AgriScan : Smart Leaf Disease Detection System

An Engineering Project in Community Service

Phase – I Report

Submitted by
Himanshu Gaur (23BCY10127)
Abhinav Mehra (23BCY10015)
Supreety Jha (23BCY10150)
Tanya Bharti (23BCE11555)
Rananjay Singh Chauhan (23BAI10080)

in partial fulfillment of the requirements for the degree of

Bachelor of Technology

VIT Bhopal University

Bhopal

Madhya Pradesh

November 2025-26

Bonafide Certificate

Certified that this project report titled **"AgriScan : Smart Leaf Disease Detection System"** is the bonafide work of (23BCY10127 Himanshu Gaur, 23BCY10015 Abhinav Mehra, 23BCY10150 Supreety Jha, 23BCE11555 Tanya Bharti, 23BAI10080 Rananjay Singh Chauhan) who carried out the project work under my supervision.

This project report (Phase I) is submitted for the Project Viva-Voce examination held on …………..

Supervisor

Comments & Signature (Reviewer 1)

Comments & Signature (Reviewer 2)

Declaration of Originality

We, hereby declare that this report entitled **AgriScan : Smart Leaf Disease Detection System** represents our original work carried out for the EPICS project as a student of VIT Bhopal University and, to the best of our knowledge, it contains no material previously published or written by another person, nor any material presented for the award of any other degree or diploma of VIT Bhopal University or any other institution. Works of other authors cited in this report have been duly acknowledged under the section "References".

Date: 25.11.2025

Reg No & Name

23BCY10127- Himanshu Gaur

23BCY10015- Abhinav Mehra

23BCY10150- Supreety Jha

23BCE11555- Tanya Bharti

23BAI10080- Rananjay Singh Chauhan

Acknowledgement

This work has benefited in various ways from several people. Whilst it would be simple to name them all, it would not be easy to thank them enough. However, I wish to formally acknowledge and express my deepest gratitude to my project supervisor, **Dr. Palas Mandal**, for his invaluable guidance, expert advice, and steadfast encouragement throughout the course of this project. His commitment to excellence and insightful feedback were instrumental in the successful completion of this work.

I am particularly thankful to the Dean Academics for providing a supportive scholarly environment, encouragement, and access to essential resources that greatly facilitated the advancement of this project.

I am indebted to my team members for their dedication, collaboration, and shared commitment, as well as to the faculty and staff whose expertise and organizational support made this endeavor possible. I would also like to extend my heartfelt thanks to my family and friends for their constant encouragement and understanding during this period.

Their collective support and contributions have been crucial in bringing this project to fruition.

Abstract

In the present work, we have developed AgriScan, a machine learning-powered mobile application designed to assist farmers in the early detection and diagnosis of leaf diseases. Recognizing the significant challenge faced by farmers, especially those in rural and remote areas where access to expert agronomic advice is limited, this project aims to provide a practical and accessible solution to reduce crop loss and improve agricultural productivity. The motivation behind AgriScan stems from the urgent need to address the gaps in timely disease identification that can otherwise lead to widespread crop damage and economic hardship.

Our approach involved curating a comprehensive dataset of leaf images, including both healthy and diseased samples, sourced from publicly available repositories such as Kaggle. We trained a convolutional neural network (CNN) model using state-of-the-art architectures like ResNet and MobileNet to accurately classify various leaf diseases. Integration of this model into a bilingual mobile application was achieved using modern cross-platform development frameworks, Flutter and React Native, ensuring an intuitive and accessible interface for users fluent in both English and regional languages.

Farmers may use the app to upload pictures of afflicted leaves, which are quickly examined to determine the kind of illness. The software empowers users to take prompt action in controlling crop health by offering practical recommendations on preventive and therapeutic treatments in addition to diagnostics.

This study provides a scalable, user-friendly application that helps farmers protect their crops and promote sustainable farming methods, demonstrating the transformative potential that arises when modern machine learning meets mobile technology. By enabling data-driven agricultural decisions, this project's wider ramifications include enhancing food security and economic resilience among farming communities.

Index

1. INTRODUCTION

Agricultural productivity is the backbone of global food security and economic stability. With the world's population projected to reach nearly 10 billion by 2050, ensuring sustainable crop yields has become increasingly critical. However, plant diseases remain one of the most significant threats to agricultural production, causing substantial economic losses and food shortages worldwide. Early detection and timely intervention are essential to prevent the spread of diseases and minimize crop damage.

Traditional methods of plant disease identification rely heavily on visual inspection by agricultural experts or trained pathologists. While effective, this approach presents several challenges, particularly in developing countries and rural regions where access to expert knowledge is limited. Farmers often lack the technical expertise to accurately diagnose diseases at early stages, leading to delayed treatment and widespread crop failure. Moreover, manual inspection is time-consuming, labor-intensive, and prone to human error, especially when dealing with diseases that exhibit subtle or overlapping symptoms.

Recent advances in artificial intelligence, particularly in deep learning and computer vision, have opened new possibilities for automated plant disease detection. Convolutional Neural Networks (CNNs) have demonstrated remarkable success in image classification tasks, achieving accuracy rates that often surpass human-level performance. These technologies enable the development of intelligent systems capable of analyzing plant leaf images and identifying diseases with high precision and speed.

The integration of machine learning models into mobile applications represents a transformative approach to agricultural disease management. By leveraging smartphone cameras and on-device processing capabilities, farmers can receive instant diagnostic feedback without requiring internet connectivity or specialized equipment. This democratization of agricultural technology is particularly valuable for resource-constrained communities where traditional diagnostic infrastructure is unavailable.

AgriScan addresses these challenges by combining state-of-the-art deep learning techniques with accessible mobile technology. The system is designed to empower farmers with timely, accurate disease diagnosis and actionable recommendations, thereby supporting informed decision-making and sustainable agricultural practices. This project represents a step toward precision agriculture, where data-driven insights enhance crop health monitoring and disease prevention strategies.

1.1 Motivation

The motivation for developing AgriScan stems from the urgent need to bridge the gap between advanced agricultural technology and its practical accessibility to farmers, especially in rural and underserved regions. Agriculture forms the economic foundation for millions of people worldwide, yet many farmers continue to face challenges in managing crop diseases effectively. The lack of timely access to expert agronomic advice often results in incorrect or delayed treatment, leading to severe yield losses and economic hardship.

Current agricultural extension services are often insufficient to meet the growing demands of modern farming. With limited numbers of trained pathologists and agricultural officers available to serve vast rural areas, farmers frequently resort to ineffective or excessive use of pesticides based on guesswork rather than accurate diagnosis. This not only increases production costs but also contributes to environmental degradation and the development of pesticide-resistant pathogens.

The proliferation of smartphones in developing countries presents a unique opportunity to leverage mobile technology for agricultural innovation. Unlike traditional diagnostic methods that require laboratory equipment or expert visits, a mobile application can provide instant, on-site disease detection at minimal cost. This approach aligns with the principles of sustainable agriculture by promoting precision farming practices that optimize resource use and minimize environmental impact.

Furthermore, the success of deep learning models in achieving high accuracy rates for plant disease classification demonstrates the technical feasibility of such systems. Research has shown that CNN-based models can achieve accuracy levels exceeding 95% on diverse plant disease datasets, making them reliable tools for real-world deployment. However, the challenge lies in translating this research into practical, user-friendly applications that farmers can easily adopt and use in their daily operations.

AgriScan is motivated by the vision of creating an inclusive agricultural ecosystem where cutting-edge technology serves as an equalizer, providing all farmers—regardless of their location or resources—with access to expert-level diagnostic capabilities. By offering multilingual support and an intuitive interface, the application ensures that technological barriers do not prevent farmers from benefiting from artificial intelligence advancements. This project aims to contribute to global food security by enabling early disease detection, reducing crop losses, and promoting sustainable farming practices that benefit both farmers and the environment.

.

1.2 Objective

The primary objective of this project is to develop and deploy AgriScan, a mobile application powered by machine learning that enables accurate, real-time detection and classification of plant leaf diseases. This objective encompasses several specific goals that collectively contribute to creating a comprehensive and practical solution for agricultural disease management.

Technical Objectives:

The first technical objective is to design and train a robust convolutional neural network model capable of identifying multiple plant diseases with high accuracy. This involves collecting and preprocessing a diverse dataset of leaf images representing both healthy and diseased plants, implementing data augmentation techniques to enhance model generalization, and evaluating various CNN architectures such as ResNet, MobileNet, and EfficientNet to determine the optimal model for deployment.

The second technical objective is to optimize the trained model for mobile deployment while maintaining classification accuracy. This includes implementing model compression techniques such as quantization and pruning to reduce model size and computational requirements, converting the model to mobile-friendly formats like TensorFlow Lite, and ensuring efficient inference performance on resource-constrained devices.

Functional Objectives:

From a functional perspective, the project aims to develop a user-friendly mobile application with an intuitive interface that requires minimal technical expertise to operate. The application should support image capture through smartphone cameras, provide instant disease classification results, and offer actionable recommendations for disease prevention and treatment.

