communication-efficient alternative by enabling model training to occur directly on local devices or nodes. In this way, raw data remains on the source device, and only model updates are shared preserving data confidentiality.

7. Conclusion and future work

This work presents AGRIFOLD, an FL framework for LDD, addressing the challenges of decentralized agricultural data while ensuring privacy. By integrating an ECA in a VGG16-based model, the framework is able to correctly classify leaf diseases in an FL environment, in non-IID conditions. The explainability of the model is further improved through heatmaps that highlight diseased leaf regions. To support real-world deployment, ablation and pruning techniques were applied, enabling the model to run efficiently on resource-constrained edge devices. Furthermore, the inclusion of an LLM-based recommender and a list of websites to consult based on disease is helpful for leaf and plant treatment. Evaluations on 12 heterogeneous datasets demonstrated the framework's robustness, achieving high classification accuracy across 10 plant disease classes at varying FL methods. Hence, by combining FL, advanced DL architectures, and AI-driven disease analysis, AGRIFOLD represents a powerful tool for modern agriculture, helping farmers reduce losses. Future work may explore the use of asynchronous FL, where the server updates and redistributes the global model immediately upon receiving updates from individual clients. This approach could significantly reduce idle time for clients with shorter local training durations, improving overall system efficiency. Additionally, this study can be expanded by incorporating a larger number of datasets and introducing more disease classes. Further improvements can also be made to the recommendation process by integrating additional information sources and enhancing the chatbot to provide a more personalized user experience.

Data and resources

The image datasets used in this study are (last accessed January 2025):

- [Apple Tree Leaf Disease Segmentation Dataset](#)
- [Corn Leaf Disease](#)
- [Dataset of Tomato Leaves](#)
- [Indigenous Dataset for Apple Leaf Disease Detection and Classification](#)
- [JMUBEN 3](#)
- [PlantDoc](#)
- [PlantVillage](#)

- [Potato Disease Leaf Dataset \(PLD\)](#)
- [Potato Leaf \(Healthy and Late Blight\)](#)
- [Rice Leaf Disease Image](#)
- [Sugarcane Leaf Disease](#)
- [Sugarcane Leaf Images](#)

CRediT authorship contribution statement

Francesco Piccialli: Conceptualization, Methodology, Validation, Supervision, Writing – review & editing; **Ciro Della Bruna:** Investigation, Data curation, Conceptualization, Methodology, Formal analysis, Writing – review & editing; **Diletta Chiaro:** Methodology, Formal analysis; **Pian Qi:** Data curation, Investigation, Methodology; **Martina Savoia:** Investigation, Data curation, Conceptualization, Methodology, Formal analysis, Writing – review & editing.

Data availability

No data was used for the research described in the article.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by:

- PNRR project FAIR - Future Artificial Intelligence Research (PE00000013), Spoke 3, under the NRRP MUR program funded by the NextGenerationEU (CUP: E63C22002150007)
- PNRR Centro Nazionale HPC, Big Data e Quantum Computing, (CN_00000013) (CUP: E63C22000980007), Innovation Fund: SOW, under the NRRP MUR program funded by the NextGenerationEU.

The authors thank the IBiSco project (Infrastructure for Big data and Scientific Computing), PON R&I 2014-2020 under Call 424-2018 - Action II.1, for the support and use of the HPC Cluster. The authors extend special thanks to Dr. Luisa Carraciulo and Eng. Davide Bottalico for their constant and continuous support in utilizing the IBiSco HPC Cluster.

References

- Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F.L., & McGrew, B. (2023). Gpt-4 technical report. Technical Report arXiv preprint. <https://doi.org/10.48550/arXiv.2303.08774>
- Agarwal, P., Mathew, M., Patel, K.R., Tripathi, V., & Swami, P. (2024). Prune efficiently by soft pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* The IEEE/CVF Conference on computer vision and pattern recognition (pp. 2210–2217).
- Aggarwal, M., Khullar, V., Goyal, N., & Prola, T.A. (2024). Resource-efficient federated learning over IoT for rice leaf disease classification. *Computers and Electronics in Agriculture*, 221, 109001.
- Antico, T.M., Moreira, L.F.R., & Moreira, R. (2024). Evaluating the potential of federated learning for maize leaf disease prediction. Technical Report arXiv preprint. <https://doi.org/10.48550/arXiv.2412.07872>
- Ashok, S., Kishore, G., Rajesh, V., Suchitra, S., Sophia, S.G., & Pavithra, B. (2020). Tomato leaf disease detection using deep learning techniques. In *2020 5th International conference on communication and electronics systems (icces)* (pp. 979–983). IEEE. <https://doi.org/10.1109/ICCES48766.2020.9137986>
- Barone, G., Battista, Bottalico, D., Carraciulo, L., Doria, Michelino, A., Pardi, D., Russo, S., Sabella, G., Spisso, G., & Bernardino. 2022 International conference on electrical, computer and energy technologies (ICECET), designing and implementing a high-performance computing heterogeneous cluster (vol. 2022). <https://doi.org/10.1109/ICECET55527.2022.9872709>
- Beutel, D.J., & et.al. (2022). Flower: A friendly federated learning framework.
- Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., & Amodei, D. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901. <https://doi.org/10.48550/arXiv.2005.14165>

