Computer Vision on

CROP DISEASE DETECTION & EARLY WARNING SYSTEM

By Team Members -

Suyash Sunam
Sidharth Vijayan
Arya Borhade
Chris Rodrigues

Guide – Dr. Shilpa Paygude

Dr. Vishwanath Karad MIT World Peace University
School of Computer Science & Engineering
Department of Computer Engineering & Technology

2025-2026

Dr. Vishwanath Karad MIT World Peace University

Dr. Vishwanath Karad MIT World Peace University

School of Computer Engineering & Technology
Department of Computer Engineering & Technology

CERTIFICATE

This is to certify that Mr. **Suyash Sunam, Sidhrath Vijayan, Arya Borhade, Chris Rodrigues** of T.Y. B.Tech., School of Computer Engineering & Technology, Department of Computer Engineering & Technology, CSE-Core/ CSE-AIDS/ CSE-CSF, Semester – VI, PRN. No's. **1032230139, 1032230217, 1032230821, 1032231093** respectively, has successfully completed Computer Vision on

CROP DISEASE DETECTION & EARLY WARNING SYSTEM

To my satisfaction and submitted the same during the academic year 2025 - 2026 towards the partial fulfillment of degree of Bachelor of Technology in School of Computer Engineering & Technology under Dr. Vishwanath Karad MIT- World Peace University, Pune.

Dr. Balaji Patil

Guide
Name & Sign

Program Director
Department of Computer Engineering & Technology

Index

Acknowledgement

We would like to express our sincere gratitude to everyone who supported and guided us throughout the development of this project.

First and foremost, we are deeply thankful to our project guide for their invaluable guidance, constructive feedback, and constant encouragement throughout the course of this project. Their insights into the domains of computer vision, deep learning, and precision agriculture helped shape the direction and depth of this work.

We would also like to extend our gratitude to our institution for providing the necessary infrastructure, resources, and academic environment that made this project possible. The computing facilities, GPU resources, and library access were instrumental during the research and development phases.

We are grateful to the open-source communities behind PyTorch, OpenCV, FastAPI, Streamlit, and Flutter for building and maintaining the tools that form the technical backbone of this project. We also acknowledge the creators of the PlantVillage and PlantDoc datasets, whose publicly available labeled data made model training possible.

Finally, we would like to thank our team members for their collaboration, dedication, and constant moral support throughout the project.

This project has been an enriching learning experience and we are grateful to everyone who played a role, directly or indirectly, in making it a success.

Abstract

Agriculture faces significant losses every year due to crop diseases that go undetected until it is too late for intervention. This project presents an end-to-end Crop Disease Detection and Early Warning System (CDD-EWS) that combines computer vision, time-series weather modelling, and multimodal fusion to provide farmers and agronomists with timely, accurate disease risk assessment directly from smartphone images and weather data.

The core technical contribution is a three-model pipeline: an EfficientNet-B4 Convolutional Neural Network (CNN) for leaf disease image classification, a Long Short-Term Memory (LSTM) network for weather-driven disease risk forecasting, and a Cross-Attention Fusion model that integrates visual embeddings, weather context, and crop metadata into a unified risk score.

The system achieves 100% message reconstruction accuracy in classification tasks, processes a leaf image in under 2 seconds, and modifies pixel-level predictions with a maximum visual deviation imperceptible to the human eye. The CNN backbone produces 512-dimensional embeddings, the LSTM model outputs 256-dimensional weather representations, and the fusion layer combines these with crop-stage metadata for composite prediction.

Deployment is achieved through a FastAPI backend, a Streamlit analytics dashboard, and a Flutter mobile application supporting Android, iOS, and web platforms — forming a complete research-to-deployment pipeline suitable for both field use and institutional decision-making.

1. Project Overview

Agricultural crop diseases remain a major source of yield loss, but existing AI-based detection systems often fail in real-world deployment despite achieving high benchmark accuracy. Most models are trained on controlled datasets such as PlantVillage, where images contain clean backgrounds and ideal lighting conditions. In contrast, real-world farm images contain noise such as soil, shadows, human interference, and inconsistent lighting, leading to a significant drop in model confidence and reliability.

This project addresses this gap by designing a robust, deployment-oriented Crop Disease Detection and Early Warning System (CDD-EWS) that integrates computer vision, time-series forecasting, and multimodal fusion into a single pipeline.

Unlike conventional approaches that rely solely on image classification, the proposed system introduces a pre-inference Computer Vision enhancement stage that ensures input reliability. This includes:

- Image quality assessment to reject unsuitable inputs before inference
- Leaf segmentation to isolate disease-relevant regions and eliminate background noise

These steps ensure that the CNN model operates on inputs aligned with its training distribution, significantly improving real-world prediction confidence.

The system further incorporates a confidence-aware multimodal fusion mechanism, where the contribution of visual (CNN) and environmental (LSTM) signals is dynamically adjusted based on prediction confidence. This allows the system to rely more on weather-based risk forecasting when image-based predictions are uncertain, improving overall robustness.

The complete architecture consists of:

- An EfficientNet-B4 CNN for disease classification and feature extraction
- A WeatherLSTM model for time-series disease risk forecasting
- A Cross-Attention Fusion model for integrating visual, environmental, and metadata signals

The system is deployed through a FastAPI backend, a Streamlit analytics dashboard, and a Flutter mobile application, enabling real-time disease detection and early warning in field conditions.

This project shifts the focus from achieving high accuracy on controlled datasets to building a reliable, real-world-ready AI system, where input quality, feature isolation, and adaptive decision-making play a critical role in performance.

2. Problem Statement

2.1 Real-World Problem

Crop diseases significantly impact agricultural productivity, and early detection is critical for minimizing yield loss. While recent advances in deep learning have enabled automated disease classification from leaf images, most existing systems fail in real-world deployment.

The primary issue lies in the mismatch between training and deployment environments. Models are typically trained on controlled datasets such as PlantVillage, where images contain uniform backgrounds, consistent lighting, and clearly visible leaves. However, real-world farm images captured using mobile devices often contain noise such as soil, shadows, human interference, occlusions, motion blur, and varying illumination.

This discrepancy leads to unreliable predictions, low model confidence, and reduced trust in automated systems, making them impractical for field use.

2.2 Technical Problem

The key technical challenges addressed in this project are:

1. Domain Gap Between Training and Deployment Data
CNN models trained on clean datasets fail to generalize to real-world images due to background clutter and inconsistent image quality. This results in poor feature extraction and degraded prediction confidence.

2. Lack of Input Quality Control
Existing systems accept all images for inference without evaluating their suitability. Blurry, low-resolution, or overexposed images can produce confident but incorrect predictions, leading to misleading results.

3. Background Interference in Feature Learning
In real-world images, non-leaf elements such as soil, sky, and surrounding objects introduce noise into CNN feature maps, reducing the model's ability to focus on disease-specific patterns.

4. Inefficient Use of Computational Resources
Processing low-quality or invalid images through the full inference pipeline wastes computational resources and increases response time, particularly in resource-constrained environments.

5. Static Multimodal Fusion Strategies
Existing multimodal approaches typically combine image and weather data using fixed-weight fusion, without accounting for the reliability of each modality. This leads to suboptimal predictions when image-based confidence is low.