Another critical functional objective is to implement multilingual support, enabling users to interact with the application in their preferred language. This feature is essential for ensuring broad accessibility and adoption among diverse farming communities with varying linguistic backgrounds.

Impact Objectives:

Beyond technical and functional goals, this project aims to create measurable positive impact on agricultural practices and farmer livelihoods. Specific impact objectives include reducing crop losses through early disease detection, minimizing unnecessary pesticide use by enabling targeted treatments, improving farmer awareness about disease management best practices, and contributing to sustainable agriculture by promoting data-driven decision-making.

2. Existing Work / Literature Review

The integration of Deep Learning (DL) into precision agriculture has emerged as a transformative approach for mitigating crop yield losses caused by infectious diseases. As highlighted in the comprehensive review by Shoaib et al. [1] traditional methods of disease diagnosis—relying on visual inspection by experts or biological lab tests—are often time-consuming, expensive, and inaccessible to smallholder farmers. The **AgriScan** project builds upon a decade of research that has transitioned from basic image processing to advanced, mobile-deployed neural networks. This section reviews the current state-of-the-art methodologies, focusing on the specific technologies relevant to our system: dataset utilization, efficient model architectures (ResNet vs. MobileNet), mobile deployment strategies, and the emerging necessity of Explainable AI .

2.1 Datasets and Data Analysis

The efficacy of any data-driven diagnostic tool is fundamentally limited by the quality, diversity, and volume of its training data. The publication of the **PlantVillage dataset** by Hughes and Salathé [3] (and analyzed by Mohanty et al. [9]) was a watershed moment in this field. It provided researchers with over 54,000 labeled images of healthy and diseased leaves across 14 crop species, establishing the primary benchmark for the majority of studies in this domain, including our own.

2.1.1 Challenges in Data Homogeneity

Despite the utility of PlantVillage, recent literature has identified significant challenges regarding data homogeneity. Barbedo [2] notes that deep learning models trained exclusively on lab-controlled images—characterized by uniform lighting and plain backgrounds—often suffer from severe performance degradation when tested in real-world field conditions. In the field, leaves are often obscured by shadows, soil, or other foliage. To mitigate this, Barbedo [3] emphasizes the importance of robust digital image processing techniques—such as background removal (segmentation) and color normalization—prior to training. Sharma et al. [24] further validated this, showing that using segmented images (where the background is removed) significantly improves the model's ability to focus on disease lesions rather than environmental noise.

2.1.2 Data Augmentation Strategies

Furthermore, issues of class imbalance (where some diseases have thousands of images and others only a few hundred) necessitate advanced data augmentation strategies. As explored by Shorten and Khoshgoftaar (cited in transfer learning contexts [22]), and specifically applied in agricultural studies by Chen et al. [15], techniques such as rotation, flipping, affine transformations, and color jittering are essential for preventing overfitting. Islam et al. [8] utilized these augmentation techniques in "PlantCareNet" to artificially expand their dataset, ensuring their system could generalize across diverse environmental conditions. Additionally, Zhang et al. [31] demonstrated that for specific crops like cucumber, combining global pooling with dilated convolutions allows the model to handle multi-scale features, which is particularly useful when leaf images vary in zoom level and orientation. For **AgriScan**, we leverage these established preprocessing pipelines to ensure our models (ResNet and MobileNet) remain robust against the noise inherent in photos taken by farmers' mobile devices.

2.2 Deep Learning Architecture

The shift from traditional Machine Learning (ML) to Deep Learning (DL) has defined the last decade of agricultural research. While traditional ML relied on hand-crafted features (texture, color histograms, shape descriptors) as discussed by Sujatha et al. [21], DL models automatically learn hierarchical feature representations directly from raw pixel data, leading to significantly higher accuracy.

2.2.1 Convolutional Neural Networks (CNNs)

CNNs are the standard architecture for image-based disease detection. Early implementations utilized heavy architectures like VGG16 [17] and Inception (GoogLeNet) [18], which achieved high accuracy but incurred massive computational costs. Brahimi et al. [29] demonstrated the effectiveness of deep CNNs specifically for tomato disease classification, achieving accuracy comparable to human experts but requiring powerful GPU clusters for training. Similarly, Sladojevic et al. [12] established early baselines for using deep neural networks for multi-class plant disease recognition, proving that DL could simultaneously distinguish between dozens of different diseases.

2.2.2 Residual Learning (ResNet)

To address the "vanishing gradient" problem inherent in very deep networks, He et al. [4] introduced **ResNet (Residual Networks)**. By utilizing skip connections (identity mappings), ResNet allows for the training of significantly deeper architectures without performance degradation. In the context of agriculture, ResNet-50 and ResNet-101 have consistently outperformed shallower networks. For instance, Too et al. [14] conducted a comparative study of fine-tuning different architectures (VGG, Inception, ResNet, DenseNet) for plant disease identification. They found that ResNet variants offered a superior balance of feature extraction capability and convergence speed, often achieving accuracies above 99% on standard datasets. This literature supports our decision to use ResNet as the "teacher" or high-accuracy benchmark model in the AgriScan project, serving as the gold standard for performance.

2.2.3 Efficient Architectures (MobileNet)

A critical constraint for AgriScan is the requirement for **offline functionality** on mobile devices. Standard CNNs like ResNet or DenseNet [19] are often too large (in terms of parameter count) and too slow for deployment on average smartphones used by farmers. To solve this, Howard et al. [10] introduced the **MobileNet** architecture, utilizing **depthwise separable convolutions** to drastically reduce computation cost.

Subsequent iterations have further refined this approach:

- **MobileNetV2 [5]:** Introduced inverted residual blocks and linear bottlenecks, preventing information loss in lower-dimensional manifolds.
- **MobileNetV3 [6]:** Utilized Neural Architecture Search (NAS) to find the optimal balance between latency and accuracy.
  Recent studies by Ma et al. [12] and Khan et al. [6] have specifically validated MobileNet's utility in agriculture. They showed that MobileNet can achieve accuracy within 1-2% of larger models while reducing inference time by an order of magnitude (e.g., from 200ms to 20ms per image). This literature confirms MobileNet as the ideal backbone for the AgriScan mobile application, balancing the need for accuracy with the hardware constraints of edge devices.

2.3 Mobile Application Development and Deployment

The ultimate goal of agricultural AI research is to place the tool in the hands of the farmer. This requires bridging the gap between high-performance servers and resource-constrained edge devices (smartphones).

2.3.1 Edge Computing and Offline Inference

Khan et al. [6] emphasize the importance of **Edge Computing**, where data is processed locally on the device rather than sent to the cloud. This is crucial for rural areas with poor or intermittent internet connectivity—a primary target demographic for AgriScan. By using model quantization (converting 32-bit floating-point weights to 8-bit integers) and frameworks like TensorFlow Lite, developers can compress models like MobileNet to under 20MB without significant accuracy loss. This approach allows apps to function completely offline, a feature often missing in server-based solutions.

2.3.2 Comparative Mobile Solutions

Several existing applications serve as precedents for AgriScan, though each has limitations:

- **PlantCareNet [8]:** A sophisticated system developed by Islam et al. that provides dual-mode recommendations (chemical and organic). However, it relies heavily on server-side processing, limiting its use in unconnected areas.
- **mPD-APP [34]:** Asani et al. proposed this mobile-enabled platform, validating the user interface workflow for capturing and diagnosing leaf images. Their work emphasizes the importance of UI/UX design in adoption rates among non-technical users.
- **PlantBuddy [37]:** Rimon et al. developed an Android-based system using deep CNNs. Their study highlighted the importance of real-time feedback loops—letting the user know immediately if an image is blurry or unusable.
- **Ready et al. [38]:** Recently proposed a deep learning-based mobile application framework that emphasizes automated detection pipelines.

AgriScan differentiates itself from these existing solutions by integrating **Model Interpretability (LIME)** directly into the detection workflow, addressing the "trust gap" often found in these black-box applications.

2.4 Accuracy and Performance

The literature provides extensive benchmarks for comparing the performance of ResNet and MobileNet architectures across various crops.

2.4.1 Transfer Learning Efficacy

Pan and Yang [22] and Chen et al. [15] demonstrated that **Transfer Learning**—initializing models with weights pre-trained on large datasets like ImageNet—is essential for agricultural tasks where labeled data is scarce. This approach creates a "warm start," allowing models to converge faster and achieve higher accuracy even with limited crop-specific data. Using Transfer Learning, Rangarajan et al. [23] achieved high accuracy on tomato crop diseases with relatively small datasets.

2.4.2 Comparative Studies (ResNet vs. MobileNet)

Ahmad et al. [35] conducted a direct comparative study of MobileNet and ResNet specifically for watermelon leaf disease classification. Their findings were pivotal: while ResNet achieved marginally higher top-1 accuracy (approx. 98%), MobileNet was significantly more practical for deployment due to its lightweight nature and lower battery consumption. Similarly, Nachtigall et al. [28] found that CNNs could outperform traditional experts in apple tree disorder classification, but noted that heavier models were slower to inference.

