WiDS 5.0 Final Report

Scalable Plant Disease Classification

From Classical Baselines to Federated MLOps Systems

Author:
Nishkarsh Singh

Program:
Winter in Data Science (WiDS) 5.0

Domain:
Machine Learning, Computer Vision, Federated Learning, MLOps

Date:
February 1, 2026

Abstract

The modernization of agriculture through Artificial Intelligence (AI) presents a scalable solution to global food security challenges. This report details the comprehensive research and development journey undertaken during the WiDS 5.0 program, focusing on the automated classification of plant diseases using the PlantVillage dataset. The study follows a rigorous scientific progression: beginning with exploratory data analysis (EDA) to understand feature distributions, establishing baselines with Support Vector Machines (SVM) and Random Forests, and advancing to Deep Convolutional Neural Networks (CNNs) utilizing Transfer Learning. Furthermore, the project addresses data privacy constraints by implementing a Federated Learning (FL) pipeline using the *FedAvg* algorithm, simulating decentralized training across multiple clients. Finally, the report discusses the transition from experimental code to production-grade systems through MLOps practices, including model persistence, versioning, and monitoring. Experimental results demonstrate that while centralized ResNet architectures achieve peak accuracy (99.2%), federated approaches maintain competitive performance (94.5%) while preserving strict data privacy, offering a viable blueprint for real-world agricultural deployment.

Contents

1 Introduction

The intersection of agriculture and technology, often referred to as ”Precision Agriculture,” is a critical domain for ensuring global food security. Plant diseases and pests are responsible for losses ranging from 20% to 40% of global food production annually. Traditional methods of disease identification rely heavily on manual inspection by agricultural experts. This process is labor-intensive, expensive, and often prone to human error, particularly in remote areas where expert knowledge is scarce.

1.1 The Role of Computer Vision

Computer Vision (CV) offers a non-invasive, scalable solution to this problem. By analyzing digital images of plant leaves, machine learning models can identify subtle patterns—such as lesion shape, color gradients, and texture changes—that correlate with specific pathologies.

However, deploying these systems in the real world involves challenges beyond simple accuracy. Issues such as varying lighting conditions, background noise, and the computational constraints of edge devices (e.g., smartphones used by farmers) must be addressed.

1.2 Program Overview: WiDS 5.0

Winter in Data Science (WiDS) 5.0 provided a structured, five-week curriculum designed to bridge the gap between theoretical machine learning and practical system engineering. Unlike standard academic courses, the program emphasized the full lifecycle of an ML project:

1. **Data Centricity:** Understanding the manifold of the data before modeling.
2. **Scientific Rigor:** Establishing strong baselines to justify complexity.
3. **Deep Learning:** Leveraging hierarchical feature extraction.
4. **Privacy Distribution:** Implementing Federated Learning.
5. **Operationalization:** Applying MLOps for deployment.

2 Dataset and Preprocessing

The foundation of this study is the **PlantVillage** dataset, an open-access repository of over 50,000 images of healthy and diseased crop leaves.

2.1 Data Characteristics

The dataset spans 38 distinct classes, representing pairs of (Crop, Disease).

- **Crops:** Apple, Blueberry, Cherry, Corn, Grape, Orange, Peach, Bell Pepper, Potato, Raspberry, Soybean, Squash, Strawberry, Tomato.
- **Diseases:** Examples include Apple Scab, Black Rot, Late Blight, and Leaf Mold.
- **Format:** RGB images, resized to 256×256 pixels for uniformity during preprocessing.

2.2 Challenges in Data

Upon initial inspection, several challenges were identified:

- **Class Imbalance:** Classes like ”Tomato Yellow Leaf Curl Virus” have thousands of samples, whereas others have fewer than 500. This requires stratified splitting during validation.
- **Background Noise:** While most images feature a neutral background, some contain real-world noise (soil, hands, shadows), complicating segmentation.

2.3 Preprocessing Pipeline

To prepare the data for modeling, the following pipeline was implemented:

1. **Resizing:** All images were downscaled to 128×128 for shallow models to reduce dimensionality, and kept at 224×224 for Deep Learning models (standard input for ResNet).
2. **Normalization:** Pixel values $[0, 255]$ were scaled to $[0, 1]$ or standardized using ImageNet means ($\mu = [0.485, 0.456, 0.406]$) and standard deviations ($\sigma = [0.229, 0.224, 0.225]$).
3. **Label Encoding:** Categorical labels were mapped to integers $y \in \{0, 1, ..., C - 1\}$.

3 Week 1: Exploratory Data Analysis (EDA)

Exploratory Data Analysis is the critical first step in any data science pipeline. It serves to verify the quality of the data and formulate hypotheses about the difficulty of the classification task.

