Strategic Realignment of Edge-Based Active Learning: A Methodological Framework for Validating Agricultural Computer Vision Pipelines

1. Executive Summary and Strategic Imperative

The deployment of deep learning models on edge devices within agricultural environments represents one of the most challenging frontiers in contemporary computer vision. The convergence of resource-constrained hardware, highly variable environmental conditions, and the scarcity of high-quality annotated data creates a "perfect storm" of complexity. The initial phase of this research, which focused on the direct application of Active Learning (AL) algorithms to the PlantDoc dataset on edge devices, yielded suboptimal results. This outcome, while frustrating, is consistent with the behavior of Active Learning systems when subjected to high-levels of aleatoric uncertainty—noise inherent in the data—before the fundamental query mechanisms have been validated in a controlled setting.

The primary failure mode identified in the initial experiments was the confounding of *informative uncertainty* (what the model needs to learn) with *noise-induced uncertainty* (what the model cannot resolve due to background clutter or label error). PlantDoc, while representative of the ultimate deployment target, contains significant label noise and background complexity that obscures the true performance of the Active Learning selection logic. When an edge model with limited capacity is forced to query samples from such a noisy distribution, it frequently succumbs to the "outlier problem," wasting its limited annotation budget on artifacts rather than disease features.

To rectify this and place the research on a rigorous trajectory, we must institute a **Curriculum Verification Protocol**. This strategy necessitates a temporary regression to lower-complexity datasets to isolate and validate the algorithmic components of the pipeline. By systematically testing the AL loop on "Toy" datasets (Fashion-MNIST, CIFAR-10), followed by "Controlled Agricultural" datasets (PlantVillage, Fruits-360), and finally "High-Quality Field" datasets (WGISD, MinneApple), we can decouple errors in software logic from errors induced by data pathology. This report outlines a comprehensive roadmap for this pivot, providing the theoretical justification, dataset specifications, and experimental protocols required to act as a "Sanity Check" for the entire edge-learning architecture.

2. Post-Mortem Analysis: The Structural Limitations of PlantDoc

Understanding precisely why the PlantDoc experiment failed is a prerequisite for designing the solution. The issues encountered were not merely significantly lower accuracy or slow convergence; they were symptomatic of a fundamental mismatch between the dataset's characteristics and the requirements of early-stage Active Learning validation.

2.1 The Confounding Influence of Background Clutter

Active Learning algorithms, particularly those based on uncertainty sampling (e.g., Entropy, Least Confidence), rely on the model's output probability distribution to measure "confusion." In a theoretical ideal, high entropy indicates that the model has identified a semantic ambiguity between two valid classes (e.g., distinguishing *Early Blight* from *Late Blight*). However, PlantDoc images are characterized by uncontrolled field settings containing mulch, soil, human hands, and other plants.

When a lightweight edge model—already compressed via quantization or pruning to fit hardware constraints —processes these images, it often fails to suppress the background features. Consequently, the model generates high uncertainty scores not because the disease morphology is ambiguous, but because the background texture is novel or complex. This phenomenon, known as "aleatoric uncertainty," is fatal to standard AL query strategies. The algorithm interprets the background noise as "informativeness" and prioritizes these useless samples for human annotation. This results in a feedback loop where the model optimizes for background recognition rather than pathology detection, leading to the stagnation of performance metrics observed in your experiments.

2.2 Label Noise as a Systemic Poison

Active Learning operates on the "Oracle Assumption"—the premise that the external annotator provides perfect ground truth for the selected queries. If the oracle provides incorrect labels, the highly weighted "informative" samples become "poisonous" to the training process. Literature reviewing PlantDoc indicates significant issues with label fidelity, including bounding boxes that fail to capture the entire leaf or miss symptoms entirely.

In a standard "Big Data" training regime (e.g., full supervised learning), the sheer volume of data can sometimes average out random label noise. However, Active Learning is a "Small Data" regime by design. The algorithm selectively constructs a small, potent training set. If even a small fraction of these high-leverage samples are mislabeled, the gradient descent trajectory is significantly altered. The "missing domain expertise" noted in the creation of PlantDoc suggests that the dataset contains systematic errors that an AL algorithm will inevitably amplify, causing the model to diverge rather than converge.