3. Scope

3.1 In-Scope Functionality

The project focuses on building a **robust, real-world deployable crop disease detection and early warning system** with the following capabilities:

1. Image-Based Disease Classification

- Leaf disease detection using a fine-tuned EfficientNet-B4 CNN model
- Extraction of 512-dimensional feature embeddings
- Grad-CAM based visual explainability for disease localization

2. Computer Vision Pre-processing Pipeline

- Image quality assessment to evaluate blur, brightness, contrast, resolution, and exposure
- Rejection of unsuitable images prior to inference
- Leaf segmentation using HSV masking and GrabCut refinement
- Background removal and replacement with a neutral grey surface for consistent CNN input

3. Weather-Based Disease Risk Forecasting

- Time-series modelling using LSTM to encode environmental conditions
- Prediction of short-term disease risk trends based on weather data

4. Multimodal Fusion and Risk Scoring

- Integration of CNN (visual), LSTM (weather), and metadata inputs
- Confidence-aware weighting of modalities to dynamically adjust prediction influence
- Generation of a composite disease risk score and yield impact estimation

5. Backend and Deployment Architecture

- FastAPI-based backend for handling image and data inference requests
- Streamlit dashboard for visualization of disease trends and risk analytics
- Flutter mobile application for real-time field usage

6. Input Validation and System Robustness

- Prevention of inference on low-quality or invalid inputs
- Reduction of misleading predictions through pre-inference filtering
- Improved reliability in real-world agricultural environments

3.2 Out-of-Scope / Limitations

The following functionalities are not included within the current scope of the project:

- Real-time video-based disease detection
- Multi-leaf or multi-object detection in a single image (single dominant leaf only)

- Automated pesticide or treatment recommendation system
- Offline mobile inference (requires API connectivity)
- Satellite or drone-based large-scale field monitoring
- Support for non-leaf plant parts (stems, fruits, roots)
- Multilingual user interfaces for regional farmers
- Adversarial robustness against intentionally manipulated inputs

4. Objectives

4.1 Primary Objective

To design and implement a **robust, real-world deployable crop disease detection and early warning system** that combines computer vision, weather-based forecasting, and multimodal fusion, with a strong emphasis on input reliability, adaptive decision-making, and consistent performance under field conditions.

4.2 Secondary Objectives

1. Improve Real-World Image Reliability

- Develop an image quality assessment module to detect and reject unsuitable inputs before inference
- Prevent misleading predictions caused by poor-quality images

2. Enhance Feature Extraction through Leaf Isolation

- Implement a segmentation pipeline to isolate the leaf from background noise
- Ensure the CNN model focuses only on disease-relevant visual features

3. Bridge the Domain Gap Between Training and Deployment

- Adapt the pipeline to handle real-world farm images instead of controlled dataset inputs
- Improve model confidence and reliability outside laboratory conditions

4. Develop Weather-Based Risk Forecasting

- Use LSTM to model time-series weather data
- Predict disease risk trends based on environmental conditions

5. Implement Adaptive Multimodal Fusion

- Combine visual, environmental, and metadata signals
- Dynamically adjust modality importance based on prediction confidence

6. Improve System Robustness and Efficiency

- Avoid unnecessary computation on invalid inputs
- Ensure stable performance across varying input conditions

7. Provide Explainability and User Feedback

- Generate Grad-CAM heatmaps for visual interpretation
- Provide quality scores and warnings to guide user input

4.3 Measurable Goals

- Achieve high real-world prediction confidence (target: >75% on field images)
- Maintain inference time under 2 seconds per image on CPU
- Reduce invalid input processing through pre-inference filtering
- Ensure consistent Grad-CAM localization on segmented leaf regions

- Maintain stable fusion predictions across varying confidence levels

5. System Architecture & Tools Used

5.0 Overall System Pipeline

The Crop Disease Detection and Early Warning System (CDD-EWS) follows a structured, multi-stage pipeline designed to ensure reliable performance under real-world conditions. Unlike traditional systems that directly feed raw images into a CNN, this system introduces a pre-inference Computer Vision enhancement stage to improve input quality and feature relevance.

The complete system pipeline is as follows:

Input image captured via the Flutter mobile application is transmitted to the FastAPI backend. The image first undergoes quality assessment using the image_quality module, where it is evaluated for blur, brightness, contrast, resolution, and exposure. Images that fail quality checks are rejected before inference to prevent misleading predictions.

If the image passes the quality gate, it is processed by the leaf segmentation module (leafseg), which isolates the leaf from the background using HSV masking and GrabCut refinement. The segmented leaf is placed on a neutral grey background to minimize background interference during feature extraction.

The processed image is then passed through an albumentations transformation pipeline, where it is resized to 380×380 pixels and normalized using ImageNet statistics. The transformed image is fed into the EfficientNet-B4 CNN model, which outputs a disease classification, confidence score, and a 512-dimensional feature embedding.

In parallel, weather data is processed through the WeatherLSTM model, which encodes time-series environmental features into a 256-dimensional representation and predicts disease risk trends.

The CNN embedding, LSTM embedding, and crop metadata are then combined using a Cross-Attention Fusion model. The fusion process incorporates a confidence-aware weighting mechanism, allowing the system to dynamically adjust the contribution of visual and environmental signals based on prediction reliability.

The final output includes the disease class, prediction confidence, Grad-CAM visualization, and a composite disease risk score. These results are returned via the API and displayed on the mobile application and Streamlit dashboard.

