Leaf by Leaf: How We Built an AI Model to Predict Plant Diseases

According to the Food and Agriculture Organization of the United Nations (FAO) the global economy loses over $220 billion annually to plant diseases, yet the first crucial warnings are often hidden on the crops' own leaves. Did you know that? Link

Crop diseases pose a significant threat to global food security and farmers' livelihoods. Traditional methods of detecting these diseases, like manual inspection, are often time-consuming, prone to errors, and can be too late to prevent substantial crop loss. This is a major problem for many farmers: how to identify infections in leaves at early stages to enable timely interventions.

This blog talks about a solution that we've come up with that can unlock unhealthy crops by transforming image and by leveraging the power of machine learning, we show how it is possible to conduct automated ***image-based disease detection***. Our project aims to classify images of plants to detect diseases, providing a user-friendly interface where individuals can directly interact with the model without needing to run code. In doing so, our project aligns with *Sustainable Development Goal (SDG) 2 (Zero Hunger)* by boosting food production efficiency and **SDG 12** (Responsible Consumption and Production) by minimizing waste.

Current Situation of the Problem or Phenomenon

Crop diseases are a silent threat, continually undermining agricultural yields, jeopardizing global food security, and directly impacting the livelihoods of farmers worldwide. This widespread challenge, the sheer volume of crops, and the subtle initial signs of infection create an almost insurmountable problem for conventional methods.

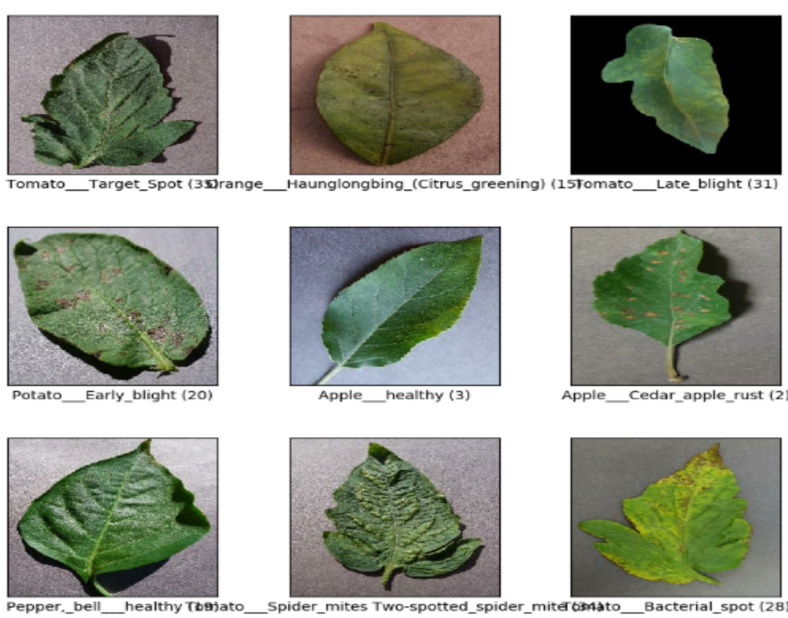
Whether you're an industry professional, a researcher, or just curious about machine learning, we hope to provide a new outlook on yet another way in which machine learning can contribute towards a more prosperous future for all.