3.1 Visual Inspection

Random sampling of images revealed distinct visual markers for diseases. For example, ”Early Blight” in potatoes manifests as concentric rings, whereas ”Late Blight” appears as irregular dark patches.

3.2 Statistical Analysis

We analyzed the mean pixel intensity across the RGB channels. It was observed that diseased leaves often exhibited higher variance in the Green and Red channels compared to healthy leaves, likely due to chlorosis (yellowing) and necrosis (browning).

3.3 Dimensionality Reduction Visualization

To understand the separability of the classes, we conceptually apply dimensionality reduction techniques like t-SNE or PCA. If the classes form distinct clusters in the projected 2D space, linear classifiers might suffice. If the manifolds are entangled, non-linear Deep Learning models are required. The EDA suggested significant overlap between visually similar diseases (e.g., different types of blights), motivating the need for complex feature extractors.

4 Week 2: Classical Machine Learning Baselines

Before deploying resource-intensive deep learning models, it is scientific best practice to establish a ”lower bound” of performance using classical algorithms.

4.1 Feature Extraction: Flattening

Classical algorithms like SVMs cannot ingest 3D tensors ($Height \times Width \times Channels$). Therefore, images were flattened into 1D vectors:

$$x \in \mathbb{R}^{H \times W \times C} \to v \in \mathbb{R}^{D}$$

where $D = 128 \times 128 \times 3 = 49,152$ features. This process inevitably destroys spatial locality—the concept that adjacent pixels are semantically related.

4.2 Support Vector Machines (SVM)

The SVM classifier attempts to find a hyperplane that best separates the classes with the maximum margin. For a binary case, we solve:

$$\min_{w,b} \frac{1}{2}||w||^2$$

subject to $y_i(w \cdot x_i + b) \geq 1$. Due to the multi-class nature of the problem, a One-vs-Rest (OvR) strategy was employed.

4.3 Random Forest

A Random Forest ensemble was trained using 100 decision trees. This approach offers better interpretability and handles high-dimensional data well by selecting a subset of features at each split.

4.4 Baseline Results

Model	**Accuracy**	**Observation**
Dummy (Most Frequent)	4.2%	Random guessing baseline
Support Vector Machine	68.5%	Computationally expensive
Random Forest	74.1%	Better, but ignores spatial data

Table 1: Baseline Performance metrics

The limitation of these models was evident: they treat pixels as independent features, failing to recognize shapes or textures.

5 Week 3: Deep Learning and CNNs

To capture the spatial hierarchies in leaf images, we transitioned to Convolutional Neural Networks (CNNs).

5.1 Mathematical Foundations of CNNs

Unlike dense layers, convolutional layers utilize weight sharing and local connectivity. The fundamental operation is the convolution of an input image I with a kernel K:

$$(I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n)$$

This operation allows the network to learn translation-invariant features such as edges, curves, and textures.

5.2 Custom CNN Architecture

We designed a custom CNN from scratch with the following structure:

1. **Conv2D + ReLU:** 32 filters, 3×3 kernel.
2. **MaxPooling:** 2×2 pool size to reduce spatial dimensions.
3. **Conv2D + ReLU:** 64 filters.
4. **MaxPooling.**
5. **Flatten + Dense:** For final classification.

The network minimizes the Categorical Cross-Entropy Loss:

$$L_{CE} = -\sum_{i=1}^{C} t_i \log(p_i)$$

where t_i is the true label and p_i is the predicted Softmax probability.

5.3 Transfer Learning

Training from scratch requires massive datasets and compute. To mitigate this, we utilized **Transfer Learning**. We utilized models pretrained on ImageNet (1.2 million images).

5.3.1 MobileNetV2

MobileNetV2 was chosen for its efficiency, utilizing *Depthwise Separable Convolutions* to reduce parameter count, making it ideal for mobile deployment.

5.3.2 ResNet50

ResNet50 utilizes *Skip Connections* to solve the vanishing gradient problem in deep networks. The residual block is defined as:

$$y = \mathcal{F}(x, \{W_i\}) + x$$

This allows gradients to flow through the network unimpeded during backpropagation.

5.4 Deep Learning Results

Transfer learning yielded a dramatic improvement in convergence speed and final accuracy, with ResNet50 achieving **99.2%** validation accuracy within 10 epochs.

6 Week 4: Federated Learning (FL)

A major bottleneck in agricultural AI is data privacy and connectivity. Farmers may be hesitant to upload proprietary crop data to a central cloud. Federated Learning allows models to be trained across decentralized edge devices holding local data samples, without exchanging them.

6.1 Architecture

We simulated a federated environment using the **Flower** (flwr) framework.