- Daphal, S., & Koli, S. (2022). Sugarcane leaf disease dataset, Mendeley data. <https://doi.org/10.17632/9424skmnrk1>
- Demilie, W.B. (2024). Plant disease detection and classification techniques: A comparative study of the performances. *Journal of Big Data*, 11(1), 5. <https://doi.org/10.1186/s40537-023-00863-9>
- Du, B., Li, W., & Qin, X. (2024). A classification and recognition model for multiple fruit tree leaf diseases. *Environmental Research Communications*, 6(10), 105034. <https://doi.org/10.1088/2515-7620/ad87b6>
- Fahim-Ul-Islam, M., Chakrabarty, A., Ahmed, S.T., Rahman, R., Kwon, H.H., & Piran, M.J. (2024). A comprehensive approach towards wheat leaf disease identification leveraging transformer models and federated learning. <https://doi.org/10.1109/ACCESS.2024.3438544>
- Fang, G., Ma, X., Song, M., Mi, M.B., & Wang, X. (2023). Depgraph: Towards any structural pruning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* The IEEE/CVF conference on computer vision and pattern recognition (pp. 16091–16101). <https://doi.org/10.48550/arXiv.2301.12900>
- Fathima, M.S., & Booba, B. (2024). Enhancing plant leaf disease detection: Integrating mobilenet with local binary pattern And visualizing insights with grad-cam (vol. 1). IEEE. <https://doi.org/10.1109/ICCPCT61902.2024.10673312>
- Feng, J., & Chao, X. (2022). Apple tree leaf disease segmentation dataset. <https://doi.org/10.11922/sciencebd.01627>
- Gaikwad, A.S., & El-Sharkawy, M. (2019). Pruning the convolution neural network (squeezenet) based on l 2 normalization of activation maps. In *2019 IEEE 9th Annual computing and communication workshop and conference (ccwc)* (pp. 392–0396). IEEE. <https://doi.org/10.1109/CCWC2019.8666597>
- Geng, X., Gao, J., Zhang, Y., & Xu, D. (2024). Complex hybrid weighted pruning method for accelerating convolutional neural networks. *Scientific Reports*, 14(1), 5570. <https://doi.org/10.1038/s41598-024-55942-5>
- Gupta, P.S., Hans, P., & Chand, S. (2020). Classification of plant leaf diseases using machine learning and image preprocessing techniques.
- Hari, P., & Singh, M.P. (2025). Adaptive knowledge transfer using federated deep learning for plant disease detection. *Computers and electronics in agriculture* (vol. 229). <https://doi.org/10.1016/j.compag.2024.109720>
- Hari, P., Singh, M.P., & Singh, A.K. (2024). An improved federated deep learning for plant leaf disease detection. *Multimedia Tools and Applications*, 83, 83471–83491. <https://doi.org/10.1007/s11042-024-18867-9>
- <https://www.britannica.com/>.
- https://en.wikipedia.org/wiki/Main_Page. (last accessed January 2025).
- <https://ipm.ucanr.edu/#gsc.tab=0>. (last accessed January 2025).
- <https://www.canr.msu.edu/>. (last accessed January 2025).
- <https://cropscience.bayer.co.uk/>. (last accessed January 2025).
- <https://www.koppert.com/>. (last accessed January 2025).
- <https://plantix.net/en/>. (last accessed January 2025).
- Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* The IEEE conference on computer vision and pattern recognition (pp. 7132–7141). <https://doi.org/10.1109/CVPR.2018.00745>
- Huang, M.L., Chang, & Ya-Han (2020). Dataset of tomato leaves, Mendeley data. <https://doi.org/10.17632/ngdgg79rz8.1>
- Indika, S., & Saputra (2023). Corn leaf disease [data set]. <https://doi.org/10.34740/KAGGLE/DS/6341289>
- J, Pandian, A., & Gopal, G. (2019). Data for: Identification of plant leaf diseases using a 9-layer deep convolutional neural network, Mendeley data (vol. 1). <https://doi.org/10.17632/twbtbsjrv.1>
- Jepkoech, J., Mugambi, D., Kenduiwywo, B., & Too, E. (2023). JMUBEN 3, Mendeley data (vol. 4). <https://doi.org/10.17632/jjn4ht687d.4>
- Ji, Z., Bao, S., Chen, M., & Wei, L. (2024). ICS-ResNet: A lightweight network for maize leaf disease classification. *Agronomy*, 14(7), 1587. <https://doi.org/10.3390/agronomy14071587>
- Jiang, D., Li, F., Yang, Y., & Yu, S. (2020). A tomato leaf diseases classification method based on deep learning. In *2020 Chinese control and decision conference (ccde)* (pp. 1446–1450). IEEE. <https://doi.org/10.1109/CCDC49329.2020.9164457>