1. The Process: Methodologies

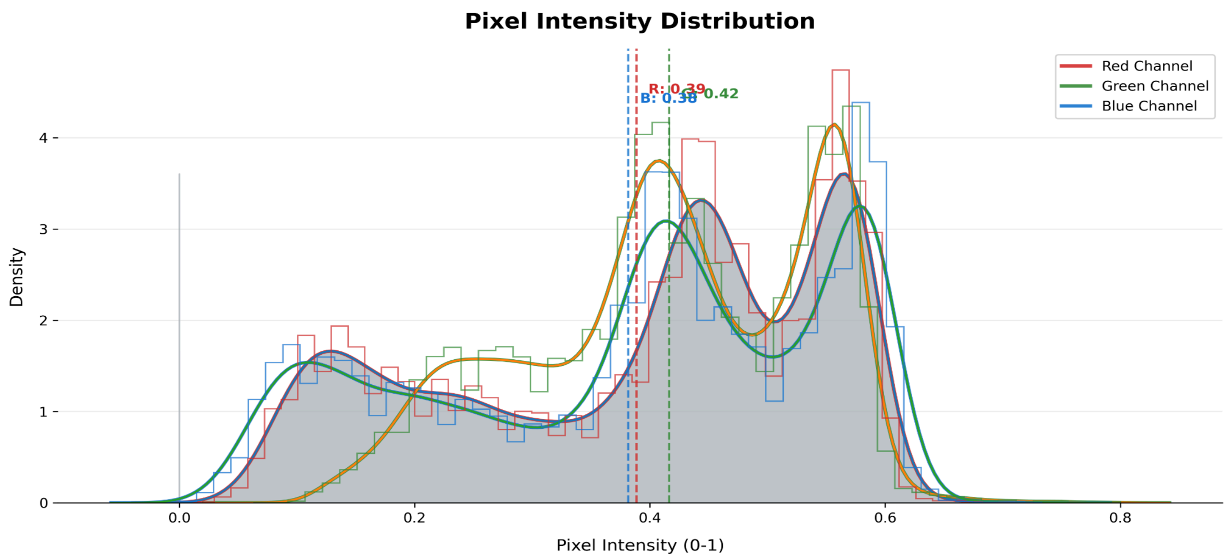
i. Methodology

Our project develops an AI model to accurately identify plant diseases from leaf images using deep learning. We primarily use **Convolutional Neural Networks (CNNs)** for their effectiveness in processing complex image patterns.

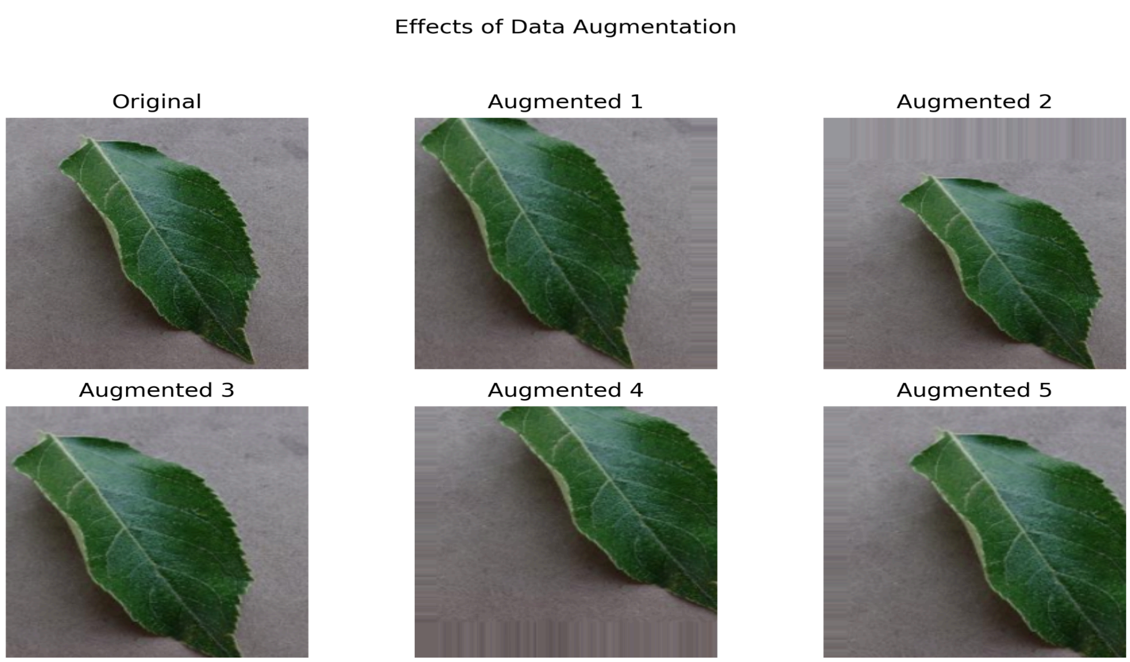
Dataset: We used a subset of the PlantVillage dataset, comprising around 87,000 images of healthy and unhealthy plant leaves. These images are categorized into 38 disease classes and organized into training and validation sets in JPEG/PNG format.



Preprocessing: Critical preprocessing steps include resizing images to 224x224 pixels and normalizing pixel values (0-255 to 0-1 range). This prepares the data for efficient model training. Labels are automatically one-hot encoded for multi-class classification. Which can be seen using this distribution plot of one sample image.



Data Augmentation: To make the model robust and prevent overfitting, we apply data augmentation. This involves random transformations like shear, zoom, and shifts in width and height to the training images, artificially expanding our dataset.



Model Architecture and Fine-Tuning: We chose MobileNet for its balance of accuracy and computational efficiency, ideal for various deployment environments. Using transfer learning, we initialized *MobileNet* with weights pre-trained on ImageNet. We excluded *MobileNet's* original classification layers, adding a custom head suited for our ***38 disease*** classes. This custom head includes a `GlobalAveragePooling2D` layer, a `Dropout(0.2)` layer for regularization, and a final `Dense` layer with `softmax` activation. Initially, we froze the pre-trained layers' weights to leverage ImageNet features effectively.