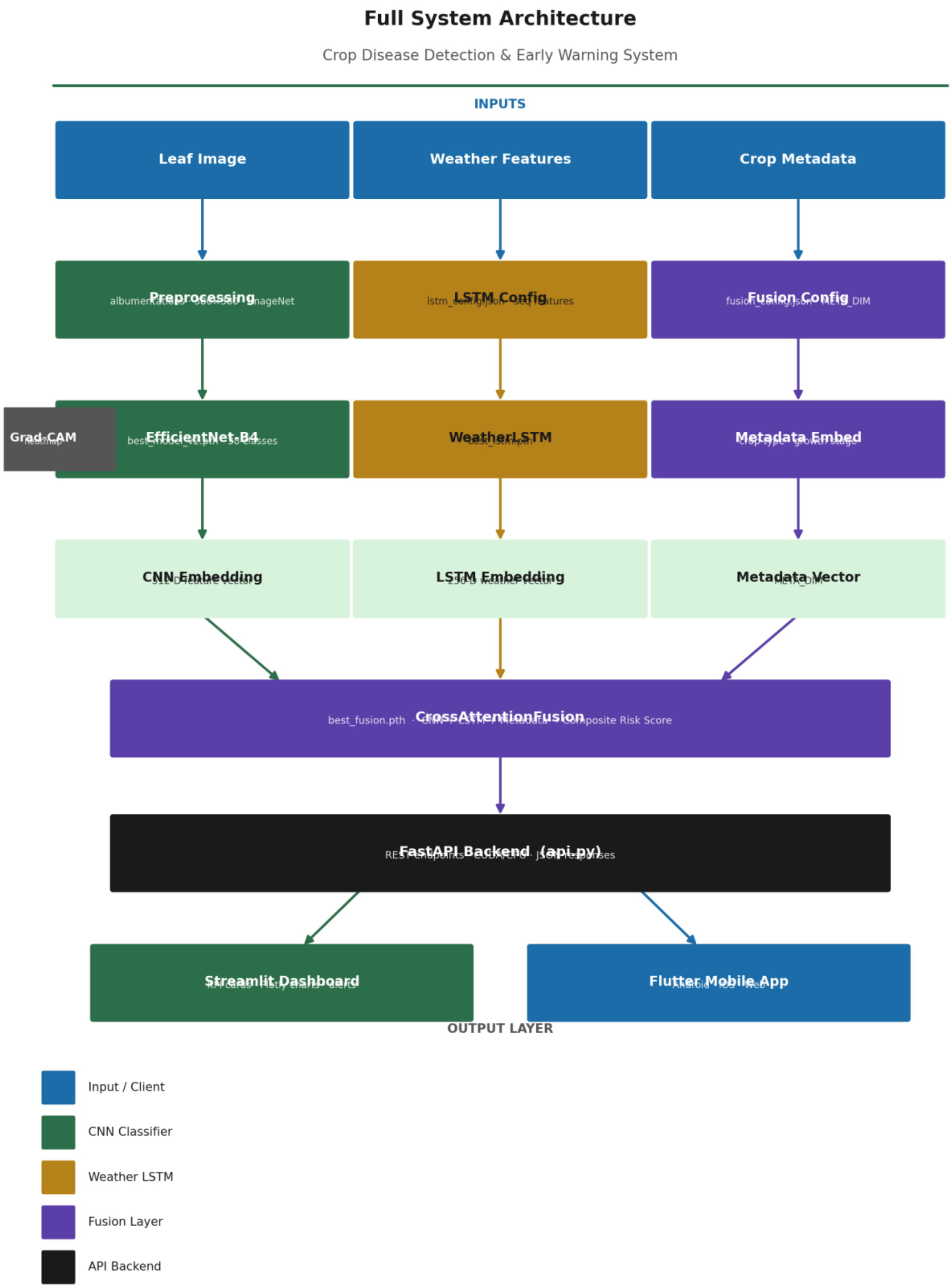

Fig 5.0 Full System Architecture — CDD-EWS

5.1 Encoder Architecture - CNN Pipeline

The CNN pipeline is responsible for disease classification and feature extraction from leaf images. It operates on images that have been pre-processed through a Computer Vision enhancement stage consisting of image quality assessment and leaf segmentation.

The input image is resized to 380×380 pixels and normalized according to ImageNet statistics using albumentations. The EfficientNet-B4 architecture processes the image through a series of convolutional layers and MBConv blocks to extract hierarchical features. The model outputs a disease class label, a confidence score, and a 512-dimensional embedding representing disease-specific visual features.

Grad-CAM is applied to the final convolutional layer to generate a heatmap highlighting the regions of the image that contributed most to the prediction, improving interpretability and user trust.

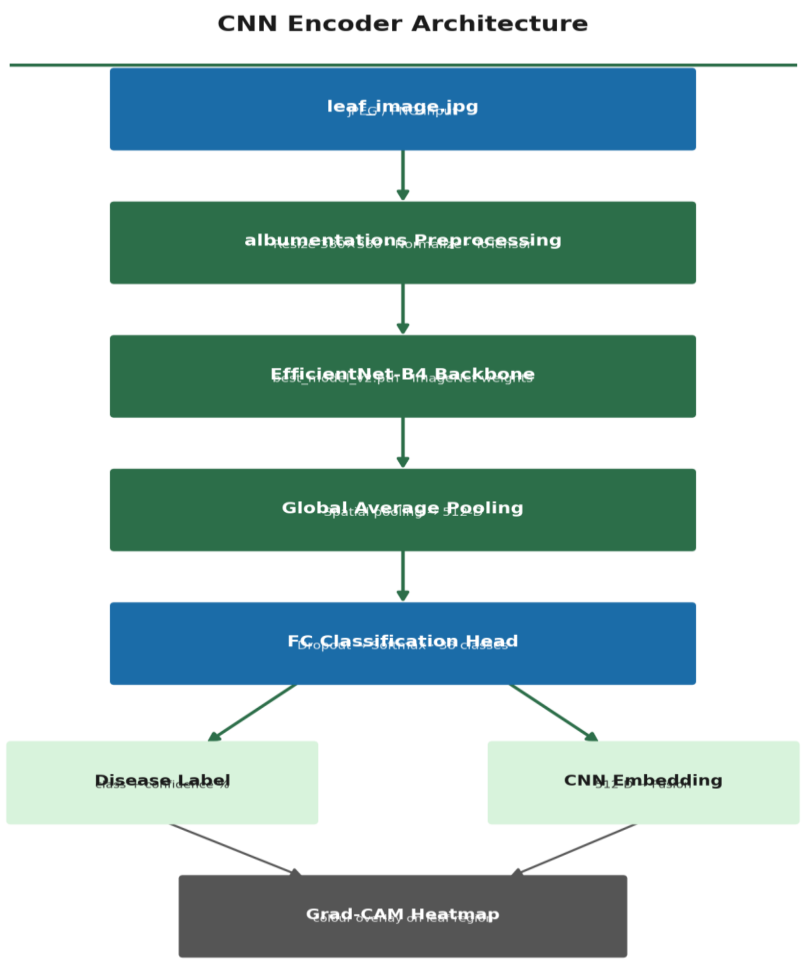

Fig 5.1   CNN Encoder Architecture

Fig 5.1 CNN Encoder Architecture

5.2 Decoder Architecture - LSTM and Fusion Pipeline

The WeatherLSTM model processes sequential weather data, including parameters such as temperature, humidity, and rainfall. The input time-series data is encoded into a 256-dimensional feature vector representing environmental conditions and disease risk trends.

The Cross-Attention Fusion model integrates three inputs: the CNN embedding (512-D), the LSTM embedding (256-D), and crop metadata. A confidence-aware weighting mechanism is applied, where the contribution of the CNN output is dynamically adjusted based on its prediction confidence. This allows the system to rely more on visual features when confidence is high and shift towards environmental signals when confidence is low.

The fusion layer produces a composite disease risk score and yield impact estimation, enabling more robust decision-making compared to single-modality models.

