# Plant Disease Detection and their treatment recommendation

Roshan Kumar Mahto, Kshitij Sah ,Tarandeep Singh, Himanshu Kumar

Computer Vision Project Interim presentation

INDRAPRASTHA INSTITUTE *of*
INFORMATION TECHNOLOGY
**DELHI**

# Problem Statement

## Problem Statement

Farmers struggle with early disease detection, leading to crop losses and excessive pesticide use. Manual inspection is slow, error-prone, and requires expert knowledge. An AI-based solution is needed to address this issue.

## Scope

This project develops an AI-driven system to detect plant diseases from leaf images. It provides real-time diagnosis and treatment recommendations via a web platform and mobile app. The goal is to reduce pesticide misuse, improve crop health, and promote sustainable farming.

## Target Users

Farmers benefit from early disease detection and treatment suggestions. Researchers refine disease detection models. Government organizations support large-scale implementation, while agri-tech companies enhance precision farming.

# Related Work

Recent research has significantly advanced plant disease detection through the application of image processing and deep learning techniques. These methodologies have enhanced the accuracy and efficiency of identifying various plant diseases, thereby supporting sustainable agricultural practices.

**Image Processing Techniques:**

Traditional approaches have utilized image processing methods for disease detection, involving steps such as image acquisition, preprocessing, feature extraction, and classification. These techniques aim to identify visual symptoms of diseases on plant leaves, facilitating early intervention and management. However, challenges such as varying illumination conditions and complex backgrounds have limited their effectiveness.

**Deep Learning Approaches:**

The integration of deep learning models, particularly Convolutional Neural Networks (CNNs), has revolutionized plant disease detection. For instance, a study constructed a stepwise disease detection model using images of diseased and healthy plant pairs, employing a CNN algorithm consisting of five pre-trained models. This model achieved high accuracy (97.09%) in classifying crops and disease types, demonstrating the potential of deep learning in enhancing detection precision.

# Related Work

**Comparative Analyses:**

Comprehensive reviews have compared various detection and classification techniques, highlighting the progress made and areas requiring improvement. These analyses underscore the importance of selecting appropriate methodologies tailored to specific crop types and disease characteristics to achieve optimal results.

**Overall Impact:**

The adoption of advanced image processing and deep learning techniques has markedly improved the accuracy and efficiency of plant disease detection. These advancements enable early intervention, reducing crop losses and minimizing the unnecessary use of pesticides. Integrating these technologies into user-friendly applications empowers farmers with accessible tools for real-time disease monitoring, thereby promoting sustainable agricultural practices.

In conclusion, the convergence of image processing and deep learning methodologies represents a transformative shift towards more efficient and eco-friendly farming practices, contributing to enhanced crop management and agricultural productivity.

References:

https://journalofbigdata.springeropen.com/articles/10.1186/s40537-023-00863-9
https://ieeexplore.ieee.org/document/9399342
https://www.researchgate.net/publication/314436486_An_Overview_of_the_Research_on_Plant_Leaves_Disease_detection_using_Image_Processing_Techniques
https://www.nature.com/articles/s41598-023-34549-2

# Dataset description and Evaluation metrics

- **Primary Datasets:**
  - **New Plant Diseases Dataset (Augmented)** – 14 plant species with augmented images (lighting, rotation, zoom).   -> total 38 class

- **Additional Datasets:**
  - **Wheat-Plant-Diseases** – Images of various wheat disease conditions.   -> 3 class
    **Rice-Leaf-Diseases-Detection** – Dataset focused on detecting diseases in rice leaves. -> 4 class
  - **AugmentedDatasetOfWheatRice** – Enriched dataset combining wheat and rice disease images.

**Dataset Split Information**

- **Total Images: 82,895**
- **Train Set: 58,026 images**
- **Validation Set: 12,434 images**
- **Test Set: 12,435 images**

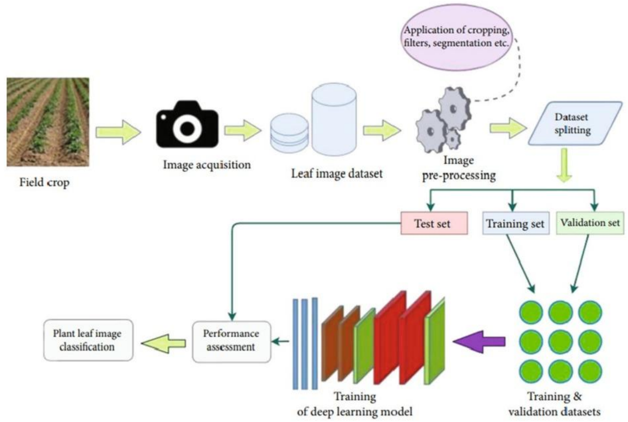
**Evaluation Metrics**

- **Accuracy** – Measures overall model performance.
- **Precision, Recall & F1-Score** – Essential for imbalanced datasets.
- **Confusion Matrix** – Helps analyze misclassifications.