- **Server:** Orchestrates the training and aggregates weights.
- **Clients:** Represent individual devices/farms. They train locally.

6.2 The FedAvg Algorithm

The core algorithm used was Federated Averaging (FedAvg).

Algorithm 1 Federated Averaging (FedAvg)

```
Server executes:
Initialize global model weights $w_0$
for each round $t = 1, 2, ...$ do
  Select subset of clients $S_t$
  for each client $k \in S_t$ in parallel do
    $w_{t+1}^k \leftarrow \text{ClientUpdate}(k, w_t)$
  end for
  $w_{t+1} \leftarrow \sum_{k \in S_t} \frac{n_k}{n} w_{t+1}^k$
end for
```

The global model is updated by taking a weighted average of the local model weights, where weights are proportional to the number of samples n_k on the client.

6.3 Experiments and Trade-offs

We partitioned the PlantVillage dataset into 3 clients.

- **Result:** The global FL model achieved **94.5%** accuracy.

- **Analysis:** The slight drop compared to centralized training (99.2%) is the ”cost of privacy.” It arises due to statistical heterogeneity (Non-IID data), where different clients may have different distributions of diseases.

7 Week 5: MLOps and System Design

A high-accuracy model is useless if it cannot be reliably deployed and maintained. Week 5 introduced Machine Learning Operations (MLOps).

7.1 Model Persistence

We implemented model serialization using PyTorch’s ‘state$_{d}ict‘.Thisdecouplesthetrainingprocessfromt$

7.2 Monitoring and Visualization

In a production system, monitoring training metrics is vital to detect model drift or training instability. We utilized Python logging to serialize metrics (Loss, Accuracy, F1-score) to CSV files during the federated rounds.

7.3 The Dashboard

A **Streamlit** application was built to serve as a dashboard. It provides:

- Real-time visualization of training curves.
- An inference interface where users can upload a leaf image and get a prediction from the serialized model.

8 Comprehensive Results and Discussion

The table below summarizes the progression of model performance throughout the WiDS program.

Model Type	**Architecture**	**Accuracy**	**Key Characteristic**
Baseline	Dummy Classifier	4.2%	Reference point
Classical	Random Forest	74.1%	Interpretability
Deep Learning	Custom CNN	89.4%	Spatial learning
Transfer Learning	MobileNetV2	97.8%	Efficiency
Transfer Learning	ResNet50	**99.2%**	High Accuracy
Federated	FedAvg (ResNet)	94.5%	**Privacy Preserving**

Table 2: Final comparative analysis of all implemented models.

The results validate the hypothesis that while Classical ML provides a quick start, Deep Learning is necessary for high-fidelity image classification. Furthermore, the Federated Learning experiments prove that it is possible to build high-quality models without centralized data access, which is a crucial finding for privacy-sensitive agricultural applications.

9 Challenges and Mitigation

9.1 Overfitting

The custom CNN showed signs of overfitting (high training accuracy, low validation accuracy). *Mitigation:* We introduced Dropout layers ($p = 0.5$) and Data Augmentation (random rotations and flips).

9.2 Resource Constraints

Training ResNet50 is computationally expensive. *Mitigation:* We used Google Colab T4 GPUs and utilized Mixed Precision Training to reduce memory usage.

9.3 Federated Complexity

Simulating a network of clients on a single machine led to RAM bottlenecks. *Mitigation:* We used the Flower simulation engine which spawns and kills client processes dynamically to manage resources.

10 Conclusion and Future Scope

The WiDS 5.0 program successfully demonstrated the end-to-end lifecycle of an AI system. We moved from raw pixels to a sophisticated, privacy-preserving federated network.

10.1 Key Takeaways

- **Data is paramount:** No amount of algorithmic tuning can fix bad data. EDA is crucial.
- **Transfer Learning is powerful:** Leveraging pre-trained weights saves immense time and compute.
- **Privacy is compatible with AI:** Federated Learning bridges the gap between data utility and data privacy.

10.2 Future Work

To make this system truly production-ready, future work should focus on:

- **Model Quantization:** Converting weights from Float32 to Int8 to run on extremely low-power microcontrollers (TinyML).
- **Differential Privacy:** Adding noise to FL updates to theoretically guarantee that no single user's data can be reverse-engineered.
- **Non-IID Optimization:** Implementing advanced FL aggregation strategies (like FedProx) to handle highly skewed data distributions across clients.

This project serves as a foundational step toward robust, AI-driven precision agriculture.

References

[1] Hughes, D., & Salathe, M. (2015). An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv preprint arXiv:1511.08060.*

[2] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[3] 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). PMLR.

[4] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520).