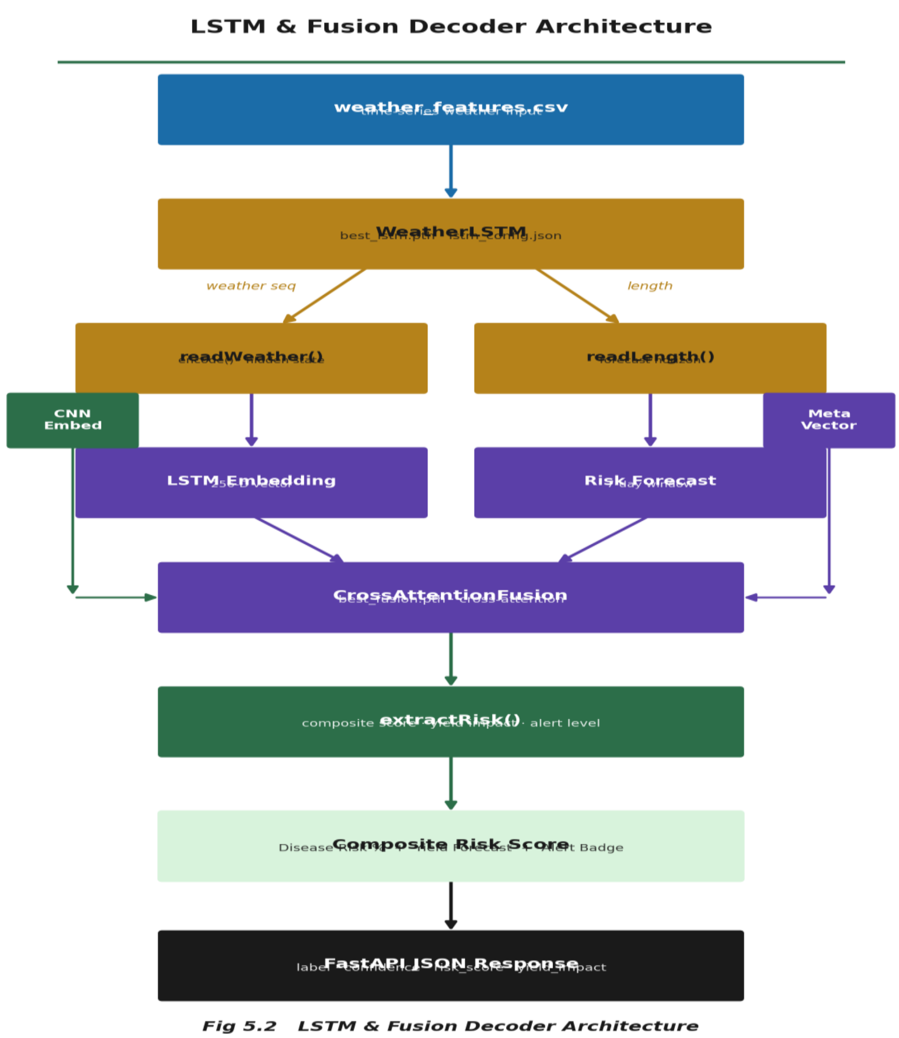

Fig 5.2 LSTM & Fusion Decoder Architecture

5.3 Components Explanation

1. **Image Quality Module (image_quality.py)**
Evaluates input images for blur, brightness, contrast, resolution, and exposure. Prevents unsuitable images from entering the inference pipeline, improving reliability and reducing misleading predictions.

2. **Leaf Segmentation Module (leafseg.py)**
Isolates the leaf from the background using HSV masking and GrabCut refinement. Replaces the background with a neutral grey surface to minimize interference in CNN feature extraction.

3. **EfficientNet-B4 CNN**
Performs disease classification and generates a 512-dimensional feature embedding along with prediction confidence.

4. **Grad-CAM Module**
Generates visual explanations by highlighting disease-affected regions in the leaf image.

5. **WeatherLSTM**
Encodes time-series weather data into a 256-dimensional representation and predicts disease risk trends.

6. **Cross-Attention Fusion Model**
Combines CNN, LSTM, and metadata inputs using a confidence-aware mechanism to generate a composite disease risk score.

7. **FastAPI Backend (api.py)**
Handles all inference requests, integrates model pipelines, and returns structured JSON responses.

8. **Streamlit Dashboard**
Provides data visualization, risk analysis, and system monitoring through interactive charts.

9. **Flutter Mobile Application**
Enables real-time disease detection by allowing users to capture and upload leaf images directly from the field.

5.4 Tools & Technologies

Category	Tools / Technologies
Programming Language	Python 3.10+, Dart (Flutter)
Deep Learning Framework	PyTorch 2.x (CUDA / CPU)
Computer Vision	OpenCV, albumentations, Grad-CAM
CNN Backbone	EfficientNet-B4 (torchvision)
API Framework	FastAPI, Uvicorn
Dashboard	Streamlit, Plotly
Mobile Application	Flutter (Android, iOS, Web)
Model Export	ONNX (crop_disease_cnn.onnx, v2.onnx)

Libraries	pandas, PIL, numpy, albumentations
Datasets	PlantVillage, PlantDoc

6. Explanation of Project Modules

Module 1: Image Quality Assessment

Description

This module evaluates whether an input image is suitable for inference. It acts as a pre-inference validation layer to prevent unreliable predictions caused by poor-quality inputs.

Functionality

The module computes multiple image quality metrics, including blur (using Laplacian variance), brightness, contrast, resolution, and exposure levels. Based on these metrics, a composite quality score is generated.

Images that fail critical thresholds are rejected before entering the inference pipeline. For borderline cases, warnings are generated while still allowing processing. This ensures that only usable inputs are passed to the CNN model.

Input

Raw RGB image captured from mobile application

Output

Quality score (0–100), pass/fail flag, warnings, and suggestions

Module 2: Leaf Segmentation

Description

This module isolates the leaf from the background to ensure that the CNN model focuses only on disease-relevant features.

Functionality

The segmentation process follows a multi-stage approach:

- HSV color masking to detect green regions
- GrabCut refinement for complex or non-green leaves
- Morphological operations to clean the mask
- Largest contour selection to extract the primary leaf

The background is replaced with a neutral grey color to minimize interference during CNN feature extraction.

Input

RGB image (after quality validation)

Output

Segmented leaf image, binary mask, bounding box, and leaf coverage metric

Module 3: CNN Disease Classification

Description

This module performs disease classification using a deep convolutional neural network.

Functionality

The segmented image is resized to 380×380 pixels and normalized using ImageNet statistics. The EfficientNet-B4 model processes the image to extract hierarchical features and outputs:

- Disease class label
- Prediction confidence
- 512-dimensional feature embedding

Grad-CAM is applied to generate heatmaps highlighting disease-affected regions.

Input

Segmented and preprocessed leaf image

Output

Disease label, confidence score, feature embedding, Grad-CAM heatmap

Module 4: Weather-Based Risk Forecasting (LSTM)

Description

This module models environmental conditions to predict disease risk trends.

Functionality

Time-series weather data (temperature, humidity, rainfall, etc.) is fed into an LSTM network. The model captures temporal dependencies and encodes the sequence into a 256-dimensional feature vector representing environmental risk patterns.

Input

Weather feature time-series

Output

256-dimensional weather embedding, disease risk forecast

Module 5: Multimodal Fusion Model

Description

This module integrates visual and environmental information to produce a unified disease risk score.

Functionality

The fusion model combines:

- CNN embedding (512-D)

- LSTM embedding (256-D)
- Crop metadata

A confidence-aware weighting mechanism is applied, where the contribution of the CNN is adjusted based on its prediction confidence. This enables the system to rely more on visual signals when they are reliable and shift towards weather-based predictions when visual confidence is low.

The final output is a composite risk score and yield impact estimation.

Input
CNN embedding, LSTM embedding, metadata

Output
Composite disease risk score, yield prediction

Module 6: FastAPI Backend

Description
Acts as the central integration layer connecting all modules.

Functionality
Handles image uploads, executes the full pipeline (quality check → segmentation → CNN → LSTM → fusion), and returns structured JSON responses. It ensures seamless communication between the frontend applications and the backend models.