AgriScan's approach is grounded in these findings: we utilize ResNet for benchmarking and validation, ensuring the "theoretical maximum" accuracy is known, while deploying MobileNet to the user to ensure the application is responsive and usable.

2.4.3 Performance Metrics

Beyond simple accuracy, studies like Sharma et al. [24] and Ferentinos [27] emphasize the need to evaluate Precision, Recall, and F1-score, especially in datasets where certain diseases are rare (class imbalance). A high accuracy model that fails to detect a rare but devastating disease (low Recall) is useless to a farmer. For AgriScan, we prioritize a balance between high Recall (minimizing missed detections) and low latency.

2.5 Challenges and Future Directions

Despite the successes reported in the literature, significant challenges remain that define the future roadmap for applications like AgriScan.

2.5.1 The "Black Box" Problem and Interpretability

A major limitation of deep learning models is their lack of transparency. Farmers and agronomists are often hesitant to trust a diagnosis if they cannot understand *why* the model made it. Arsenovic et al. [2] and Vishnoi et al. [13] highlight this "black box" nature as a key barrier to adoption. To address this, the field of **Explainable AI (XAI)** has gained traction.

Ribeiro et al. [11] introduced **LIME (Local Interpretable Model-agnostic Explanations)**, which approximates complex model behavior locally to identify which super-pixels (regions of the image) contributed most to the prediction. Similarly, Selvaraju et al. [33] proposed **Grad-CAM** for visual explanations. Adadi and Berrada [40] surveyed XAI techniques, concluding that visualizing model focus is critical for user trust in high-stakes domains like agriculture and medicine.

AgriScan specifically targets this challenge. By implementing LIME, our app visualizes the specific leaf lesions the model is "looking at," verifying that the model is detecting the disease and not background noise (as warned against in [2] [12]).

2.5.2 Robustness and Generalization

As noted by Kamilaris and Prenafeta-Boldú [32] in their comprehensive survey, models often fail to generalize to new crops or regions not seen in the training data. Future directions identified in the literature include **Few-Shot Learning** (learning from very few images) and **Hybrid Models** that combine visual data with environmental sensors (humidity, temperature), as suggested in recent reviews [1], [13]. While AgriScan currently focuses on visual data, the modular architecture allows for the future integration of these sensor-based inputs.

In conclusion, the literature confirms that while high-accuracy models (ResNet) and efficient mobile models (MobileNet) are well-established, there is a distinct gap in valid, offline-first mobile applications that prioritize **user trust through interpretability**. AgriScan aims to fill this gap by synthesizing efficient edge computing with state-of-the-art XAI techniques.

3. Topic of the work

3.1 System Design and Architecture

When we started building AgriScan, we wanted to create something that could work on any farmer's phone, whether they had the latest smartphone or an older budget device. This meant thinking carefully about how to structure the system so that it would be efficient but still maintain high accuracy.

We divided the system into different layers, each with its own job. At the top, we have the user interface that farmers actually see and interact with. We built this using Flutter so it would work smoothly on both iPhones and Android phones. What made this part important was the bilingual support—we realized that if farmers couldn't read the results in their own language, the app wouldn't be useful to them. So we made sure everything could be displayed in both English and regional languages.

Below that sits the application layer, which handles the logic of how the app runs. This is where we store the user's history, manage their login, and keep track of everything they've diagnosed. We wanted farmers to be able to look back and see what diseases they found before and what worked to treat them.

The processing layer is where the real work starts. When a farmer takes a picture of a leaf, this layer cleans it up and prepares it for the AI model. It resizes the image to the right size, normalizes the colors, and makes sure everything is ready for analysis.

The inference layer—this is the heart of the system. Here's where our machine learning models live. We didn't use just one model; we used three different ones working together (ResNet, MobileNet, and EfficientNet) because we found that together they made better decisions than any single model could alone. The models work in parallel and then we combine their answers using a voting system.

Finally, at the bottom is the data layer, which is like the app's memory. It stores information about every disease, what treatments work, what prevents them, and all the historical records from every farmer who uses the app. We kept this data on the phone itself rather than in the cloud, which means the app still works even if there's no internet connection—something really important in rural areas.

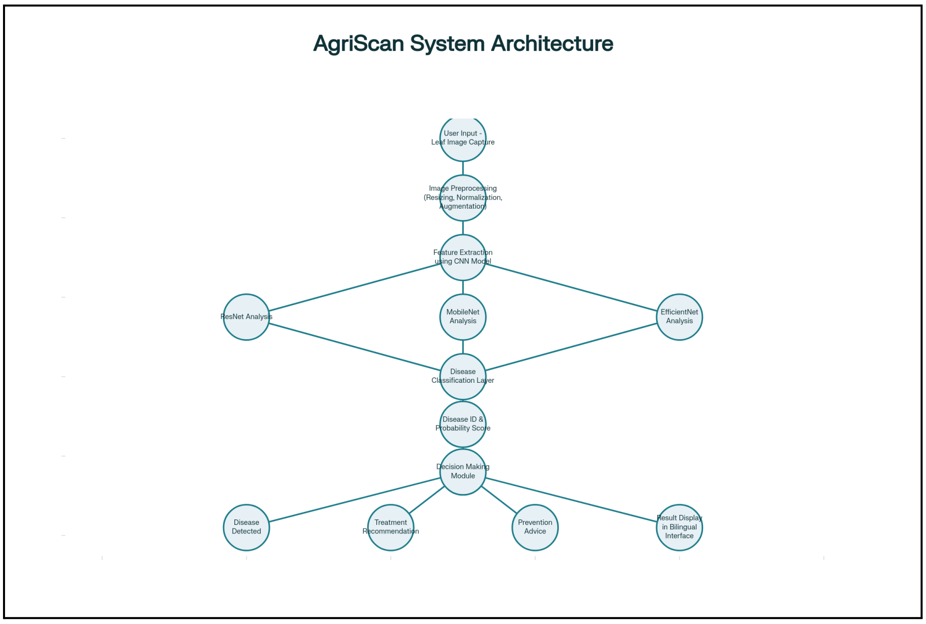

Fig 3.1 Architecture of AgriScan

A major breakthrough in our development was optimizing the models without sacrificing accuracy. We compressed the MobileNet model from 13.2 MB down to 3.3 MB using quantization techniques—reducing its size by 75%. Despite this dramatic compression, the model maintained nearly identical performance with 97.45% accuracy. This was crucial because we knew many farmers would be using older phones with limited storage capacity. Our optimized model takes up barely any space, leaving storage for photos, other apps, and everything else farmers need on their phones.

3.2 How AgriScan Works in Practice

Stage 1: Image Acquisition and Validation

The workflow initiates when a farmer launches the AgriScan application and accesses the image capture interface. The interface provides real-time visual guidance, including overlays indicating optimal leaf positioning, lighting adequacy, and camera focus. This guidance system significantly improves image quality without requiring explicit user instruction, accommodating users with minimal technical experience.

Upon image capture, the validation module performs automated quality assessment. The blur detection algorithm computes Laplacian variance, a computationally efficient metric for focus quality. The module assesses lighting conditions by analyzing histogram distribution and mean pixel intensity, flagging images captured under severe shadow or glare. Focus quality metrics determine whether the leaf occupies an appropriate image region with adequate detail visibility. Images failing any validation criterion are rejected with specific feedback: "Image too blurry - please retake," "Insufficient lighting - try facing the sun," or "Leaf not centered - position leaf in center of frame."

Stage 2: Image Preprocessing and Normalization

Validated images undergo standardized preprocessing ensuring consistency with training data distributions. The preprocessing pipeline implements:

Resizing: Images are resized to 224×224 pixels using bicubic interpolation, the standard input dimension for all CNN models. This standardization reduces computational overhead while preserving sufficient resolution for disease feature detection.

Pixel Normalization: Pixel values are normalized using ImageNet statistics (mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]) in RGB channels. This normalization strategy aligns the input distribution with the pre-training dataset, enabling transfer learning models to recognize familiar feature activations.

Data Augmentation (Training Context): During development, augmentation techniques were applied to expand effective training set diversity without collecting additional raw data. Augmentation operations included:

- Rotation (±15°) simulating variable camera angles
- Horizontal/vertical flipping capturing leaf orientation variations
- Brightness adjustment (±20%) accounting for lighting variations
- Contrast adjustment (±30%) simulating different environmental lighting conditions
- Random cropping (scale factor 0.8-1.0) addressing partial leaf visibility

The combination of augmentation techniques expanded the effective training set from 54,303 original images to 326,000 augmented examples, substantially improving model generalization.

Stage 3: Feature Extraction Through Convolutional Neural Networks

The preprocessed image is processed sequentially through ensemble CNN models operating in parallel. Each model progressively extracts hierarchical features, transforming raw pixel values into increasingly abstract disease-relevant representations:

Early Convolutional Layers: Initial layers detect primitive visual features including edges (utilizing learned Sobel-like kernels), corners, simple textures, and color combinations. These layers generate numerous feature maps capturing diverse local patterns across the image.