2.3 The Computational Bottleneck of Edge Learning

The initial experiments also likely suffered from the friction between the heavy preprocessing required for PlantDoc and the limited resources of edge devices. High-resolution, cluttered images require significant computational power for feature extraction. On devices like the Raspberry Pi or NVIDIA Jetson, the latency incurred during the "inference" phase of the AL loop (scanning the unlabeled pool) becomes prohibitive. If the model takes too long to compute uncertainty for 2,000 images, the "real-time" advantage of edge learning is lost. Furthermore, the need for heavy data augmentation to make PlantDoc usable adds to the computational overhead, potentially causing thermal throttling or memory overflows that disrupt the training process.

3. Phase I: Algorithmic Verification via "Toy" Datasets

The first step in our remedial protocol is to strip away the agricultural context entirely. We must prove that the Active Learning software stack—the code responsible for ranking, querying, and updating the model—is mathematically sound. For this, we utilize standard computer vision benchmarks. These datasets are "toys" only in terms of semantic complexity; in terms of statistical rigor, they are the industry standard for debugging machine learning pipelines.

3.1 Fashion-MNIST: The Baseline for Logic Validation

Dataset Rationale: Fashion-MNIST was developed specifically to replace the MNIST handwritten digit dataset, which had become too trivial for modern algorithms. It consists of 60,000 grayscale images (28x28 pixels) across 10 classes of clothing (e.g., T-shirt, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot).
Why It Fits the "Sanity Check":

- **Balanced Classes:** Unlike agricultural data, Fashion-MNIST is perfectly balanced (6,000 images per class). This eliminates class imbalance as a confounding variable, allowing us to test if the query strategy works under ideal conditions.
- **Low Dimensionality:** The small image size allows for extremely rapid training cycles. An edge-compatible model (e.g., a simple 3-layer CNN or a quantized MobileNet) can train to convergence in minutes on a standard GPU or reasonably quickly on a CPU-based edge device. This enables us to run 50+ repetitions of the AL experiment to generate statistically significant confidence intervals, a practice recommended by recent benchmarks like CDALBench.
- **Benchmark Integrity:** If our AL strategy (e.g., Entropy Sampling) cannot outperform Random Sampling on Fashion-MNIST, it effectively proves that the error lies in the *code* or the *hyperparameters*, not the data. We should expect to achieve >90% accuracy with only a fraction of the labeled data.

Implications for Edge AL: Using Fashion-MNIST allows us to profile the *energy consumption* and *latency* of the AL loop on the edge device without the noise of processing HD images. We can measure exactly how many milliwatts are consumed during the "query" phase versus the "training" phase, establishing a baseline for the hardware's capabilities.

3.2 CIFAR-10: Validating RGB Processing and Robustness

Dataset Rationale: CIFAR-10 serves as the next step up in complexity. It contains 60,000 color images (32x32 pixels) across 10 classes (Airplane, Automobile, Bird, Cat, Deer, Dog, Frog, Horse, Ship, Truck).
Why It Fits the "Sanity Check":

- **Color Channels:** Unlike Fashion-MNIST, CIFAR-10 requires the model to learn from RGB information. This is critical for plant disease detection, where color (chlorosis, necrosis) is a primary diagnostic feature. Validating the pipeline on CIFAR-10 ensures that the edge model's input layers and feature extractors are correctly handling multi-channel data.
- **Intra-Class Variance:** The visual diversity within a class in CIFAR-10 (e.g., a "Dog" can be a Dalmatian or a Poodle, facing left or right) is significantly higher than in Fashion-MNIST. This tests the AL model's ability to generalize features rather than just memorizing templates.

- **Adversarial Sensitivity:** Recent research indicates that models trained on CIFAR-10 are sensitive to adversarial perturbations and noise. This makes it an excellent proxy for testing how the AL algorithm handles "blurry" or "low-contrast" images, which are common in field agriculture. If the AL strategy gets confused by a blurry frog, it will definitely get confused by a wind-blown leaf.

Table 1: Comparative Analysis of "Toy" Verification Datasets

Feature	Fashion-MNIST	CIFAR-10	Relevance to Research Pivot
Resolution	28x28 (Grayscale)	32x32 (RGB)	Validates input processing logic.
Class Balance	Perfect (10x6000)	Perfect (10x6000)	Isolates query logic from imbalance issues.
Background	None (Black)	Varied (Natural)	Tests background suppression capabilities.
Edge Train Time	< 5 Minutes	~20 Minutes	Allows for high-frequency iteration/debugging.
Success Metric	>90% Acc @ 10% Data	>80% Acc @ 20% Data	Benchmarks for AL efficiency.