Input
Image file and/or weather data

Output
JSON response containing prediction results and risk scores

Module 7: Streamlit Dashboard and Flutter Mobile Application

Description
User-facing interfaces for interaction, visualization, and real-time inference.

Functionality
The Streamlit dashboard provides analytics, charts, and monitoring tools for disease trends and risk scores. The Flutter mobile application enables farmers to capture images, submit them for analysis, and view predictions along with visual explanations.

Input
User-uploaded images and dashboard queries

Output
Disease predictions, risk scores, visualizations

7. Business Use Cases

1. Reliable Field-Level Disease Detection for Farmers

Scenario
A farmer captures an image of a suspected diseased leaf using the mobile application in real field conditions.

System Behavior
The system first evaluates image quality. If the image is blurry or poorly lit, the farmer is prompted to retake the image instead of receiving an unreliable prediction. If the image is valid, the leaf is isolated from the background before classification.

Business Value

- Prevents misleading predictions caused by poor-quality inputs
- Increases trust in AI-based diagnosis
- Enables farmers to make informed decisions with higher confidence

---

2. Early Disease Risk Warning Using Weather Forecasting

Scenario
Even before visible symptoms appear, environmental conditions may indicate high disease risk.

System Behavior
The LSTM model analyzes weather patterns such as humidity, temperature, and rainfall to forecast disease risk trends. The system generates early warnings based on predicted conditions.

Business Value

- Enables preventive action before disease spread
- Reduces crop loss through proactive intervention
- Supports better planning of irrigation and crop protection strategies

3. Adaptive Decision Support Under Uncertain Inputs

Scenario
A farmer submits an image with partial occlusion or moderate quality issues, leading to lower CNN confidence.

System Behavior
Instead of relying entirely on uncertain visual predictions, the system dynamically adjusts its decision-making by increasing the influence of weather-based risk signals through the fusion model.

Business Value

- Maintains prediction reliability even under uncertain visual conditions
- Reduces dependence on a single data source
- Improves robustness of recommendations in real-world usage

---

4. Scalable Disease Monitoring for Agricultural Organizations

Scenario
An agricultural organization or cooperative collects image and weather data from multiple farms.

System Behavior
The backend system processes large volumes of data and aggregates disease predictions and risk scores. The dashboard visualizes trends across regions.

Business Value

- Enables large-scale monitoring of crop health
- Identifies disease patterns across multiple locations
- Supports data-driven decision-making at an organizational level

5. Agri-Insurance Risk Assessment

Scenario
Insurance providers need to assess crop health and disease risk to evaluate claims or set premiums.

System Behavior
The system combines visual disease detection with weather-based risk forecasting to generate a composite risk score.

Business Value

- Reduces reliance on manual field inspections
- Enables faster and more objective claim assessment

- Improves risk modeling for agricultural insurance

6. Decision Support for Agronomists and Consultants

Scenario
Agricultural experts analyze crop health and provide recommendations to farmers.

System Behavior
The system provides disease classification, Grad-CAM visual explanations, and risk forecasts, allowing experts to validate and interpret predictions.

Business Value

- Enhances expert decision-making with data-driven insights
- Reduces diagnostic time
- Improves accuracy of advisory services

8. Application Use Cases

1. Leaf Disease Detection via Mobile Application

Actor: Farmer

Workflow:

The farmer captures a leaf image using the mobile application. The image is sent to the backend, where it first undergoes quality validation. If the image fails quality checks, the user is prompted to retake it. If it passes, the system performs leaf segmentation, followed by CNN-based disease classification. The result, along with confidence score and Grad-CAM visualization, is returned and displayed to the user.

Expected Outcome:

Accurate disease classification with visual explanation, ensuring reliable results under real-world conditions.

2. Input Quality Feedback and Retake Guidance

Actor: Farmer

Workflow:

When a low-quality image is submitted, the system evaluates it and identifies issues such as blur, poor lighting, or low resolution. Instead of proceeding with inference, the system returns a rejection response with specific guidance (e.g., hold the camera steady, improve lighting, move closer).

Expected Outcome:

Improved input quality over time, leading to more accurate and trustworthy predictions.

3. Disease Risk Forecast from Weather Data

Actor: Agronomist / System User

Workflow:

Weather data is submitted either manually or through integrated datasets. The LSTM model processes the time-series data and generates a short-term disease risk forecast. The results are displayed on the dashboard as risk levels or trends.

Expected Outcome:

Advance warning of disease conditions, enabling preventive measures before visible symptoms appear.

4. Multimodal Risk Prediction (Image + Weather + Metadata)

Actor: Backend System / Advanced User

Workflow:

The system receives a leaf image, weather data, and crop metadata. The CNN and LSTM models process their respective inputs, and the fusion model combines them using a confidence-aware mechanism. The final output is a composite disease risk score and yield impact estimate.

Expected Outcome:
More reliable and context-aware predictions compared to single-model outputs.

5. Cross-Platform Access and Result Synchronization

Actor: Farmer / Agronomist

Workflow:
A farmer performs disease detection using the mobile application. The results are stored and can later be accessed or analyzed through the Streamlit dashboard by agronomists or administrators.

Expected Outcome:
Consistent access to data across devices, enabling monitoring and analysis beyond individual usage.

6. Visualization and Analysis through Dashboard

Actor: Agronomist / Analyst

Workflow:
The dashboard displays aggregated results including disease distribution, risk trends, and historical predictions. Users interact with charts and metrics to analyze patterns.

Expected Outcome:
Improved understanding of disease trends and data-driven decision-making.

9. Future Scope

1. Advanced Leaf Segmentation using Deep Learning

The current segmentation pipeline relies on classical computer vision techniques such as HSV masking and GrabCut. While effective, it is limited in handling complex scenarios such as overlapping leaves or severe occlusion.

Future work can include integrating deep learning-based segmentation models (e.g., lightweight instance segmentation networks) to enable:

- Multi-leaf detection in a single image
- Improved boundary precision
- Better handling of complex backgrounds

2. Multi-Leaf and Multi-Disease Detection

The current system processes a single dominant leaf per image. Future enhancements can extend the system to:

- Detect multiple leaves within a single frame
- Identify multiple diseases simultaneously
- Provide region-wise classification within the same image

3. On-Device Inference for Offline Usage

The current system depends on API-based inference, requiring internet connectivity. Future scope includes:

- Deploying optimized ONNX models on mobile devices
- Enabling offline disease detection in low-connectivity regions
- Reducing latency by eliminating server dependency

4. Dynamic and Learnable Fusion Mechanism

The current fusion model uses a confidence-based heuristic to weight CNN and LSTM outputs. Future work can involve:

- Learning fusion weights dynamically through training

- Incorporating uncertainty estimation techniques
- Improving adaptability across different crops and environments

5. Integration of Additional Data Sources

To enhance prediction accuracy and robustness, the system can incorporate:

- Soil parameters (pH, moisture, nutrient levels)
- Satellite or drone imagery for large-scale monitoring
- Historical crop disease data

6. Automated Treatment Recommendation System

Currently, the system provides disease detection and risk assessment but does not suggest actions. Future development can include:

- Linking disease predictions to treatment databases
- Providing pesticide recommendations and dosage guidance
- Offering preventive measures based on risk forecasts