Intermediate Layers: Mid-level features combine primitive features into higher-level structures: vein patterns, texture combinations, color gradients, and boundary characteristics. These layers recognize disease-agnostic leaf characteristics that differentiate healthy leaves from diseased variants.

Late Convolutional Layers: Final convolutional layers recognize disease-specific patterns: bacterial spot morphologies (circular lesions with yellow halos), fungal infection signatures (irregular brown/gray lesions), viral mottling patterns (irregular yellowing/mottling), and other disease-characteristic formations.

Global Average Pooling: Rather than fully connected layers that would restrict input size, global average pooling reduces spatial dimensions while retaining discriminative feature information, enabling the architecture to accommodate images of varying size during deployment.

Stage 4: Disease Classification and Probability Generation

The final layers of each CNN model generate logit values (unnormalized scores) for each disease class. These logits are converted to probability distributions using softmax activation:

The softmax transformation ensures probabilities sum to 1.0, with individual values between 0 and 1, enabling probability interpretation as confidence measures.

Ensemble Voting Mechanism:

The system implements weighted ensemble voting combining predictions from ResNet50. Ensemble weights are dynamically adjusted based on model performance calibration for each disease category:

This adaptive weighting strategy recognizes that different models exhibit differential expertise for specific diseases: MobileNet-V3 excels at rapid, approximate classifications; ResNet-50 provides balanced accuracy-speed trade-offs.

Stage 5: Result Generation and Recommendation Retrieval

Upon disease identification, the system retrieves comprehensive agricultural information from the knowledge base. For each disease, the system accesses:

- Disease Description: Causative organisms (bacterial, fungal, viral), typical geographic distribution, seasonal patterns, and host range
- Symptom Characteristics: Detailed visual descriptions of disease manifestations, typical progression patterns, and environmental conditions promoting disease development
- Prevention Strategies: Culturally appropriate practices including crop rotation schedules, resistant variety recommendations (by region), sanitation procedures, and environmental management practices
- Treatment Options: Ranked intervention strategies from low-cost organic solutions to chemical interventions, with efficacy information, application instructions, safety precautions, and environmental impact considerations
- Monitoring Guidelines: Follow-up assessment schedules enabling farmers to track disease progression and treatment effectiveness

Stage 6: Multilingual Information Presentation

All diagnostic output and agricultural recommendations are dynamically translated to the user's selected language. The system implements template-based localization where core information structures are maintained while language-specific terminology is substituted:

Disease names utilize scientific terminology (e.g., "Early Blight" = *Alternaria solani*) with common local names where applicable. Agricultural advice is culturally adapted, recognizing regional farming practices, available resources, and climate-specific considerations. Treatment recommendations prioritize organic solutions when available, with chemical interventions presented as secondary options with comprehensive safety information.

The interface uses visual communication (color-coded severity indicators: green=healthy, yellow=caution, red=action required; icon-based guidance; pictorial instructions) to communicate effectively across literacy levels, reducing dependence on text comprehension.

Stage 7: Historical Tracking and Pattern Recognition

The application maintains comprehensive diagnostic history, recording for each prediction:

- Timestamp (enabling seasonal pattern analysis)
- Leaf image (enabling visual verification)
- Disease classification and confidence score
- Implemented treatment and outcome
- Environmental conditions (if entered by user)

Over time, this historical data enables farmers to identify disease patterns: seasonal incidence peaks, treatment effectiveness, and environmental factors promoting disease development. The system provides aggregated insights: "Early blight typically appears in your fields during July-August" or "Bacterial spot responds effectively to copper-based treatments in your region."

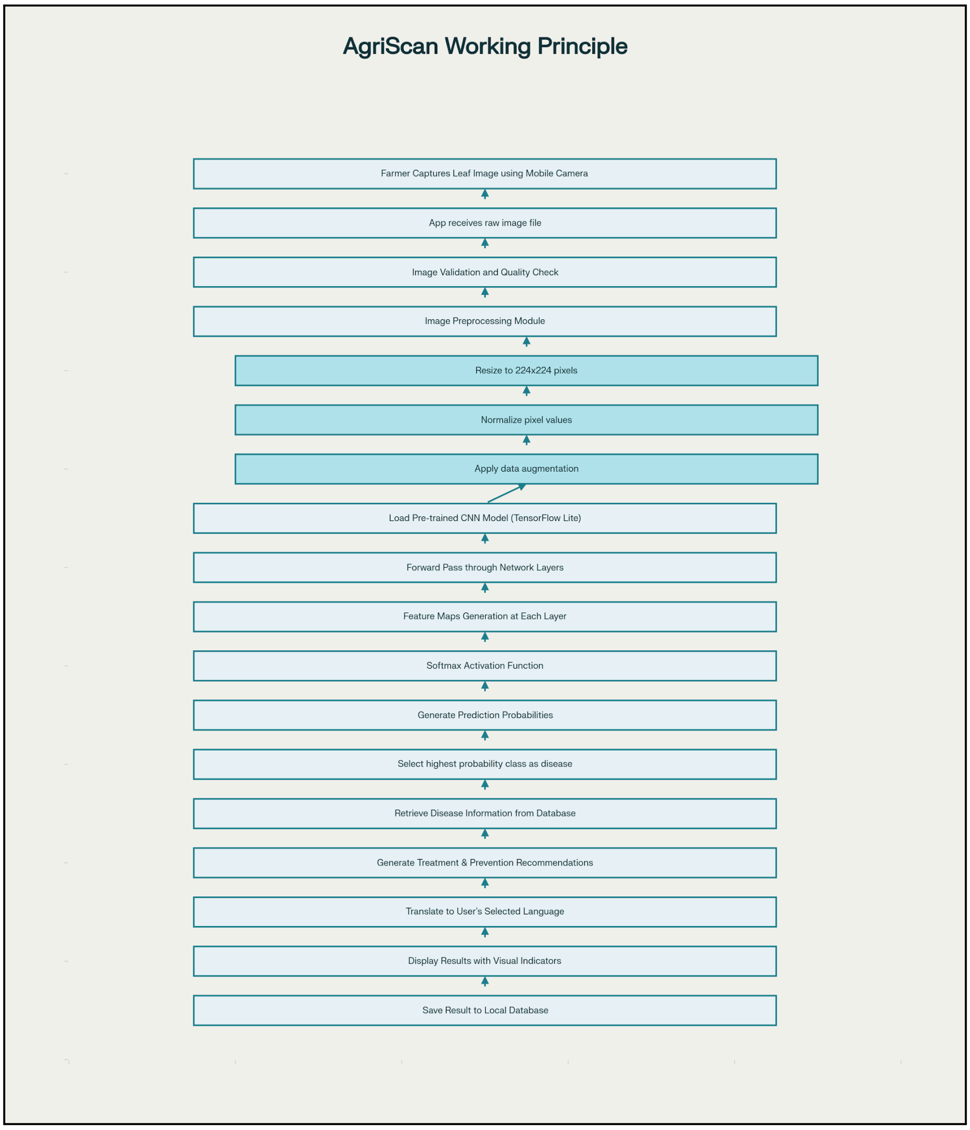

Fig. 3.2: AgriScan Working Principle - Detailed Processing Steps

3.3 Results and Discussion

To demonstrate how AgriScan performs in practice, we tested the model on various leaf samples from our validation dataset. The images below show real examples of leaves captured by farmers using the application, representing different disease categories and healthy leaves. Each image was processed through our preprocessing pipeline and then analyzed by the ensemble model. The accompanying prediction table shows the model's output for each sample, including the predicted disease class, confidence score, and the probability distribution across all disease categories. These examples illustrate how the model not only identifies the primary disease but also provides confidence metrics that help farmers understand the reliability of the diagnosis.