4. Phase II: Feature Learning via Controlled Agricultural Datasets

Once the algorithmic pipeline is validated on toy data, we must transition to the agricultural domain. However, we must resist the temptation to jump back into the chaos of field data. Instead, we use "clean" agricultural datasets to teach the edge model the fundamental visual vocabulary of plants—leaves, stems, spots, and lesions—without the interference of environmental noise. This phase addresses the "Cold Start" problem: ensuring the model has learned relevant features *before* it begins the Active Learning loop.

4.1 PlantVillage: The "Gold Standard" for Feature Extraction

Dataset Rationale: PlantVillage is the most widely cited dataset in plant disease research, comprising over 54,000 images of healthy and diseased leaves across 38 categories (species-disease pairs). The defining characteristic of this dataset is that all images were captured in a laboratory setting with uniform gray or black backgrounds, removing occlusion and lighting variability.

Strategic Value for Edge AL:

- **Transfer Learning Source:** The most effective way to improve AL performance on edge devices is to start with a "knowledgeable" model. Instead of initializing our edge model with weights from ImageNet (which is trained on dogs, cars, and consumer objects), we can pre-train the model on PlantVillage. This forces the convolutional filters to specialize in agricultural textures (e.g., fungal spores, bacterial wilt marks). When we eventually deploy this model to the field, it will already "know" what a leaf looks like, reducing the number of AL queries needed to reach convergence.
- **Fine-Grained Classification Test:** PlantVillage contains visually similar diseases (e.g.,

Tomato Early Blight vs. *Tomato Late Blight*). Distinguishing between these requires the model to focus on subtle textural differences. If our AL query strategy works well here, successfully selecting the "hard" cases that lie on the decision boundary between these similar classes, we have validated the model's precision capabilities.
- **Background Invariance:** By using a dataset with no background noise, we eliminate the risk of the model learning spurious correlations (e.g., associating "grass" with "healthy" and "soil" with "diseased"). This provides a clean baseline for the model's classification head.

4.2 Fruits-360: Stress-Testing Scalability and Imbalance

Dataset Rationale: The Fruits-360 dataset contains over 90,000 images of 131 different classes of fruits and vegetables (apples, bananas, cherries, etc.). The images are captured on a pure white background as the fruit is rotated 360 degrees.
Strategic Value for Edge AL:
- **Class Imbalance Simulation:** Real-world agricultural datasets are notoriously imbalanced—a farmer sees thousands of healthy plants for every one diseased plant. We can artificially subsample Fruits-360 to create severe class imbalances (e.g., 5,000 images of "Red Apple" vs. 50 images of "Rainier Cherry"). We can then test if our Active Learning strategy (e.g., Weighted Entropy) is capable of "hunting" for the rare minority class. If the AL loop ignores the rare cherries, we know our query function needs to be adjusted (e.g., by adding a density-weighting term) before we attempt field deployment.
- **Output Layer Scalability:** With 131 classes, Fruits-360 tests the scalability of the edge model's final classification layer. On an edge device, a large output layer consumes significant memory and compute. Benchmarking AL on this dataset allows us to optimize the "Last Layer" training strategies, such as only updating the weights of the queried classes to save energy.

5. Phase III: The "Sweet Spot" – High-Quality Field Datasets

This phase represents the critical bridge between laboratory theory and field reality. We transition to datasets that were captured in the *wild* (real-world conditions) but possess *high-quality, verified annotations*. These datasets introduce the necessary environmental complexity (lighting, weather, occlusion) but maintain the "Oracle Integrity" required for Active Learning to function correctly.

5.1 WGISD: Wine Grape Instance Segmentation Dataset

Dataset Rationale: The WGISD, produced by Embrapa, provides 300 high-resolution images of wine grapes in field conditions. It supports 5 classes (Chardonnay, Cabernet Franc, Cabernet Sauvignon, Sauvignon Blanc, Syrah) and includes annotations for both **Object Detection** (Bounding Boxes) and **Instance Segmentation** (Binary Masks).
Why It Is Superior to PlantDoc for Verification:
- **Verified Ground Truth:** Unlike PlantDoc, the annotations in WGISD were supervised by domain experts (viticulturists) and are intended for precision robotics. This reliability ensures that if the AL model is confused, it is due to the image features, not label noise.