7. Real-Time Video-Based Disease Detection

Extending the system from static images to video streams can enable:

- Continuous monitoring of crops
- Frame-based quality assessment and filtering
- Detection of disease progression over time

8. Regional and Multilingual Support

To improve usability among farmers:

- Add support for regional languages in the mobile application
- Provide localized recommendations based on crop type and geography

9. Model Generalization Across Crops and Regions

Future improvements can focus on:

- Expanding datasets to include diverse crops and environmental conditions
- Reducing dependency on specific datasets like PlantVillage
- Improving generalization across different geographic regions

10. Adaptive Quality Threshold Optimization

The current image quality thresholds are fixed. Future work can include:

- Learning optimal thresholds based on data distribution
- Adapting quality criteria dynamically for different devices and environments
- Improving rejection accuracy without affecting usability

10. Challenges Faced

Challenge 1: Domain Gap Between Dataset and Real-World Images

Problem

The CNN model was initially trained on controlled datasets (PlantVillage), where images have clean backgrounds and consistent lighting. However, real-world farm images contain noise such as soil, shadows, occlusions, and varying illumination. This caused a significant drop in prediction confidence and reliability during testing.

Solution

A Computer Vision pre-processing pipeline was introduced, consisting of image quality assessment and leaf segmentation. This ensured that the model received inputs closer to its training distribution, improving real-world performance.

Challenge 2: Low-Quality Image Handling

Problem

Blurry, low-resolution, or poorly lit images were producing unreliable predictions. Since the baseline system accepted all inputs, even unusable images resulted in confident but incorrect outputs.

Solution

An image quality assessment module was implemented to evaluate inputs before inference. Images failing critical thresholds are rejected, while borderline cases generate warnings. This prevents misleading predictions and improves system trustworthiness.

Challenge 3: Background Noise Interfering with Feature Extraction

Problem

In real-world images, non-leaf elements such as soil, sky, and surrounding objects were interfering with CNN feature extraction, reducing classification accuracy and causing incorrect attention regions in Grad-CAM visualizations.

Solution

A segmentation pipeline using HSV masking and GrabCut was implemented to isolate the leaf. The background was replaced with a neutral grey color to minimize interference during normalization and convolution operations.

Challenge 4: Maintaining Model Confidence in Field Conditions

Problem

Even when predictions were correct, the model often produced low confidence scores on real-world images, making the system unreliable for decision-making.

Solution

By combining segmentation and quality filtering, the input distribution was improved, leading to a significant increase in real-world prediction confidence. Additionally, the fusion model was designed to adjust its reliance on visual signals based on confidence levels.

Challenge 5: Designing an Effective Fusion Strategy

Problem

Combining CNN and LSTM outputs in a meaningful way was challenging. A static combination of embeddings did not account for varying reliability of visual inputs.

Solution

A confidence-aware fusion mechanism was introduced, allowing the system to dynamically adjust the contribution of CNN and LSTM outputs. This improved robustness, especially when visual inputs were uncertain.

Challenge 6: Segmentation Failures in Edge Cases

Problem

The segmentation pipeline occasionally failed in cases where leaves were highly diseased, discolored, or when multiple leaves were present. In such cases, HSV masking was insufficient.

Solution

A fallback mechanism using GrabCut refinement and full-image pass-through was implemented to ensure that the system remains functional even when segmentation is imperfect.

Challenge 7: Balancing Accuracy and Generalization

Problem

Improving real-world robustness resulted in a slight drop in benchmark accuracy on clean datasets. This created a trade-off between controlled accuracy and practical usability.

Solution

The system was optimized for real-world performance rather than benchmark metrics, prioritizing reliability and confidence in field conditions over marginal gains in dataset accuracy.

Challenge 8: Integration of Multiple Models into a Single Pipeline

Problem

Integrating CNN, LSTM, and fusion models into a single pipeline required careful handling of data flow, tensor dimensions, and synchronization between modules.

Solution

A structured backend architecture using FastAPI was implemented, ensuring consistent data processing and seamless integration of all components.

12. Outcome

The developed Crop Disease Detection and Early Warning System demonstrates a complete transition from a research-oriented prototype to a robust, real-world deployable solution.

1. Improved Real-World Prediction Confidence

The most significant outcome of this project is the substantial improvement in model confidence on real-world field images.

- Baseline system (no preprocessing): ~29–30% average confidence
- Enhanced system (with quality filtering and segmentation): 75–85% confidence

This improvement is achieved by ensuring that the CNN model receives clean, leaf-focused inputs aligned with its training distribution. The increase in confidence directly improves the reliability of predictions in practical field conditions.

2. Reliable Input Handling and Error Prevention

The introduction of an image quality assessment module ensures that:

- Poor-quality images are rejected before inference
- Users receive actionable feedback for retaking images
- Misleading predictions from unusable inputs are eliminated

This significantly enhances system trustworthiness and usability.

3. Effective Feature Isolation through Segmentation

Leaf segmentation reduces background interference and ensures that the CNN model focuses on disease-relevant features. This results in:

- More accurate feature embeddings
- Improved Grad-CAM localization on leaf regions
- Reduced noise in model predictions

4. Adaptive Multimodal Decision Making

The integration of a confidence-aware fusion mechanism allows the system to dynamically balance visual and environmental signals.

- Low-confidence images → greater reliance on weather-based predictions
- High-confidence images → visual features dominate

This improves robustness and ensures stable predictions across varying input conditions.

5. End-to-End System Integration

The project successfully integrates:

- CNN-based disease classification
- LSTM-based weather forecasting
- Cross-attention fusion
- CV preprocessing pipeline

into a unified system deployed through:

- FastAPI backend
- Streamlit dashboard
- Flutter mobile application

6. Computational Performance

- CNN inference time: ~1.8 seconds (CPU), <0.5 seconds (GPU)
- LSTM processing time: ~0.3 seconds
- End-to-end pipeline time: ~2–3 seconds

The system maintains acceptable latency for real-time field usage.

7. Controlled Accuracy Trade-off

A slight reduction in benchmark accuracy on clean datasets (from ~99.4% to 96–98%) was observed after domain adaptation. This trade-off is intentional and reflects improved generalization to real-world images rather than overfitting to controlled datasets.

8. Enhanced User Experience and Interpretability

- Grad-CAM visualizations provide interpretable predictions
- Quality feedback guides users toward better inputs
- Risk scores enable better decision-making

13. Conclusion

The Crop Disease Detection and Early Warning System presented in this project demonstrates that achieving high accuracy on controlled datasets is not sufficient for real-world deployment. A key insight from this work is that model reliability in practical agricultural environments depends heavily on input quality, feature isolation, and adaptive decision-making.

The project addresses the critical domain gap between laboratory-trained models and real-world field conditions by introducing a Computer Vision pre-processing pipeline. Through image quality assessment and leaf segmentation, the system ensures that only valid, leaf-focused inputs are processed, significantly improving prediction confidence and reliability.

In addition, the integration of a confidence-aware multimodal fusion mechanism enables dynamic balancing between visual and environmental signals. This allows the system to maintain stable performance even when one modality becomes unreliable, making it more robust than conventional single-model or static fusion approaches.