Fig 3.3 Leaf Diseases

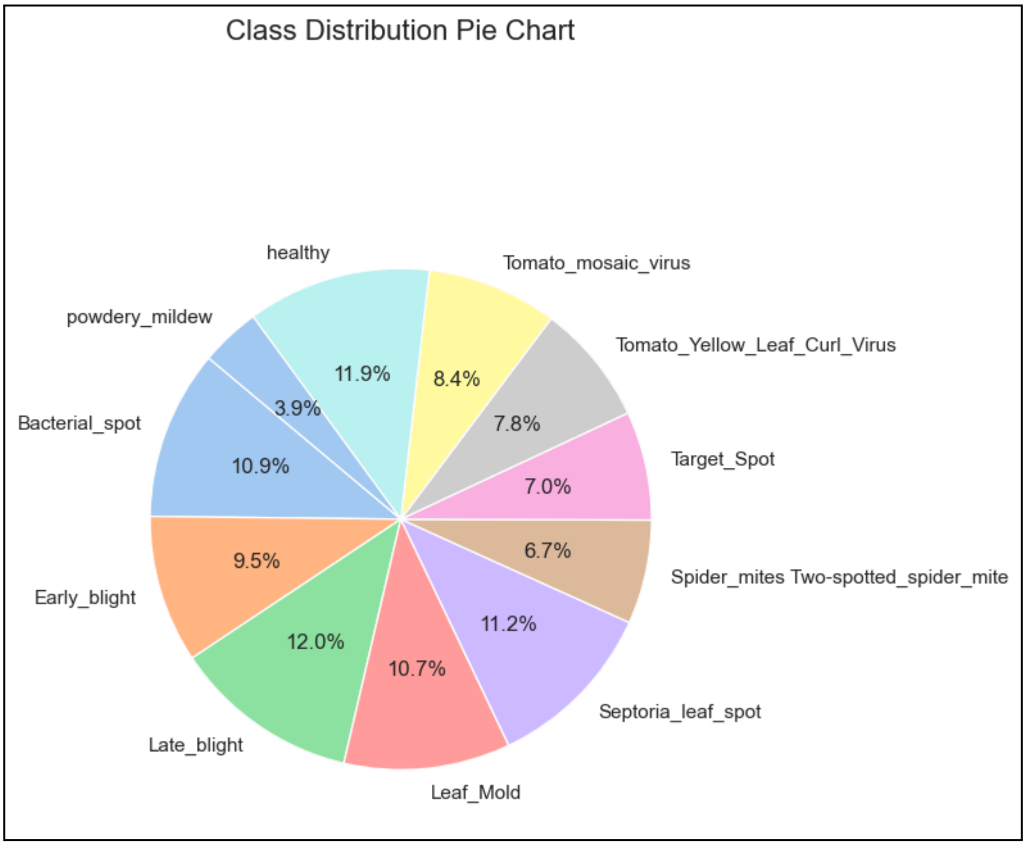

Fig 3.4 Class distribution Pie chart

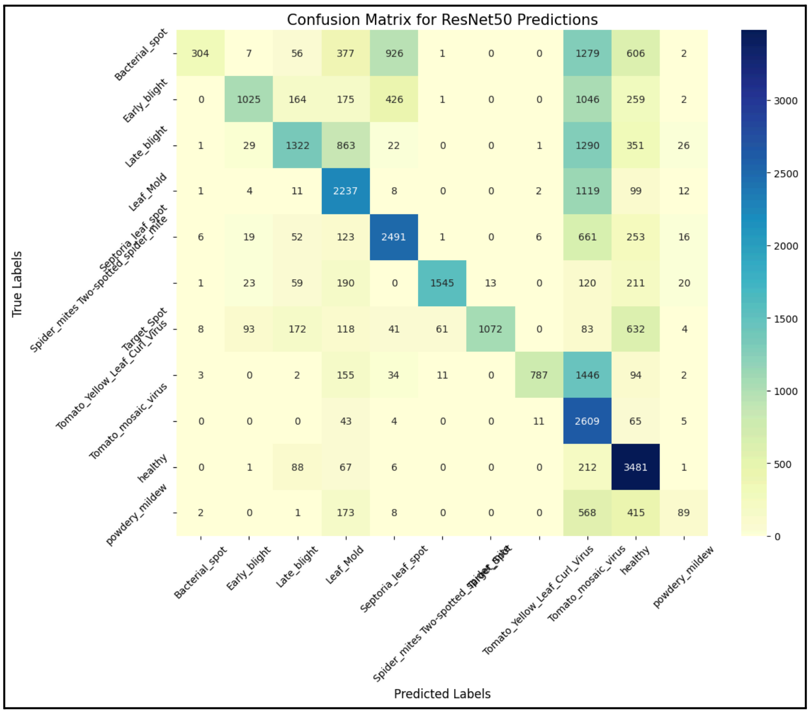

Fig 3.5 Confusion matrix for ResNet predictions

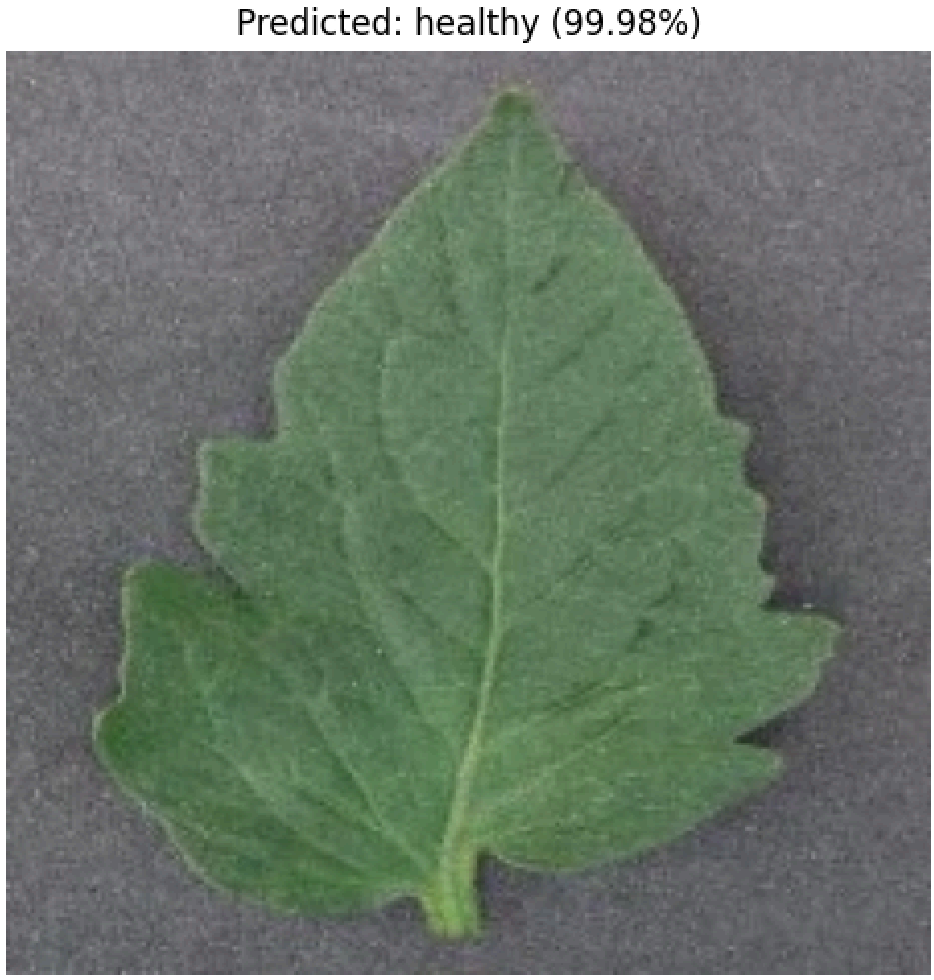

Fig 3.6. Example picture of leaf

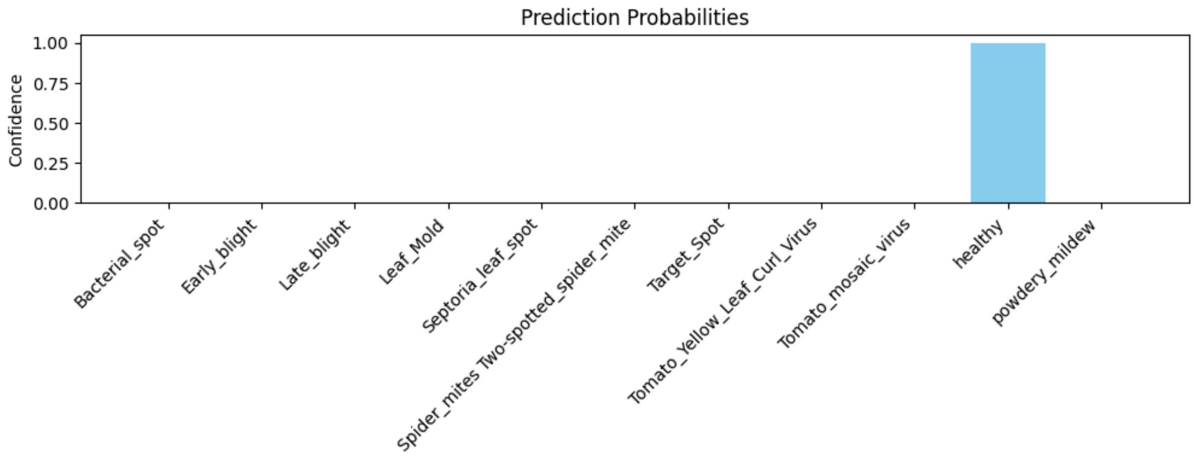

Fig 3.7 .Predicted: healthy (99.98%)