# System, Baseline results and their analyses

We first resize each leaf image to 256×256 pixels, then apply CLAHE (clipLimit 2.0, tileGridSize 16×16) in LAB space to boost contrast moderately. Next, we generate a hybrid mask by combining a gentle HSV green filter with dilated Canny edges.. We crop to the largest leaf contour (area >500px), resize to the final target (e.g., 128×128), convert to a tensor, and normalize. This pipeline yields uniformly sized, well-contrasted leaves on clean white backgrounds while preserving subtle disease details.

| Model | Accuracy (%) | Runtime (s) | Memory Usage (MB) | Bandwidth (Mbps) | Remarks |
|---|---|---|---|---|---|
| Custom CNN | **96.27** | 160.4 | 460.6 | 2.84 | Highest accuracy, longest runtime |
| Pretrained ResNet-18 | 94.73 | 164.96 | 483.25 | 2.93 | High accuracy, slightly higher memory |
| Support Vector Machine | 92.8 | **61.72** | 81.55 | 2.23 | Shortest runtime, efficient |
| Random Forest | 85.53 | 216.41 | **1034.38** | **0.64** | Lowest accuracy, most memory efficient |



# System, Baseline results and their analyses

We experimented with multiple models for plant leaf detection, testing **different dataset combinations, and fine-tuning strategies** to optimize performance. Our results show a clear trade-off between accuracy and computational efficiency.

The **Custom CNN** achieved the highest accuracy (96.27%) but required more memory and longer processing time. The **SVM** provided a strong balance between accuracy (92.8%) and efficiency, making it a practical choice. The **Random Forest** was the most computationally efficient but had the lowest accuracy (85.53%).

Ultimately, the choice of model depends on deployment constraints—**CNNs excel in high-accuracy applications, while SVMs offer a solid balance between performance and efficiency**.

| Model | Accuracy | Precision (Macro Avg) | Recall (Macro Avg) | F1-score |
|---|---|---|---|---|
| Custom CNN | 96.27% | 96.49% | 96.27% | 96.15% |
| ResNet | 94.73% | 95.17% | 94.77% | 94.77% |
| SVM | 92.80% | 93.00% | 93.00% | 93.00% |
| Random Forest | 85.53% | 86.00% | 86.00% | 86.00% |

# System, Baseline results and their analyses

**We also encountered some problems, like wheat diseases classification failure**

**Overall Model Performance**

- Achieved **75.24% accuracy** across all plant disease categories.
- High precision and recall for most crop diseases.
- Strong classification results for **tomato, grape, orange, soybean, and apple diseases**.

Failure in Wheat Disease Classification and also Rice

- low Precision, Recall, and F1-Score for multiple wheat diseases:
    - Wheat Common Root Rot ,Wheat Black Rust,Wheat Yellow Rust, Wheat Septoria

- Causes:
    - unbalanced dataset and their are many similarity between corn, rice, corn.

**Final Result analysis:**

Our web and mobile application is  working and give me precise recommendations and disease classification

# Next Steps and a summary of individual contributions

**Current Progress**

We developed end to end project but some issue so in Future work we resolve this issue and these issue have following

**Contribution:**

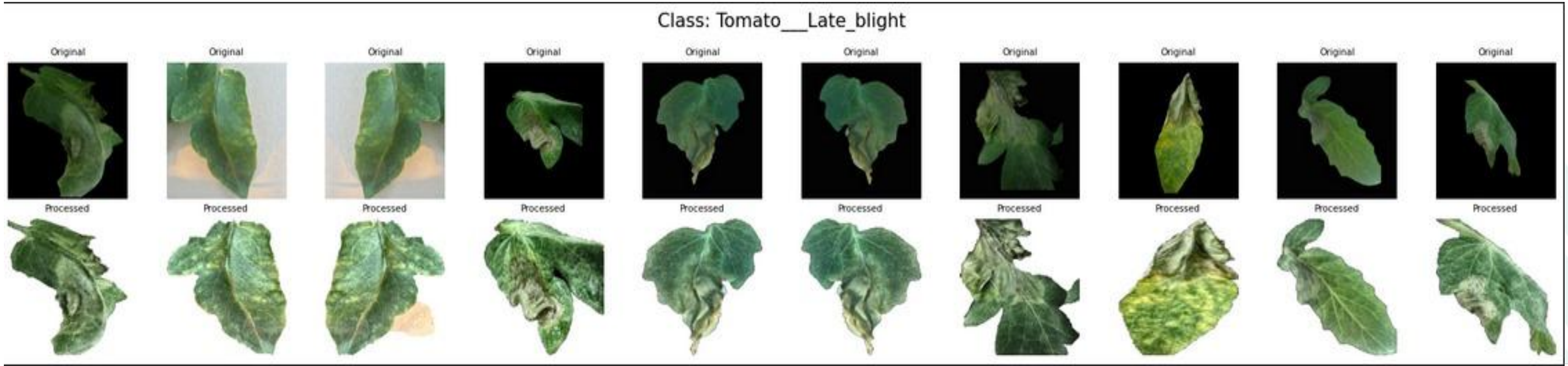
**Kshitij:** NLP task, DataLoader, Mobile web

**Himanshu :** Web Application, Mobile App, Model Training

**Tarandeep:** Preprocessing, Model Robustness, Web App

**Roshan :**Training CNN, ResNet, SVM, Random Forest, web app

**Image for the preprocessing**



# About Recommendation Model

## LLM Configuration (Key Points)

- **Default:** Llama-2-7b-chat (Meta), 8-bit quantized, runs locally, balanced performance and detail.
- **Lightweight:** Phi-2 (Microsoft), 2.7B parameters, fast, low-resource, ideal for edge devices.
- **API:** GPT-4o-mini (OpenAI), high-quality via API, no GPU needed, internet and API key required.
- Use streamlit for the web                                        use kotlin for making mobile app