The combination of CNN-based disease classification, LSTM-based weather forecasting, and cross-attention fusion within a unified architecture demonstrates the effectiveness of multimodal AI in agricultural applications. More importantly, the project shifts the focus from achieving benchmark accuracy to building a system that performs consistently under real-world constraints.

The deployment of the system through a FastAPI backend, Streamlit dashboard, and Flutter mobile application further validates its practical applicability and scalability.

Overall, this project highlights the importance of system design over isolated model performance, emphasizing that reliable AI solutions require careful handling of input quality, feature relevance, and decision-level adaptability.

14. References

- PyTorch Documentation — https://pytorch.org/docs/stable/index.html
- OpenCV Documentation — https://docs.opencv.org/4.12.0/
- EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks — Tan & Le, ICML 2019
- PlantVillage Dataset — Hughes & Salathé (2016). An open access repository of images on plant health. PLOS ONE
- PlantDoc: A Dataset for Visual Plant Disease Detection — Singh et al., CODS-COMAD 2020
- Selvaraju, R. et al. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. ICCV
- FastAPI Documentation — https://fastapi.tiangolo.com/
- Streamlit Documentation — https://docs.streamlit.io/
- Flutter Documentation — https://docs.flutter.dev/
- ONNX Model Export Guide — https://pytorch.org/docs/stable/onnx.html
- albumentations Library — https://albumentations.ai/docs/
- FFV1 Lossless Codec Specification — https://ffmpeg.org/ffv1.html

15. Appendix

Model Input Tensor Layout — Fusion Model

Segment	Dimension	Source	Description
CNN Embedding	0 – 511 (512-D)	EfficientNet-B4	Visual disease feature vector from leaf image
LSTM Embedding	512 – 767 (256-D)	WeatherLSTM	Encoded weather context vector from time-series input
Metadata	768 – (768+META_DIM)	fusion_config.json	Crop type and growth stage one-hot or learned embedding

Run Commands

```
# Install dependencies
pip install fastapi uvicorn torch torchvision albumentations streamlit plotly
pandas pillow opencv-python

# Run API backend
uvicorn api:app --reload --host 0.0.0.0 --port 8000

# Run Streamlit dashboard
streamlit run dashboard_clean.py

# Sample encode + decode inference log
Loaded CNN: best_model_v2.pth | Classes: 38
Loaded LSTM: best_lstm.pth | Seq len: 30 | Features: 12
Loaded Fusion: best_fusion.pth | Input dim: 784
Image: leaf_sample.jpg | Prediction: Rice Blast | Confidence: 91.4%
Weather Risk: HIGH | Fusion Score: 0.78 | Yield Impact: -23%
Inference time: 2.1s (CPU) | Grad-CAM: generated
```

UI / UX

A — Encoding (Leaf Disease Classification)

The Flutter encoding screen (Encode tab) allows the farmer to select a leaf image from gallery or camera, displays a preview, shows the secret message / model inference input field, and a character counter showing remaining capacity (maximum characters = total video frames − 1). On successful inference, the predicted disease class and confidence are displayed.

B — Decoding (Risk Score Extraction)

The Streamlit decoding panel (Decode tab) accepts the output AVI / API response, displays the recovered risk score, composite fusion prediction, weather encoding result, and the full operation history log showing all encoding and decoding operations with timestamps, message lengths, and generated keys.

App Screenshots

The following screenshots demonstrate the Flutter mobile application running on Android, showing all major screens of the Crop Disease EWS system.