- **Occlusion and Lighting:** The grapes are naturally occluded by leaves and branches, and the images feature variable outdoor lighting. This introduces "valid" noise—the kind of noise we *want* the AL model to learn to handle. It allows us to test "Aleatoric Uncertainty" metrics: does the model know when it can't see the grape?
- **Multi-Task Learning:** WGISD allows us to benchmark AL for different tasks. We can start with simple **Detection** (finding the cluster) using a YOLO architecture, and then advance to **Segmentation** (masking the berries). Segmentation labeling is expensive (costly oracle), making it an ideal candidate for proving the ROI of Active Learning—if we can reduce segmentation labeling by 50%, the economic argument for the system is cemented.

Table 2: WGISD Class Distribution and Complexity

Variety	Images	Boxed Clusters	Masked Clusters	Complexity Factors
Chardonnay	65	840	308	Green skin (low contrast with leaves)
Cabernet Franc	65	1,069	513	Purple skin (high contrast)
Cabernet Sauv.	57	643	306	Dense clusters, heavy occlusion
Sauvignon Blanc	65	1,316	608	High cluster count per image
Syrah	48	563	285	Variable lighting conditions
Total	**300**	**4,431**	**2,020**	**Ideal for "Small Data" AL**

5.2 MinneApple: Dense Object Counting and Localization

Dataset Rationale: The MinneApple dataset focuses on apple detection and segmentation in orchard environments. It contains 1,000 images with over 41,000 annotated object instances.
Strategic Value for Edge AL:

- **Density Challenges:** Unlike WGISD, where clusters are distinct, MinneApple images are dense, often containing 50-100 small apples per image. This challenges the **Object Detection** heads of lightweight models (e.g., YOLO-Nano), which often struggle with small objects due to aggressive downsampling.
- **Patch-Based Active Learning:** Labeling an entire image with 100 apples is incredibly time-consuming and expensive. MinneApple allows us to experiment with **Region-Based AL** strategies. Instead of querying the whole image, can the model query just a specific 256x256 "patch" where it is uncertain? This "Fine-Grained Querying" is a cutting-edge technique in Edge AI that maximizes the information gain per pixel transmitted.
- **Counting Logic:** This dataset allows us to test AL for **Regression/Counting** tasks. The uncertainty metric here isn't "What class is this?" but "How many apples are here?" This requires a different mathematical formulation (e.g., variance in count estimation across dropout passes), adding depth to the research.

6. Edge Computing Architecture and Implementation

The shift to easier datasets allows us to focus on optimizing the system architecture, ensuring that the software stack is robust enough to handle field deployment eventually.

6.1 Distributed Active Learning (The "Fog" Hierarchy)

Processing every unlabeled sample on a constrained edge device (e.g., Raspberry Pi 4, Jetson Nano) is often unfeasible due to latency. We propose implementing a **Distributed Active Learning** architecture.

- **The Edge Node (Student):** Runs a highly quantized, lightweight model (e.g., MobileNetV3-Small). Its job is to perform a rapid "first pass" on the incoming video stream. It calculates a rough uncertainty score using efficient metrics (e.g., Softmax Entropy).
- **The Fog/Cloud Node (Teacher):** Only the samples deemed "highly uncertain" by the Edge Node are transmitted to the Fog Node. The Fog Node, possessing more compute power, runs a larger "Teacher" model (e.g., ResNet-50 or EfficientNet) to verify the sample's informativeness.
- **Benefit:** This hierarchical filtering significantly reduces the bandwidth required for data transmission—a critical KPI for agricultural IoT. We can simulate this setup using WGISD by designating a subset of images as "local" (Edge) and the rest as "remote" (Cloud).

6.2 Data-Centric Optimization: Pruning and Coresets

Recent research in Edge AI emphasizes **Data-Centric AI**—optimizing the dataset rather than just the model.

- **Dataset Pruning:** Using the "Warm-up" phase of training (on simple datasets like Fashion-MNIST), we can calculate the "forgetting score" of each sample. Samples that are learned once and never forgotten are "easy" and can be permanently pruned from the training set. This reduces the storage footprint on the edge device.
- **Coreset Selection:** We will implement **Coreset** algorithms (e.g., k-Center Greedy) to select a subset of data that geometrically covers the feature space. This allows the edge model to train on a representative 10% of the data while achieving performance comparable to training on the full set, saving energy and time.