- Joshi, J., Aeri, M., Mukreja, V., & Mehta, S. (2024). Revolutionizing in agriculture: Federated cnn models for sunflower leaf diseases. In *2024 11th International conference on reliability, infocom technologies and optimization (trends and future directions)(icrito)* (pp. 1–6). IEEE. <https://doi.org/10.1109/ICRITO61523.2024.10522441>
- Jung, M., Song, J.S., Shin, A.Y., Choi, B., Go, S., Kwon, S.Y., & Kim, Y.M. (2023). Construction of deep learning-based disease detection model in plants. *Scientific Reports*, 13(1), 7331. <https://doi.org/10.1038/s41598-023-34549-2>
- Kalita, K., Ramesh, J. V.N., Cepova, L., Pandya, S.B., Jangir, P., & Abualigah, L. (2024). Multi-objective exponential distribution optimizer (moedo): A novel math-inspired multi-objective algorithm for global optimization and real-world engineering design problems. *Scientific Reports*, 14(1), 1816.
- Karim, M.J., Goni, M. O.F., Nahiduzzaman, M., Ahsan, M., Haider, J., & Kowalski, M. (2024). Enhancing agriculture through real-time grape leaf disease classification via an edge device with a lightweight. *CNN Architecture and Grad-CAM*. *Scientific Reports*, 14(1), 16022. <https://doi.org/10.1038/s41598-024-66989-9>
- Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., & Suresh, A.T. (2020). Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning* (pp. 5132–5143). <https://doi.org/10.48550/arXiv.1910.06378>
- Khobragade, P., Shriwas, A., Shinde, S., Mane, A., & Padole, A. (2022). Potato leaf disease detection using cnn. Technical Report SMART GENCON. <https://doi.org/10.1109/SMArtGENCON56628.2022.10083986>
- Kini, A.S., Prema, K.V., & Pai, S.N. (2024). Early stage black pepper leaf disease prediction based on transfer learning using convnets. *Scientific Reports*, 14(1), 1404. <https://doi.org/10.1038/s41598-024-51884-0>
- Kiriella, S., Fernando, S., Sumathipala, S., & Udayakumara, E. (2023). Explainable ai techniques for deep convolutional neural network based plant disease identification. In *2023 8th International conference on information technology research (icitr)* (pp. 1–6). IEEE. <https://doi.org/10.1109/ICITR61062.2023.10382942>
- Li, T., Sahu, A. K., Talwalkar, A., & Smith, V. (2020a). Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3), 50–60. <https://doi.org/10.1109/MSP.2020.2975749>
- Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020b). Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2, 429–450. <https://doi.org/10.48550/arXiv.1812.06127>
- Li, X., Jiang, M., Zhang, X., Kamp, M., & Dou, Q. (2021). Fedbn: Federated learning on non-iid features via local batch normalization. Technical Report arXiv preprint.
- Lin, T., Kong, L., Stich, S.U., & Jaggi, M. (2020). Ensemble distillation for robust model fusion in federated learning. *Advances in Neural Information Processing Systems*, 33, 2351–2363.
- Liu, J., Zhou, Q., He, Y., Chen, W., Yin, Z., Yan, F., & Wu, Y. (2024). Research on group-level pruning algorithm based on progressive sparsity regularization. In *2024 3rd International conference on electronics and information technology* (pp. 810–814). IEEE. <https://doi.org/10.1109/EIT63098.2024.10762470>
- Liu, T., Wang, R., Chen, J., Han, S., & Yang, J. (2018). Fine-grained classification of product images based on convolutional neural networks. *Advances in Molecular Imaging*, 8(4), 69–87. <https://doi.org/10.4236/ami.2018.84007>
- López, M.M., Bertolini, E., Olmos, A., Caruso, P., Gorris, M.T., Llop, P., & Cambra, M. (2003). Innovative tools for detection of plant pathogenic viruses and bacteria. *International Microbiology*, 6, 233–243. <https://doi.org/10.1007/s10123-003-0143-y>
- Lovenia, J.D.L., Jemima, D.D., Rahghul, R., & Christopher, N. (2021). Plant disease classification using pruning techniques. In *2021 3rd International conference on signal processing and communication (ICPSC)* (pp. 350–352). <https://doi.org/10.1109/ICPSC51351.2021.9451730>
- Mac, T.T., Nguyen, T.D., Dang, H.K., Nguyen, D.T., & Nguyen, X.T. (2024). Intelligent agricultural robotic detection system for greenhouse tomato leaf diseases using soft computing techniques and deep learning. *Scientific Reports*, 14(1), 23887. <https://doi.org/10.1038/s41598-024-75285-5>