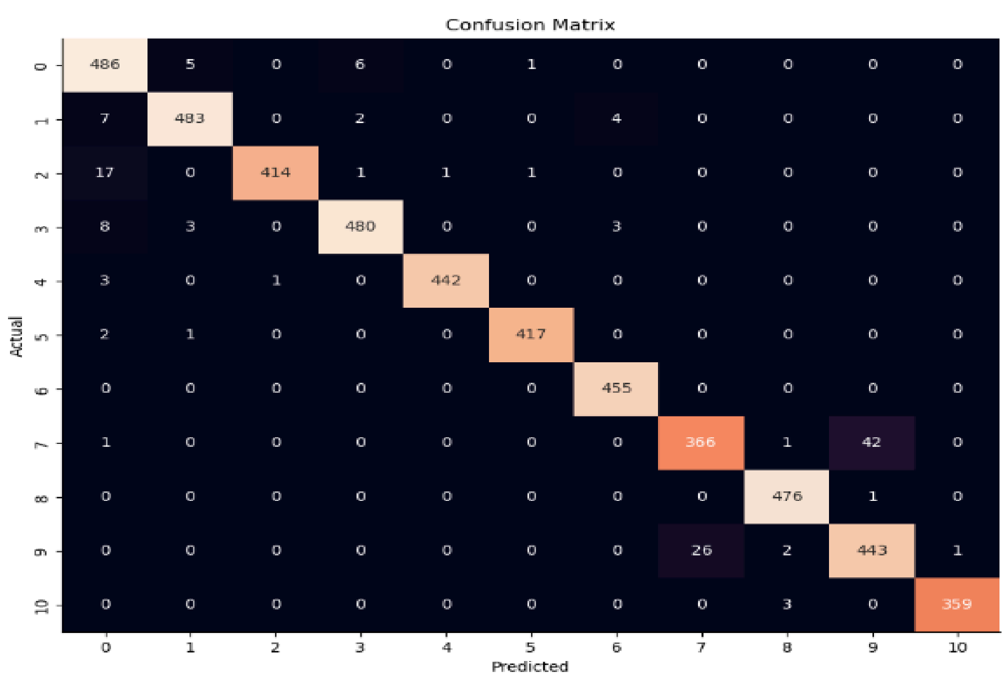
Hyperparameters: The Adam optimizer, with default parameters, was used. The model was compiled with `CategoricalCrossentropy` loss and `CategoricalAccuracy`/`accuracy` metrics. Training ran for 10 epochs, with `steps_per_epoch=150` and `validation_steps=100`.

ii. Results and Findings

Model performance was evaluated on the test dataset using a confusion matrix and plots of training/validation loss and accuracy.

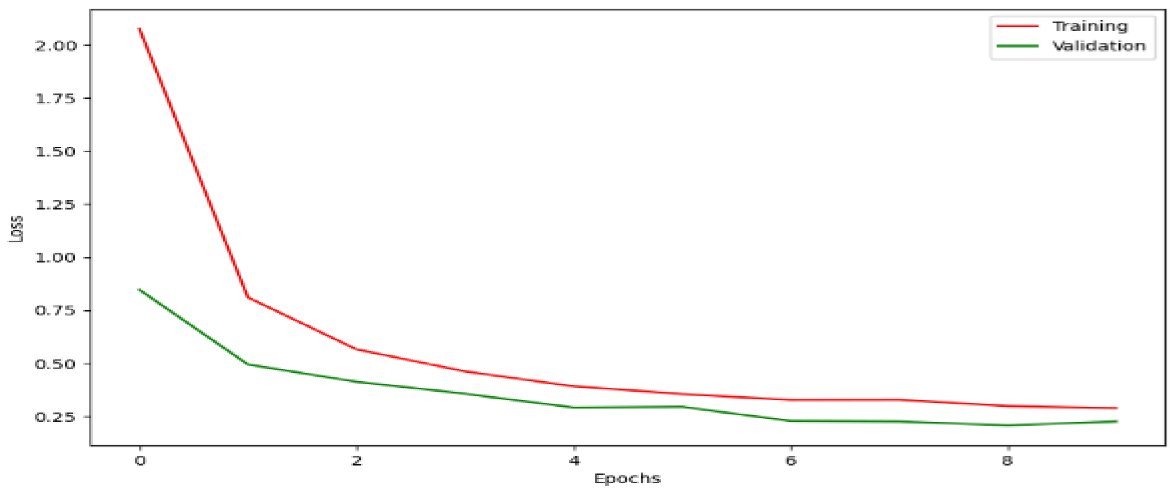
Confusion Matrix:

The diagonal dominance (e.g., 486, 483, 414) shows the model accurately predicts most classes (0-10), with high precision for common disease categories, critical for reliable disease detection



Training and Validation Loss

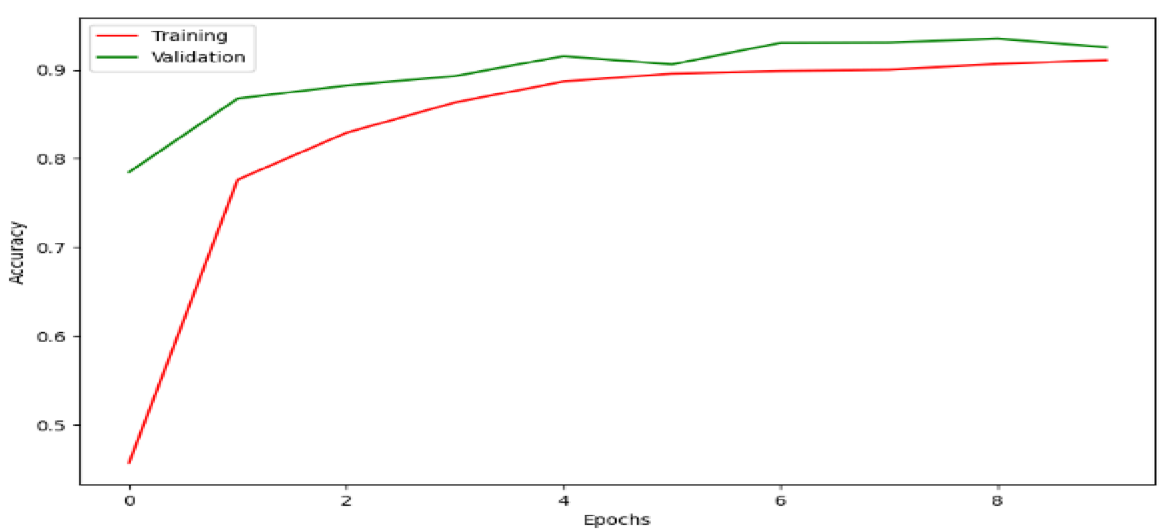
This plot shows the decrease in training and validation loss over epochs. The training and validation loss both decrease steadily over 8 epochs, indicating effective model learning, with training loss starting at ~2.0 and stabilizing around 0.25, suggesting good convergence.



Training and Validation Accuracy:

Both training and validation accuracy increase consistently, reaching ~0.9 by epoch 8, demonstrating the model's strong ability to classify disease-related images with high reliability.

The close alignment of training and validation accuracy suggests robust model performance.



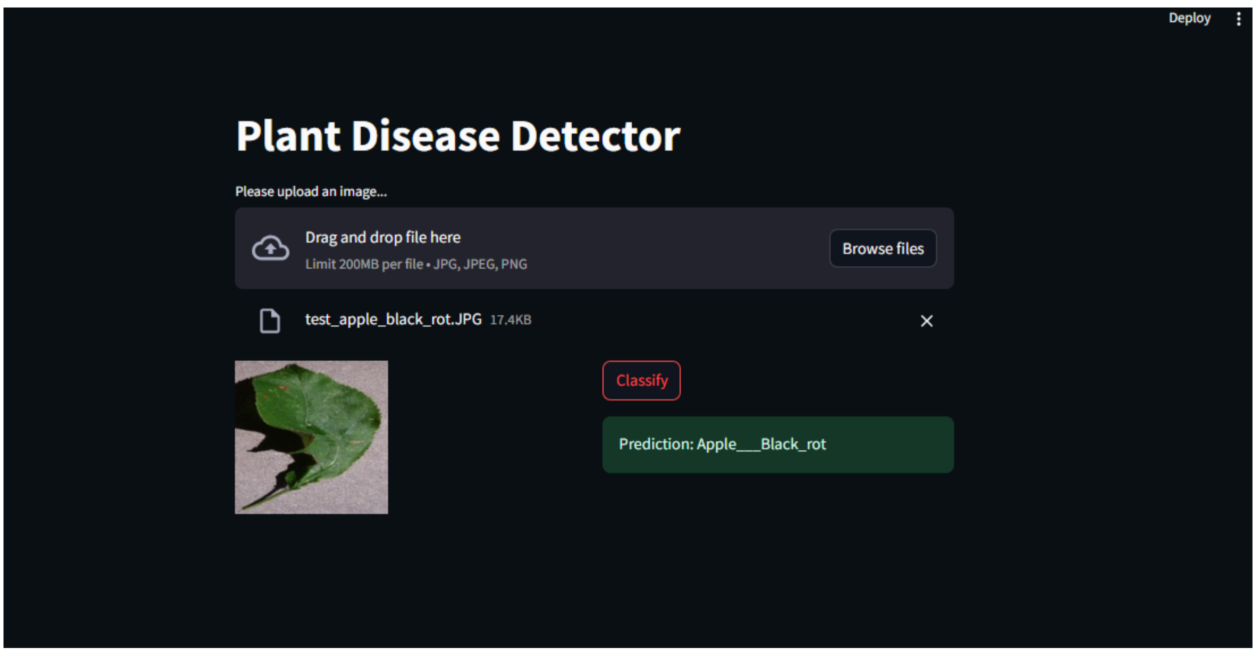
3. Deployment – Implementation with Streamlit