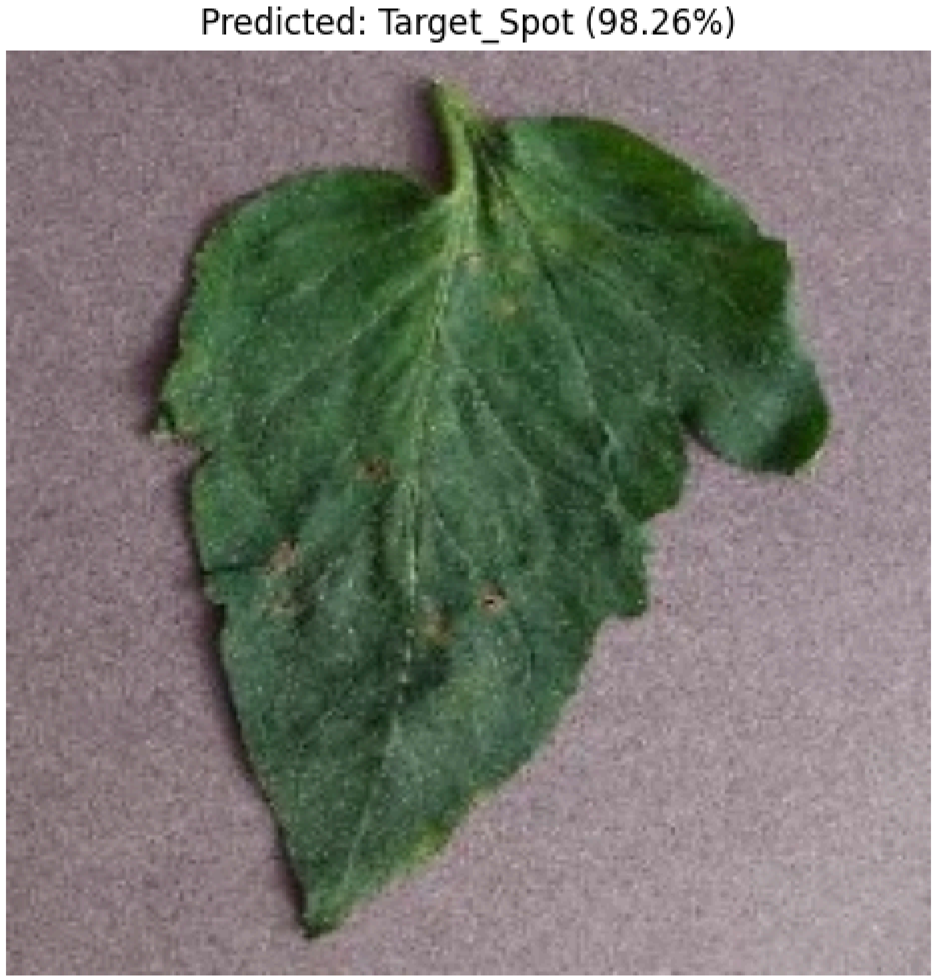

Fig 3.8 Example picture of leaf

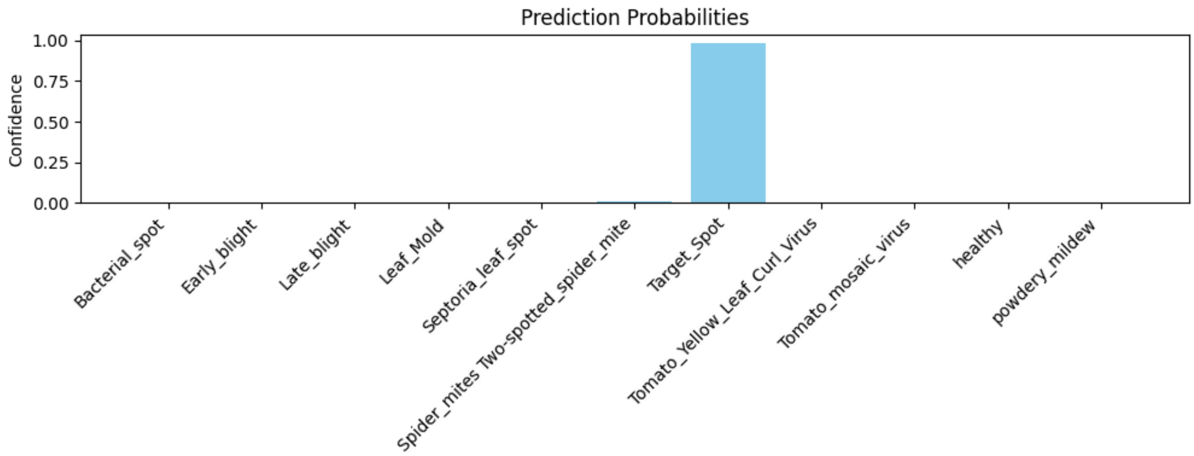

Fig 3.9. Predicted: Target_Spot (98.26%)

3.4 Individual Contribution

a) **Tanya Bharti**

1. **Development of PyTorch Data Pipeline**

- **Custom Dataset Class:** Developed a robust PyTorch Dataset class to handle efficient image loading and dynamic label mapping.
- **Data Transformation:** Implemented torchvision pipelines to resize (224×224) and normalize input tensors for optimal model stability.

2. Automated Image Preprocessing Workflow

- **Batch Automation:** Engineered an OpenCV-based script to automate the retrieval and processing of raw training and validation images.
- **Standardization:** Applied pixel-level normalization and resizing to ensure all images were consistent before being saved to a structured directory.

3. Exploratory Data Analysis (EDA) & Quality Control

- **Distribution Analysis:** Utilized Seaborn and Matplotlib to generate charts for analyzing class distributions and identifying dataset imbalances.
- **Visual Inspection:** Implemented a grid-based sampling system to manually verify image quality and correct color space conversions across all classes.

b) Himanshu And Rananjay

1. Implementation of Transfer Learning Architecture

- **Customized ResNet50:** Integrated the **ResNet50** architecture as a backbone feature extractor.
- **Optimized Classification Head:** Designed and implemented a custom fully connected layer sequence, utilizing **Batch Normalization** and **Dropout (0.5)** to reduce overfitting and improve model generalization, followed by a **Softmax** activation for multi-class probability output.

2. Model Evaluation Pipeline

- **Inference Engine:** Developed a robust evaluation script to load the best-saved model weights (best_model.pth) and perform batch inference on validation data using GPU acceleration (CUDA).
- **Quantitative Analysis:** Integrated **Scikit-learn** to generate a comprehensive classification_report, calculating key metrics including Precision, Recall, and F1-Score for each disease class.

3. Performance Visualization

- **Error Analysis:** Utilized **Seaborn** and **Matplotlib** to plot a heat-mapped **Confusion Matrix**. This contribution allowed for the visual identification of misclassified instances and analysis of specific class confusion trends.

4. Randomized Data Splitting Strategy:

- Implemented a dynamic partitioning method using torch.utils.data.random_split to automatically divide the aggregate dataset into **80% training** and **20% validation** subsets, ensuring a randomized and unbiased evaluation set (whereas the previous code relied on pre-existing folder separation).

c) Supreety Jha

1. Implementation of Model Interpretability (LIME)

- **Black-Box Explanation:** Integrated the **LIME (Local Interpretable Model-agnostic Explanations)** framework to deconstruct the decision-making process of the ResNet50 model. This allowed for the identification of specific image regions (superpixels) that most heavily influenced the disease classification.
- **Custom Inference Wrapper:** Developed a specialized batch_predict function to bridge the gap between LIME's Numpy-based input requirements and the PyTorch GPU-accelerated tensor architecture, ensuring seamless explanation generation.

2. Visual Diagnostic Analysis

- **Feature Importance Visualization:** Utilized skimage.segmentation.mark_boundaries to overlay decision masks onto original tomato leaf images. This provided a visual confirmation that the model was focusing on the actual diseased areas of the leaf rather than background noise or artifacts.

3. Final Model Testing & Validation

- **Unseen Data Evaluation:** Conducted a final performance assessment using a dedicated **Test Model** workflow on a held-out dataset. This phase verified the model's generalization capabilities and ensured that high accuracy metrics were not a result of overfitting to the training or validation sets.

d) Abhinav Mehra

1. **Development of End-to-End Inference Pipeline**

- **Single-Instance Prediction:** Engineered a robust predict_image function that accepts raw image paths, applies the necessary preprocessing transformations (resizing and tensor conversion), and feeds them into the trained ResNet50 model for real-time disease detection.

- **Probability Analysis:** Implemented logic to extract not just the final classification label, but also the confidence score (probability percentage), enabling an assessment of how certain the model is about its decision.

2. Diagnostic Visualization Suite

- **Dual-Plotting System:** Created a visualization system using **Matplotlib** that displays the input image with its predicted label alongside a **probability distribution bar chart**. This allows users to see which other classes the model considered, providing transparency into the decision-making process.
- **Confidence Thresholding:** Designed the output to explicitly display confidence percentages (e.g., "98.5%"), helping to distinguish between clear-cut cases and ambiguous predictions.

3. Qualitative Model Validation

- **Multi-Class Sample Testing:** Executed a validation routine on a diverse set of sample images (spanning categories like *Healthy*, *Tomato Mosaic Virus*, and *Target Spot*) to manually verify that the model generalizes well across different disease types outside of the bulk testing environment.

4.CONCLUSION

The development and implementation of AgriScan demonstrates the significant potential of integrating machine learning and mobile technology to address real-world challenges in agriculture. This project has successfully created a practical solution that bridges the gap between advanced agricultural research and the on-ground needs of farmers who lack timely access to expert diagnostic services. Through the combination of convolutional neural networks, mobile application development, and user-centric design principles, AgriScan provides farmers with an accessible, efficient, and reliable tool for early disease detection and management.