- Martinelli, F., Scalenghe, R., Davino, S., Panno, S., Scuderi, G., Ruisi, P., & Dandekar, A.M. (2015). Advanced methods of plant disease detection. A review. *Agronomy for sustainable development* (vol. 35). <https://doi.org/10.1007/s13593-014-0246-1>
- Mcmahan, B., Moore, E., Ramage, D., Hampson, S., & Arcas, B.A. (2017). Communication-efficient learning of deep networks from decentralized data. In *Artificial Intelligence and Statistics* (pp. 1273–1282). <https://doi.org/10.48550/arXiv.1602.05629>
- Mehedi, M. H.K., Hosain, A.S., Ahmed, S., Promita, S.T., Muna, R.K., Hasan, M., & Reza, M.T. (2022). Plant leaf disease detection using transfer learning and explainable ai. In *2022 IEEE 13th Annual information technology, electronics and mobile communication conference (iemcon)* (pp. 166–0170). IEEE. <https://doi.org/10.1109/IECON56893.2022.9946513>
- Nigar, N., Faisal, H.M., Umer, M., Oki, O., & Lukose, J. (2024). Improving plant disease classification with deep learning based prediction model using explainable artificial intelligence. <https://doi.org/10.1109/ACCESS.2024.3428553>
- Oad, A., Abbas, S.S., Zafar, A., Akram, B.A., Dong, F., Talpur, M. S.H., & Uddin, M. (2024). Plant leaf disease detection using ensemble learning and explainable ai. <https://doi.org/10.1109/ACCESS.2024.3484574>
- Pandy, S.B., Kalita, K., Jangir, P., Cep, R., Migdady, H., Chohan, J. S., & Mallik, S. (2024a). Multi-objective rime algorithm-based techno economic analysis for security constraints load dispatch and power flow including uncertainties model of hybrid power systems. *Energy Reports*, 11, 4423–4451.
- Pandy, S.B., Kalita, K., Jangir, P., Ghadai, R.K., & Abualigah, L. (2024b). Multi-objective geometric mean optimizer (momo): A novel metaphor-free population-based math-inspired multi-objective algorithm. *International Journal of Computational Intelligence Systems*, 17(1), 91.
- Patil, P., Pamali, S.K., Devagiri, S.B., Sushma, A.S., & Mirje, J. (2024). Plant leaf disease detection using xai. In *2024 3rd International conference on artificial intelligence for internet of things (aiiot)* (pp. 1–6). IEEE. <https://doi.org/10.1109/AlloT58432.2024.10574617>
- Prasad, K.V., Vaidya, H., Rajashekhar, C., Karekal, K.S., Sali, R., & Nisar, K.S. (2024). Multiclass classification of diseased grape leaf identification using deep convolutional neural network (dcnn) classifier. *Scientific Reports*, 14(1), 9002. <https://doi.org/10.1038/s41598-024-59562-x>
- Rahman, H., Ahmad, I., Jon, P.H., Salam, A., & Rabbi, M.F. (2024). Automated detection of selected tea leaf diseases in Bangladesh with convolutional neural network. *Scientific Reports*, 14(1), 14097. <https://doi.org/10.1038/s41598-024-62058-3>
- Rakib, A.F., Rahman, R., Razi, A.A., & Hasan, A.T. (2024). A lightweight quantized cnn model for plant disease recognition. *Arabian Journal for Science and Engineering*, 49(3), 4097–4108. <https://doi.org/10.1007/s13369-023-08280-z>
- Rizwan, M., Saeed, & Rashid, D.J. (2021). Potato disease leaf dataset(PLD) [data set]. Kaggle. <https://doi.org/10.34740/KAGGLE/DS/1562973>
- Rozaqi, A.J., & Sunyoto, A. (2020). Identification of disease in potato leaves using convolutional neural network (CNN) algorithm. In *2020 3rd International conference on information and communications technology (icoict)* (pp. 72–76). IEEE. <https://doi.org/10.1109/ICOIACT50329.2020.9332037>
- Sankaran, S., Mishra, A., Ehsani, R., & Davis, C. (2010). A review of advanced techniques for detecting plant diseases: Computers and electronics in agriculture (vol. 72). <https://doi.org/10.1016/j.compag.2010.02.007>
- Sarkar, M.S., Lifat, M. A.S., Hasan, R., & Hassan, M.M. (2024). Discovering the depths of cotton leaf disease detection: Integrating hypertuned residual networks with gradcam xai for enhanced understanding and diagnosis. In *2024 3rd International conference on advancement in electrical and electronic engineering (icaeee)* (pp. 1–6). IEEE. <https://doi.org/10.1109/ICAEEE62219.2024.10561737>