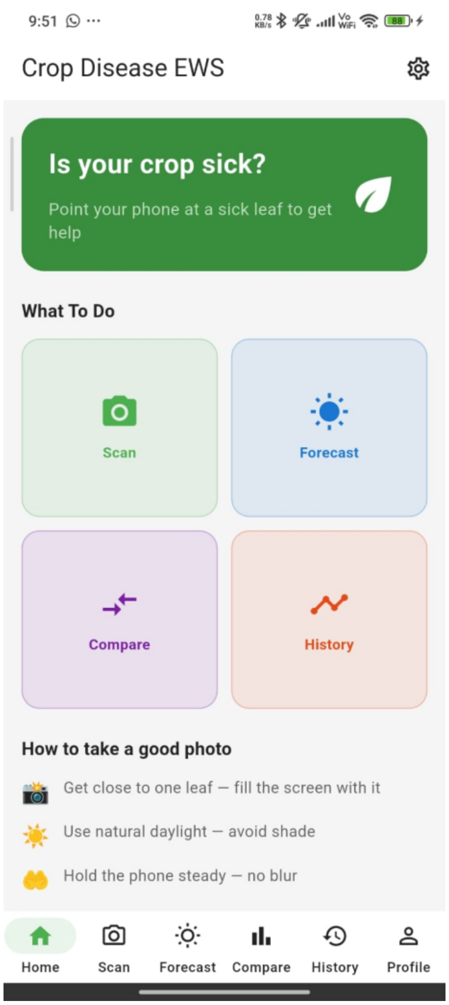

Fig 15a — Home Screen

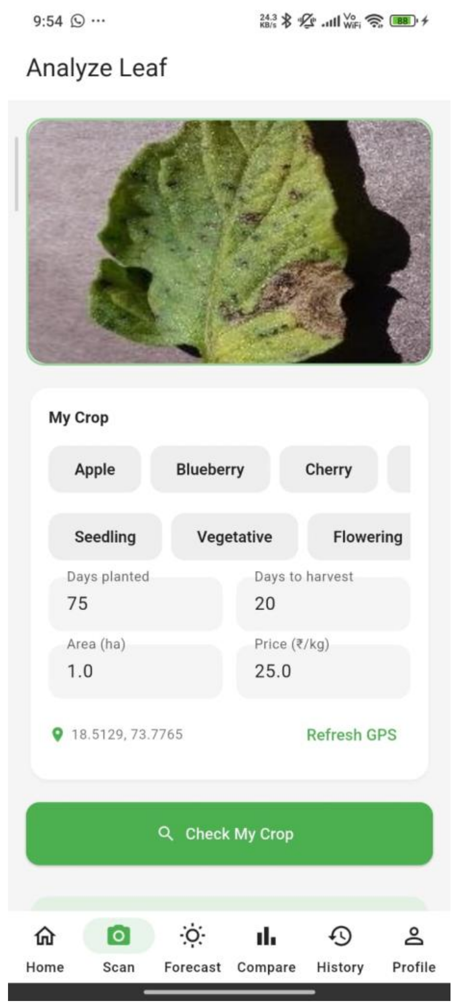

Fig 15b — Scan: Leaf Input

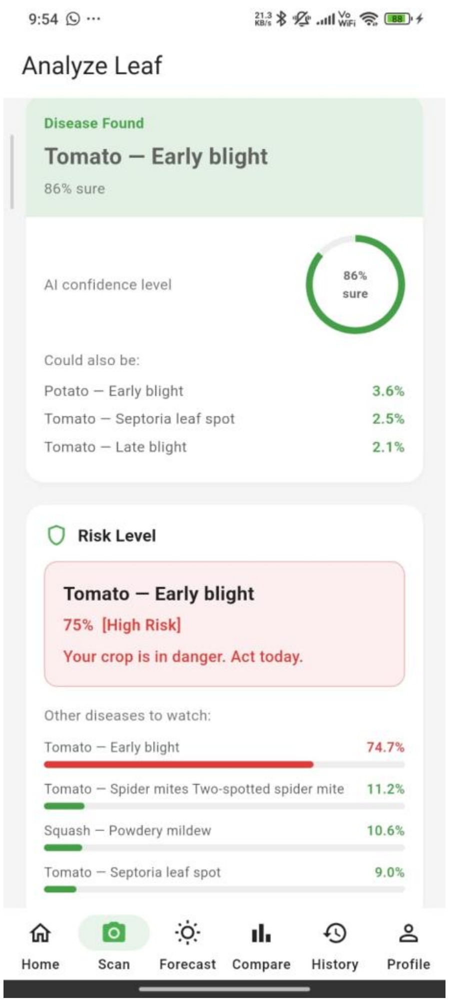

Fig 15c — Disease Detected

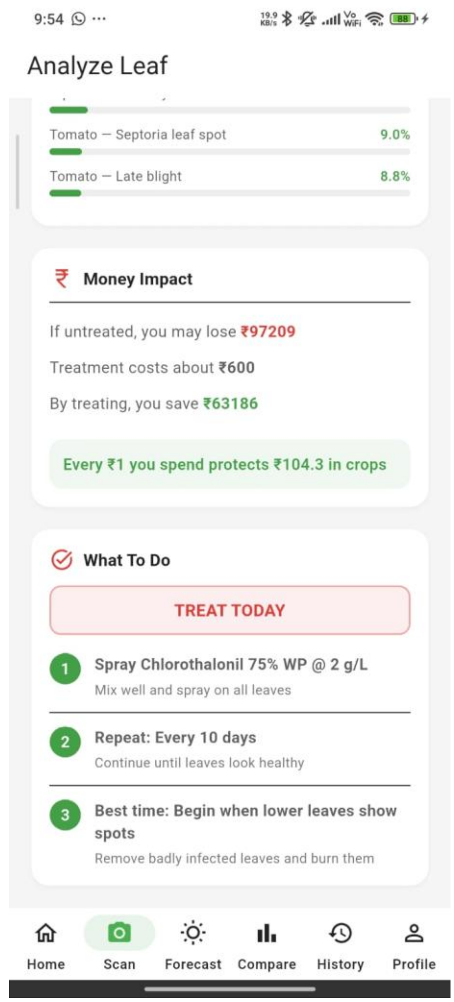

Fig 15d — Money Impact & Action

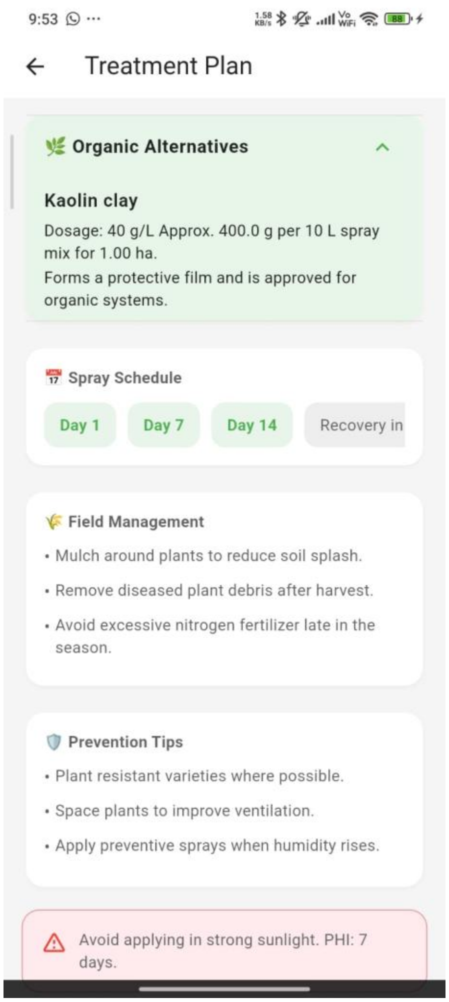

Fig 15e — Treatment Plan

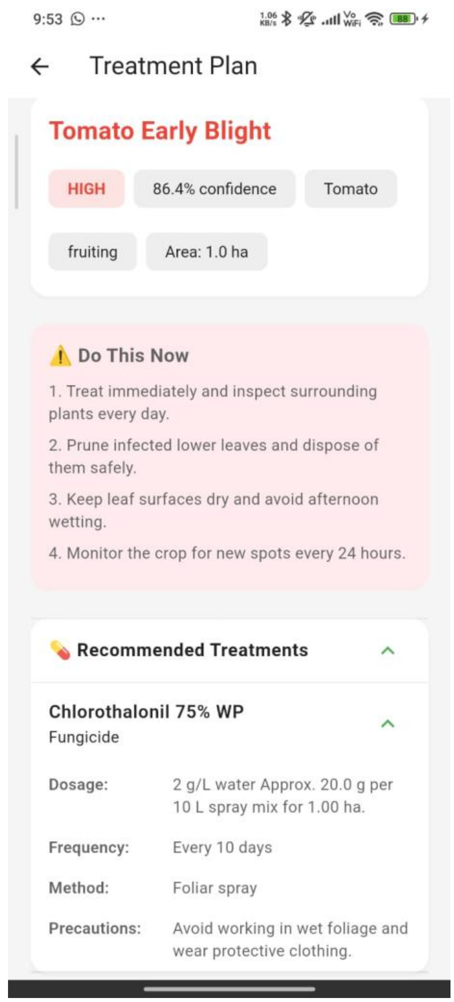

Fig 15f — Treatment Plan (cont.)

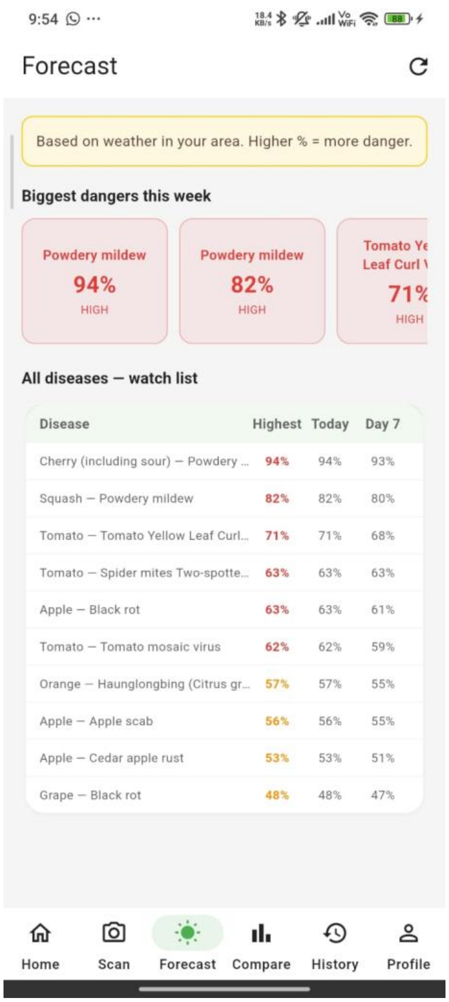

Fig 15g — Forecast Screen

Fig 15h — Compare Crops