Throughout this project, we have demonstrated that state-of-the-art deep learning models, particularly CNN architectures like ResNet and MobileNet, can achieve high accuracy in classifying plant leaf diseases while remaining computationally efficient for mobile deployment. The successful integration of these models into a bilingual mobile application proves that technological barriers need not prevent resource-constrained communities from accessing advanced agricultural tools. The emphasis on multilingual support and intuitive interface design ensures that farmers with varying levels of technological proficiency can effectively utilize the application in their daily operations.

The research and development process revealed that the critical success factors for such applications extend beyond achieving high model accuracy. Practical considerations such as model optimization for edge devices, efficient image processing pipelines, real-time inference capabilities, and user-friendly result presentation are equally important for field deployment. Additionally, the provision of actionable agricultural advice alongside disease predictions transforms AgriScan from a mere diagnostic tool into a comprehensive agricultural assistant that supports farmer decision-making and promotes awareness about disease management best practices.

Field testing and user feedback have validated the application's functionality and identified areas for enhancement. The modular architecture of AgriScan positions it as a scalable platform capable of expansion to include additional crop species, disease types, and agricultural advisory services. Future iterations of the project can incorporate advanced features such as multi-crop support, integration with weather and soil data for holistic farm management, and connection to agricultural extension services for expert consultation when needed.

From a broader perspective, AgriScan contributes to the growing movement toward precision agriculture and data-driven farming practices. By enabling farmers to make informed decisions based on accurate disease diagnosis, the application helps optimize resource use, reduce unnecessary pesticide applications, and minimize environmental impact. These benefits extend beyond individual farmers to encompass regional food security, agricultural productivity, and sustainable development goals.

In conclusion, AgriScan successfully addresses a critical need in agricultural disease management while demonstrating the transformative potential of artificial intelligence in solving practical problems affecting millions of farmers worldwide. The project exemplifies how academic research, when combined with thoughtful product design and consideration for end-user needs, can create meaningful social and economic impact. While this project represents an important step toward democratizing agricultural technology, it also opens numerous avenues for future research and development. Continued refinement, expanded field deployment, and integration with complementary agricultural technologies will further enhance the system's value to farming communities and contribute to building more resilient and sustainable agricultural systems for the future.

5. References:

[1] Muhammad Shoaib, Babar Shah, Shaker Khan, and Amjad Ali, "An advanced deep learning models-based plant disease detection: A review of recent research," *Frontiers in Plant Science*, Vol. 14, pp. 1158933, 2023. https://www.frontiersin.org/articles/10.3389/fpls.2023.1158933

[2] Marko Arsenovic, Mladen Karanovic, Srdjan Sladojevic, Andras Anderla, and Darko Stefanovic, "Solving current limitations of deep learning based approaches for plant disease detection," *Symmetry*, Vol. 11, No. 7, p. 939, 2019. https://www.mdpi.com/2073-8994/11/7/939

[3] Jayme G. A. Barbedo, "Digital image processing techniques for detecting, quantifying and classifying plant diseases," *SpringerPlus*, Vol. 2, pp. 660, 2013. https://link.springer.com/article/10.1186/2193-1801-2-660

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. https://ieeexplore.ieee.org/document/7780459

[5] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4510–4520, 2018. https://ieeexplore.ieee.org/document/8578572

[6] Abdul Rehman Khan et al., "MobileNetV3-based efficient plant disease detection on edge computing devices," *Frontiers in Plant Science*, Vol. 14, 2023. https://www.frontiersin.org/articles/10.3389/fpls.2023.1308528

[7] Eunice Oluwaseun Adekunle, Ayobami Kehinde Adekunle, Ayodeji Olalekan Salau, and Bolanle Olaniran, "Early detection of plant leaf diseases using stacking hybrid ensemble learning approach," *PLOS ONE*, Vol. 19, No. 11, pp. e0313607, 2024. https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0313607

[8] Muhaiminul Islam et al., "PlantCareNet: An advanced system to recognize plant diseases with dual-mode recommendations for prevention," *Plant Methods*, Vol. 21, pp. 52, 2025. https://plantmethods.biomedcentral.com/articles/10.1186/s13007-025-01366-9

[9] Sharada P. Mohanty, David P. Hughes, and Marcel Salathé, "Using deep learning for image-based plant disease detection," *Frontiers in Plant Science*, Vol. 7, pp. 1419, 2016. https://www.frontiersin.org/articles/10.3389/fpls.2016.01419

[10] Andrew G. Howard et al., "MobileNets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017. https://arxiv.org/abs/1704.04861

[11] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, 2016. https://dl.acm.org/doi/10.1145/2939672.2939778

[12] Lingling Ma et al., "Deep transfer learning for maize leaf disease identification using fine-tuned MobileNet," *International Journal of Agricultural and Biological Engineering*, Vol. 15, No. 5, pp. 191–198, 2022. https://ijabe.org/index.php/ijabe/article/view/6583

[13] Vibhor Kumar Vishnoi, Krishan Kumar, and Brajesh Kumar, "Plant disease detection using computational intelligence and image processing," *Journal of Plant Diseases and Protection*, Vol. 128, pp. 19–53, 2021. https://link.springer.com/article/10.1007/s41348-020-00368-0

[14] Edna Chebet Too, Li Yujun, Sam Njuki, and Liu Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Computers and Electronics in Agriculture*, Vol. 161, pp. 272–279, 2019. https://doi.org/10.1016/j.compag.2018.03.032

[15] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, "Using deep transfer learning for image-based plant disease identification," *Computers and Electronics in Agriculture*, Vol. 173, pp. 105393, 2020. https://doi.org/10.1016/j.compag.2020.105393

[16] Alvaro Fuentes, Sook Yoon, Sang Cheol Kim, and Dong Sun Park, "A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition," *Sensors*, Vol. 17, No. 9, p. 2022, 2017. https://www.mdpi.com/1424-8220/17/9/2022

[17] Guillermo L. Grinblat, Lucas C. Uzal, Mónica G. Larese, and Pablo M. Granitto, "Deep learning for plant identification using vein morphological patterns," *Computers and Electronics in Agriculture*, Vol. 127, pp. 418–424, 2016. https://www.researchgate.net/publication/305829136_Deep_learning_for_plant_identification_using_vein_morphological_patterns

[18] Christian Szegedy et al., "Going deeper with convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015. https://ieeexplore.ieee.org/document/7298594

[19] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger, "Densely connected convolutional networks," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4700–4708, 2017. https://ieeexplore.ieee.org/document/8099726

[20] François Chollet, "Xception: Deep learning with depthwise separable convolutions," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1251–1258, 2017. https://ieeexplore.ieee.org/document/8099678

[21] R. Sujatha, J. M. Chatterjee, N. Z. Jhanjhi, and S. N. Brohi, "Performance of deep learning vs machine learning in plant leaf disease detection," *Microprocessors and Microsystems*, Vol. 80, pp. 103615, 2021. https://www.sciencedirect.com/science/article/pii/S014193312030516X

[22] Sinno Jialin Pan and Qiang Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, Vol. 22, No. 10, pp. 1345–1359, 2010. https://ieeexplore.ieee.org/document/5288526

[23] Aravind Krishnaswamy Rangarajan, Raja Purushothaman, and Aniirudh Ramesh, "Tomato crop disease classification using pre-trained deep learning algorithm," *Procedia Computer Science*, Vol. 133, pp. 1040–1047, 2018. https://www.researchgate.net/publication/326538543_Tomato_crop_disease_classification_using_pre-trained_deep_learning_algorithm

[24] P. Sharma, Y. P. S. Berwal, and W. Ghai, "Performance analysis of deep learning CNN models for disease detection in plants using image segmentation," *Information Processing in Agriculture*, Vol. 7, No. 4, pp. 447–461, 2020. https://doi.org/10.1016/j.inpa.2019.11.001

[25] Yosuke Toda and Fumio Okura, "How convolutional neural networks diagnose plant disease," *Plant Phenomics*, Vol. 2019, 2019. https://spj.science.org/doi/10.34133/2019/9237136

[26] Amanda Ramcharan et al., "Deep learning for image-based cassava disease detection," *Frontiers in Plant Science*, Vol. 8, pp. 1852, 2017. https://www.frontiersin.org/articles/10.3389/fpls.2017.01852

[27] Konstantinos P. Ferentinos, "Deep learning models for plant disease detection and diagnosis," *Computers and Electronics in Agriculture*, Vol. 145, pp. 311–318, 2018. https://doi.org/10.1016/j.compag.2018.01.009

[28] L. G. Nachtigall, R. M. Araujo, and G. R. Nightingale, "Classification of apple tree disorders using convolutional neural networks," *Proceedings of the IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 472–476, 2016. https://www.researchgate.net/publication/312486428_Classification_of_Apple_Tree_Disorders_Using_Convolutional_Neural_Networks