- Sethy, P.K., Barpanda, N.K., Rath, A.K., & Behera, S.K. (2020). Deep feature based rice leaf disease identification using support vector machine. *Computers and Electronics in Agriculture*, 175, 105527. <https://doi.org/10.1016/j.compag.2020.105527>
- Shen, Y., Xi, W., Cai, Y., Fan, Y., Yang, H., & Zhao, J. (2025). Multi-objective federated learning: Balancing global performance and individual fairness. *Future Generation Computer Systems*, 162, 107468.
- Sholihati, R.A., Sulistijono, I.A., Risnumawan, A., & Kusumawati, E. (2020). Potato leaf disease classification using deep learning approach. In *2020 International electronic symposium (ies)* (pp. 392–397). IEEE. <https://doi.org/10.1109/IES50839.2020.9231784>
- Shrivastava, R.K., Dwivedi, A.K., Gupta, S.K., Hazra, D., Maurya, A., Gupta, A., & Singh, B.P. (2024). A generalized federated learning approach for robust crop disease classification across diverse farms. In *2024 IEEE International conference on advanced networks and telecommunications systems (ants)* (pp. 78–83). IEEE. <https://doi.org/10.1109/ANTS63515.2024.10898874>
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. Technical report arXiv preprint. <https://doi.org/10.48550/arXiv.1409.1556>
- Singh, A.K., Chattopadhyay, P., & Singh, L. (2024). A new federated learning framework for plant leaf disease diagnosis. In *2024 International conference on communication, control, and intelligent systems (ccis)* (pp. 1–7). IEEE. <https://doi.org/10.1109/CCIS63231.2024.10932109>
- Singh, D., Jain, N., Jain, P., Kayal, P., Kumawat, S., Batra, & Nipun, B. (2020). Plantdoc: A dataset for visual plant disease detection. In *Proceedings of the 7th the 7th (pp. 249–253)*. Association for Computing Machinery. <https://doi.org/10.1145/3371158.3371196>
- Singla, P., Sharma, R., Kukreja, V., & Bansal, A. (2022). Deep learning based multi-classification model for rice disease detection. In *2022 10th International conference on reliability, infocom technologies and optimization (trends and future directions)(ICRITO)* (pp. 1–5). IEEE. <https://doi.org/10.1109/ICRITO56286.2022.9965117>
- Suryavanshi, A., Kukreja, V., Malhotra, S., Mehta, S., & Choudhary, A. (2024a). Transforming crop diagnosis: The role of federated learning CNN in cauliflower leaf disease detection. In *2024 3rd International conference for innovation in technology (INOCON)* (pp. 1–6). IEEE. <https://doi.org/10.1109/INOCON60754.2024.10512092>
- Suryavanshi, A., Kukreja, V., Upadhyay, D., Mehta, S., & Thapliyal, S. (2024b). Federated learning cnn for broccoli leaf disease control: A path to sustainable farming. In *2024 11th International conference on reliability, infocom technologies and optimization (trends and future directions)(ICRITO)* (pp. 1–5). IEEE. <https://doi.org/10.1109/ICRITO61523.2024.10522286>
- Tang, Z., Yang, J., Li, Z., & Qi, F. (2020). Grape disease image classification based on lightweight convolution neural networks and channelwise attention. *Computers and Electronics in Agriculture*, 178, 105735. <https://doi.org/10.1016/j.compag.2020.105735>
- Thite, S., Suryavanshi, Y., Patil, K., & Chumchu, P. (2023). Sugarcane leaf image dataset. Mendeley Data. <https://doi.org/10.17632/9twjtv92vk.1>