6.3 Hardware-Specific Considerations

- **Quantization Effects:** We must test how 8-bit integer quantization (INT8), which is standard for accelerating inference on Edge TPUs (Coral) and DSPs, affects the uncertainty estimates. Does a quantized model become "overconfident," breaking the AL strategy? We can test this by comparing AL curves of float32 vs. INT8 models on the Fashion-MNIST and WGISD datasets.
- **Thermal Constraints:** Running continuous training loops on edge devices generates heat. We will implement "Duty Cycling" in our AL protocol, allowing the device to cool down between query batches, and measuring the impact of this latency on the overall learning speed.

7. Proposed Roadmap and Milestones

Phase 1: The "Clean Room" Validation (Weeks 1-3)

- **Objective:** Validate the Active Learning code logic (Ranking, Querying, Retraining).
- **Dataset:** Fashion-MNIST (Classification).
- **Model:** Simple CNN (2 conv layers).
- **Success Criteria:** The AL strategy (e.g., Entropy) must outperform Random Sampling by a margin of >5% accuracy at a 10% data budget.
- **Deliverable:** A stable, bug-free Python pipeline compatible with PyTorch Mobile or TensorFlow Lite.

Phase 2: Feature Adaptation & Knowledge Transfer (Weeks 4-6)

- **Objective:** Train a robust Feature Extractor for agricultural textures.
- **Dataset:** PlantVillage (Classification).
- **Model:** MobileNetV3 (Edge-optimized).
- **Action:** Train on PlantVillage, freeze the backbone layers, and save the weights.
- **Insight:** This creates a "Agricultural Pre-trained Model" that solves the Cold Start problem for subsequent phases.

Phase 3: The Real-World Proxy (Weeks 7-10)

- **Objective:** Benchmark Object Detection AL in a reliable field setting.
- **Dataset:** WGISD (Detection/Segmentation).
- **Model:** YOLOv8-Nano (or YOLOv10-Nano for efficiency).
- **Experiment:** Compare "Localization Uncertainty" (box jitter) vs. "Classification Uncertainty" (class confusion).
- **Simulation:** Simulate edge constraints (memory limits, quantization) during the training loop.

Phase 4: Future Expansion (Weeks 11+)

- **Objective:** Return to high-complexity data with a proven pipeline.
- **Target:** FieldPlant or PlantWild.
- **Strategy:** Apply the WGISD-verified pipeline to these harder datasets. Use the "Fog Filtering" mechanism to handle the background clutter that originally caused the PlantDoc failure.

8. Conclusion and Supervisor’s Note

The initial failure with PlantDoc should be reframed not as a dead end, but as a necessary calibration of our experimental parameters. In experimental science, attempting to solve the hardest problem (noisy, cluttered, few-shot learning on edge) without first solving the constituent sub-problems (AL logic, feature extraction, robust detection) is a common pitfall.

By pivoting to this **Curriculum Verification Protocol**—starting with Fashion-MNIST/CIFAR-10 to prove the math, moving to PlantVillage to learn the features, and validating on WGISD/MinneApple to prove the application—we are building a defensible, rigorous thesis. We are essentially constructing a "Ladder of Complexity." This approach allows us to isolate

variables: if the model fails on Fashion-MNIST, it's a code bug. If it fails on PlantVillage, it's a capacity issue. If it fails on WGISD, it's an environmental adaptation issue.
This structured, step-by-step validation is the hallmark of senior-level research. It transforms a "failed experiment" into a "robust methodology for edge-based active learning deployment." Let us proceed with setting up the Fashion-MNIST benchmark immediately to secure our algorithmic foundations.
Primary References Included in Analysis:

- *PlantDoc Issues:*
- *Active Learning Benchmarks:*
- *Edge Computing Constraints:*
- *WGISD & MinneApple Specs:*
- *PlantVillage & Fruits-360 Utility:*