The implementation involves loading and preprocessing data, model definition, compilation, training, and after saving the model for deployment.

The trained model is saved in ***HDF5(.h5)*** format for storage and later loading. We deploy the model locally via a **Streamlit** web application.

How to Use:

Users upload an image, and the app performs preprocessing, model prediction, and result display. This provides a user-friendly interface without requiring code execution. The Streamlit app loads the serialized Keras model and class mappings, processes uploaded images, and displays predictions. This is an on-premises solution.



4. Future Possibility

Our project successfully demonstrates the effectiveness of MobileNet with transfer learning and data augmentation for plant disease detection. High accuracy on the PlantVillage dataset indicates strong classification capability. Learning curves show effective training and generalization.

However, limitations exist. The PlantVillage dataset may not fully capture real-world variability (lighting, backgrounds), potentially affecting generalization.

Future improvements could include:

- **Data Diversity:** Add more varied training data from diverse field conditions and regions to enhance generalization.
- **Real-time Capabilities:** Optimize for edge deployment (e.g., TensorFlow Lite) for offline use and faster inference.
- **Robust Monitoring:** Implement comprehensive monitoring (performance metrics, logging inputs/predictions, alerting) for production.
- **API Development:** Create a REST API (e.g., Flask, FastAPI) for programmatic access.
- **Enhanced Security:** Implement strong security measures for public deployments (authentication, HTTPS, input validation).
- **Model Drift Detection:** Monitor for changes in data properties to ensure timely model retraining.

6. Conclusion – Implications and Inspirations

We developed an AI-driven system for plant disease detection using deep learning and image recognition. Our project uses the MobileNet architecture, trained on the PlantVillage dataset, to accurately classify healthy and diseased plant leaves. A user-friendly Streamlit web application provides direct interaction and visual disease prediction, offering an efficient alternative to manual inspection.

Recommendations: What should people do in the future? Ideas or suggestions on a way to move forward The potential of AI in agriculture is immense. We recommend:

- **Scaling and Accessibility:** Deploy AI solutions on broader platforms (cloud services, specialized hosting) for wider access by farmers and agricultural officers.
- **Continuous Improvement:** Continuously retrain models with new, diverse data to adapt to evolving disease patterns. Model drift detection for this project is crucial.
- **Integration with IoT and Drone Technology:** Combine AI disease detection with drone-mounted imaging for broader field coverage and early warning systems, creating comprehensive monitoring solutions.
- **Community Engagement:** Engage with farmers and professionals to ensure tools are practical and intuitive, driving user adoption.
- **Open Source and Low-Cost Solutions:** Prioritize affordable or open-source deployment to benefit smallholder farmers globally.

5. References

- The persistent threat of emerging plant disease pandemics to global food security - PNAS, accessed April 10, 2025, hps://www.pnas.org/doi/10.1073/pnas.2022239118

- Plant Disease: A Growing Threat to Global Food Security - MDPI, accessed Apr 10, 2025, hps://www.mdpi.com/2073-4395/14/8/1615

- The Impact of Plant Diseases - Safefood, accessed April 10, 2025, hps://www.safefood.net/food-safety/news/impact-plant-diseases.

- Plant health and its effects on food safety and security in a One Health framework: four case studies - PubMed, accessed April 10, 2025, hps://pubmed.ncbi.nlm.nih.gov/33829143/

- Special Issue : The Impact of Plant Disease on Food Security - MDPI, accessed April 10, 2025, hps://www.mdpi.com/journal/agriculture/special_issues/plant_disease.