- Tilahun, N. (2020). Potato leaf (healthy and late blight) (vol. 1). Mendeley Data. <https://doi.org/10.17632/v4w72bsts5.1>
- Upadhyay, D., Verma, A., Kukreja, V., & Mehta, S. (2024). Tech-driven agronomy: Federated learning cnn's for aloe vera leaf disease diagnosis. In *2024 11th international conference on reliability, infocom technologies and optimization (trends and future directions)(ICRITO)* (pp. 1–6). IEEE. <https://doi.org/10.1109/ICRITO61523.2024.10522456>
- Vats, S., Kukreja, V., & Mehta, S. (2024). A new era in agritech: Federated learning cnn for jute leaf disease identification (vol. 2). IEEE. <https://doi.org/10.1109/IATMSI60426.2024.10502815>
- Vishnoi, V.K., Kumar, K., Kumar, B., Mohan, S., & Khan, A.A. (2022). Detection of apple plant diseases using leaf images through convolutional neural network. *IEEE Access*, 11, 6594–6609. <https://doi.org/10.1109/ACCESS.2022.3232917>
- Wang, P., Niu, T., Mao, Y., Liu, B., Yang, S., He, D., & Gao, Q. (2021a). Fine-grained grape leaf diseases recognition method based on improved lightweight attention network. *Frontiers in Plant Science*, 12, 738042. <https://doi.org/10.3389/fpls.2021.738042>
- Wang, P., Niu, T., Mao, Y., Zhang, Z., Liu, B., & He, D. (2021b). Identification of apple leaf diseases by improved deep convolutional neural networks with an attention mechanism. *Frontiers in Plant Science*, 12, 723294. <https://doi.org/10.3389/fpls.2021.723294>
- Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q. (2020a). ECA-Net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* the IEEE/CVF conference on computer vision and pattern recognition (pp. 11534–11542). <https://doi.org/10.1109/CVPR42600.2020.01155>
- Wang, Q., Wu, B., Zhu, P., Li, P., Zuo, W., & Hu, Q. (2020b). ECA-Net: Efficient channel attention for deep convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* The IEEE/CVF conference on computer vision and pattern recognition (pp. 11534–11542). <https://doi.org/10.1109/CVPR42600.2020.01155>
- Xiao, Z., Shi, Y., Zhu, G., Xiong, J., & Wu, J. (2023). Leaf disease detection based on lightweight deep residual network and attention mechanism. *IEEE Access*, 11, 48248–48258. <https://doi.org/10.1109/ACCESS.2023.3272985>
- Yang, Q., Liu, Y., Chen, T., & Tong, Y. (2019). Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2), 1–19. <https://doi.org/10.1145/3298981>
- Yatoo, A., & Sharma, A. (2024). Indigenous dataset for apple leaf disease detection and classification. Mendeley data (vol. 3). <https://doi.org/10.17632/9m2dcb5mmr.3>
- Yebasse, M., Shimelis, B., Warku, H., Ko, J., & Cheoi, K.J. (2021). Coffee disease visualization and classification. *Plants*, 10(6), 1257. <https://doi.org/10.3390/plants10061257>
- Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., & Gao, Y. (2021). A survey on federated learning. Knowledge-based systems. <https://doi.org/10.1016/j.knosys.2021.106775>
- Zhong, Y., & Zhao, M. (2020). Research on deep learning in apple leaf disease recognition. *Computers and electronics in agriculture* (vol. 168). <https://doi.org/10.1016/j.compag.2019.105146>