Works cited

1. Recent advances in plant disease detection: challenges and opportunities - PMC, https://pmc.ncbi.nlm.nih.gov/articles/PMC12570820/ 2. (PDF) PlantDoc: A Dataset for Visual Plant Disease Detection - ResearchGate, https://www.researchgate.net/publication/337531775_PlantDoc_A_Dataset_for_Visual_Plant_Disease_Detection 3. A Unified Self-Supervised Framework for Plant Disease Detection on Laboratory and In-Field Images - MDPI, https://www.mdpi.com/2079-9292/14/17/3410 4. Lightweight Edge Learning via Dataset Pruning - arXiv, https://arxiv.org/html/2602.00047v1 5. Plant disease classification in the wild using vision transformers and mixture of experts, https://pmc.ncbi.nlm.nih.gov/articles/PMC12213485/ 6. Distributed Active Learning Strategies on Edge Computing - ResearchGate, https://www.researchgate.net/publication/336254817_Distributed_Active_Learning_Strategies_on_Edge_Computing 7. State-of-the-Art Results with the Fashion-MNIST Dataset - MDPI, https://www.mdpi.com/2227-7390/12/20/3174 8. A Cross-Domain Benchmark for Active Learning - NeurIPS, https://proceedings.neurips.cc/paper_files/paper/2024/file/72de9826cfe582e175520f404eaad473-Paper-Datasets_and_Benchmarks_Track.pdf 9. Adversarial Machine Learning: Attacking and Safeguarding Image Datasets - arXiv, https://arxiv.org/html/2502.05203v1 10. A Active learning, in more detail, https://proceedings.neurips.cc/paper_files/paper/2023/file/1ed4723f12853cbd02aecb8160f5e0c9-Supplemental-Conference.pdf 11. Plant Leaf Disease Detection Using Deep Learning: A Multi-Dataset Approach - MDPI, https://www.mdpi.com/2571-8800/8/1/4 12. Fruits-360 dataset - Kaggle, https://www.kaggle.com/datasets/moltean/fruits 13. An Efficient Automatic Fruit-360 Image Identification and Recognition Using a Novel Modified Cascaded-ANFIS Algorithm - MDPI, https://www.mdpi.com/1424-8220/22/12/4401 14. thsant/wgisd: Embrapa Wine Grape Instance Segmentation Dataset - GitHub, https://github.com/thsant/wgisd 15. A Grape Dataset for Instance Segmentation and Maturity Estimation - MDPI, https://www.mdpi.com/2073-4395/13/8/1995 16. A Grape Dataset for Instance Segmentation and Maturity Estimation - ResearchGate, https://www.researchgate.net/publication/372728110_A_Grape_Dataset_for_Instance_Segmentation_and_Maturity_Estimation 17. MinneApple: A Benchmark Dataset for Apple Detection and Segmentation, https://conservancy.umn.edu/items/e1bb4015-e92a-4295-822c-d21d277ecfbd
18. MinneApple: A Benchmark Dataset for Apple Detection and Segmentation - ResearchGate, https://www.researchgate.net/publication/338495204_MinneApple_A_Benchmark_Dataset_for_

Apple_Detection_and_Segmentation 19. MinneApple: A Benchmark Dataset for Apple Detection and ..., https://rsn.umn.edu/projects/orchard-monitoring/minneapple 20. MinneApple: A Benchmark Dataset for Apple Detection and Segmentation - arXiv, https://arxiv.org/abs/1909.06441 21. Intelligent data analysis in edge computing with large language models: applications, challenges, and future directions - Frontiers, https://www.frontiersin.org/journals/computer-science/articles/10.3389/fcomp.2025.1538277/full 22. A Lightweight and Efficient Plant Disease Detection Method Integrating Knowledge Distillation and Dual-Scale Weighted Convolutions - MDPI, https://www.mdpi.com/1999-4893/18/7/433 23. FieldPlant: A dataset of field plant images for plant disease detection and classification with deep learning - IEEE Xplore, https://ieeexplore.ieee.org/iel7/6287639/6514899/10086516.pdf 24. Benchmarking In-the-Wild Multimodal Plant Disease Recognition and A Versatile Baseline, https://arxiv.org/html/2408.03120v1 25. Data-centric annotation analysis for plant disease detection: Strategy, consistency, and performance - Frontiers, https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2022.1037655/full 26. Unlabeled data selection for active learning in image classification - PMC, https://pmc.ncbi.nlm.nih.gov/articles/PMC10764761/ 27. Applications of Federated Learning; Taxonomy, Challenges, and Research Trends - MDPI, https://www.mdpi.com/2079-9292/11/4/670