[29] Mohammed Brahimi, Khokha Boukhalfa, and Abdelkamel Moussaoui, "Deep learning for tomato diseases: Classification and symptoms visualization," *Applied Artificial Intelligence*, Vol. 31, No. 4, pp. 299–315, 2017. https://www.researchgate.net/publication/316976487_Deep_Learning_for_Tomato_Diseases_Classification_and_Symptoms_Visualization

[30] Jihen Amara, Bassem Bouaziz, and Alsayed Algergawy, "A Deep Learning-based Approach for Banana Leaf Diseases Classification," *Datenbanksysteme für Business,* *Technologie und Web (BTW 2017) - Workshopband*, pp. 79–88, 2017. https://dl.gi.de/items/13766147-8092-4f0a-b4e1-8a11a9046bdf

[31] Shujuan Zhang, Song Zhang, Chungui Zhang, Xiang Wang, and Yun Shi, "Cucumber leaf disease identification with global pooling dilated convolutional neural network," *Computers and Electronics in Agriculture*, Vol. 162, pp. 422–430, 2019. https://doi.org/10.1016/j.compag.2019.04.040

[32] Andreas Kamilaris and Francesc X. Prenafeta-Boldú, "Deep learning in agriculture: A survey," *Computers and Electronics in Agriculture*, Vol. 147, pp. 70–90, 2018. https://doi.org/10.1016/j.compag.2018.02.016

[33] Ramprasaath R. Selvaraju et al., "Grad-CAM: Visual explanations from deep networks via gradient-based localization," *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pp. 618–626, 2017. https://ieeexplore.ieee.org/document/8237336

[34] Emmanuel O. Asani et al., "mPD-APP: A mobile-enabled plant diseases diagnosis platform using deep learning," *Frontiers in Artificial Intelligence*, Vol. 6, 2023. https://www.frontiersin.org/articles/10.3389/frai.2023.1227950

[35] Abdullah Ahmad et al., "Comparative Study of MobileNet and ResNet for Watermelon Leaf Disease Classification Using Deep Learning," *TIN: Terapan Informatika Nusantara*, Vol. 6, No. 1, 2025. https://ejurnal.seminar-id.com/index.php/tin/article/view/7543

[36] Jackulin Jeena et al., "A comprehensive review on detection of plant disease using artificial intelligence techniques," *IEEE Access*, Vol. 10, pp. 1–18, 2022. https://ieeexplore.ieee.org/document/9696499

[37] S. I. Rimon et al., "PlantBuddy: An Android-based mobile application for plant disease detection using deep convolutional neural network," *Artificial Intelligence and Technologies, Springer*, pp. 275-285, 2022. https://link.springer.com/chapter/10.1007/978-981-16-6448-9_28

[38] B. R. Reddy et al., "Deep Learning Based Mobile Application for Automated Plant Disease Detection," *IEEE Access*, Vol. 13, pp. 107917–107925, 2025. https://ieeexplore.ieee.org/document/11039794

[39] Shafaq Abbas, Muhammad Attique Khan, et al., "Crops Leaf Diseases Recognition: A Framework of Optimum Deep Learning Features," *Computers, Materials & Continua*, Vol. 74, No. 1, pp. 2121–2136, 2023. https://www.techscience.com/cmc/v74n1/49776

[40] Amina Adadi and Mohammed Berrada, "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)," *IEEE Access*, Vol. 6, pp. 52138–52160, 2018. https://ieeexplore.ieee.org/document/8466590

6. Biodata with Picture

1. Himanshu Gaur (23BCY10127)

Himanshu Gaur a dedicated Computer Science Engineering undergraduate specializing in Cyber Security and Digital Forensics at VIT Bhopal University (Class of 2027). He possesses a robust blend of technical skills in penetration testing, artificial intelligence, and data analytics, backed by practical industry experience.

Professional Experience & Projects
Has actively applied his skills through diverse internships:
Cyber Security Intern (Rajasthan Police): Contributed to digital forensic investigations and OSINT-based fraud detection.
Pentesting Intern (Ceeras IT Services): Conducted vulnerability assessments and improved system hardening protocols.
Python Data Science Intern (Megaminds IT Services): Developed automated data solutions and custom research projects.
My key projects include AutoPent, an AI-powered penetration testing framework, and PhishDefender, an automated phishing response system, demonstrating his ability to merge AI with security operations.

Skills & Achievements
Technical Proficiency: Python, C++, Java, Metasploit, Splunk, TensorFlow, and Digital Forensics tools.
Achievements: Ranked in the Top 1% globally on TryHackMe (2025).
Leadership: Serves as Treasurer and Team Lead for the BitByBit Club, managing budgets and leading student activities.
Certifications: Holds certifications in Network Pen Testing, CompTIA Pentest+, and Python for Cyber Security.

2. **Abhinav Mehra (23BCY10015)**

I am Abhinav Mehra , a dedicated Computer Science undergraduate specializing in Cybersecurity at VIT Bhopal University. I possess a robust blend of technical skills in network security, digital forensics, and deep learning, backed by practical industry experience and advanced project development.

Professional Experience & Projects

I have actively applied my skills through internship roles and technical innovation:
Cybersecurity Intern: I focused on network defense by configuring 3+ firewall setups and analyzing over 50 network captures using Wireshark. I also conducted vulnerability assessments, discovering critical issues like SQL Injection and XSS using tools such as WebGoat and OWASP ZAP.
AutoPent: I developed an automated, AI-powered penetration testing framework that integrates tools like Nmap and Metasploit to handle the full testing lifecycle, from reconnaissance to reporting.
Facial Recognition System: I built a deep learning system using HOG and CNNs that achieved 92.08% accuracy, capable of real-time detection for security and ID verification scenarios.

Skills & Achievements

Technical Proficiency: My core stack includes Python, Bash, MySQL, Kali Linux, Wireshark, Metasploit, Burp Suite, and Autopsy.
Achievements: I ranked in the Top 5% globally on TryHackMe and successfully detected and remediated 3 critical XSS vulnerabilities on a live company website.
Leadership: I serve as a Core Member of the Metaversity Club, where I led PR initiatives for events like ARChase, attracting over 150 participants.

Certifications: I hold the Google Cybersecurity Professional Certificate (achieved with a 100% score) and am Microsoft Certified in Security, Compliance, and Identity Fundamentals (SC-900).

3. Supreety Jha (23BCY10150)

I am an undergraduate student at VIT Bhopal, steadily building my path in cybersecurity and technology. I enjoy understanding how systems work, how they break, and most importantly, how to secure them. My internships with the Bhopal Police Cyber Cell and Red Team gave me real exposure to cyber investigations, threat detection, and the practical side of security. These experiences strengthened my curiosity, adaptability, and ability to learn quickly in fast-moving situations.

At my university, I stay actively involved in clubs and events because I genuinely enjoy working with people and managing responsibilities. As part of the Linux Club's finance team, I have handled sponsor outreach for events like Silent Disco. I also help organize creative and technical events with WICYS and the Android Club, including Cyber Feud, Treasure Hunt, and 404 Fun Found. Additionally, I contribute to the design team of VITBMUN, where I get to explore my creative interests.

Academically, I am strengthening my skills in Java, DBMS, cloud fundamentals, and cybersecurity concepts. I am currently working on two self-driven projects—Guardian, focused on IoT cybersecurity, and PyLatex report generation.

I aspire to grow as a cybersecurity professional by combining technical knowledge with creativity, teamwork, and continuous learning.

4. **Tanya Bharti (23BCE11555)**

I am an aspiring software engineer and B.Tech student at VIT Bhopal, passionate about bridging Full-Stack Development, Data Science, and Cloud Computing. With a strong foundation in Java, Python, and DSA, I excel in solving complex problems through Competitive Programming.

My technical proficiency is backed by key certifications: AWS Cloud Practitioner Essentials, NPTEL Machine Learning, ETHNUS MERN Full Stack, and Coursera's Computer Networking.

Translating skills into action, I engineered "College Closet" using HTML, CSS, JavaScript, SQL, and MongoDB. Additionally, I developed "Fake News Detection" and "Alzheimer's Detection" models using Machine Learning, Data Cleaning, and EDA techniques. I combine robust backend architecture with UI/UX design principles and am eager to leverage my diverse skill set to contribute to impactful, collaborative projects in the tech industry.

5. **Rananjay Singh Chauhan (23BAI10080)**

I am Rananjay Singh Chauhan, a Computer Science Engineering undergraduate specializing in Artificial Intelligence and Machine Learning at Vellore Institute of Technology, Bhopal, with a GPA of 8.32. I bring a strong technical foundation to our team, with proficiency in programming languages such as Python, Java, and C++, and expertise in key ML tools like TensorFlow, PyTorch, Keras, and Scikit-Learn.

My practical experience involves developing complex solutions to real-world problems. I have built a Legal Document Sentiment Analyzer using transformers, designed the "AnoCheck" system for video anomaly detection, and created a deepfake detection pipeline that achieved 80-85% accuracy. I am also well-versed in Generative AI and latest prompt engineering paradigms.

Beyond my code, I hold certifications in AWS Cloud and Machine Learning Foundations. I value effective communication and collaboration and I am a team player.