PLANT AI — Leaf and Disease Identification

Using Deep Learning and Image Classification

An AI-powered system designed to identify plant leaves and detect diseases using deep learning techniques. The model analyzes leaf images to provide accurate classification and assist farmers, students, and researchers.

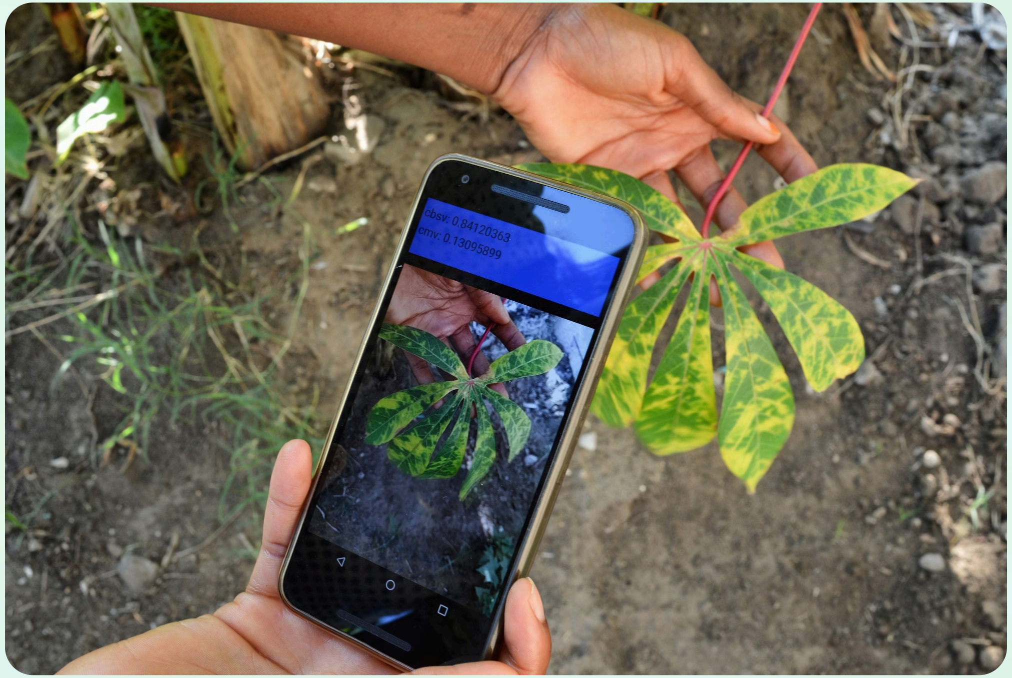
Presented by:

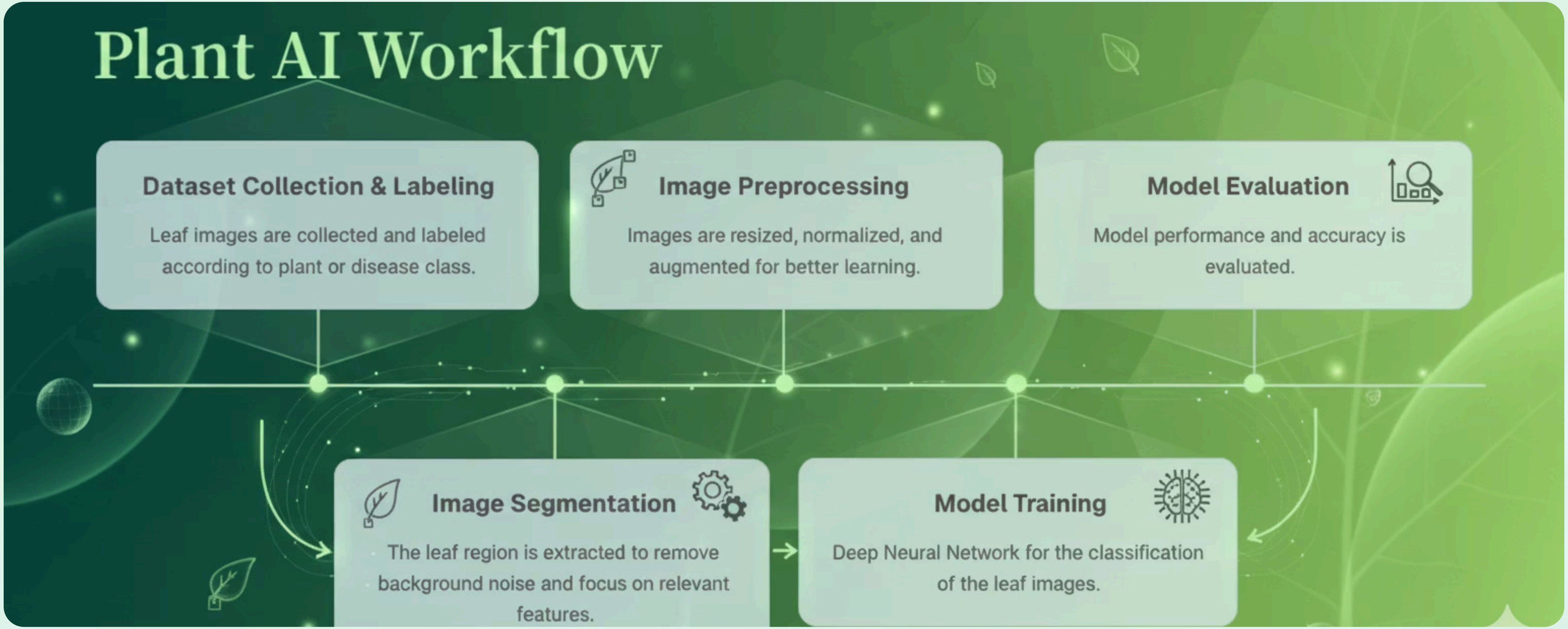
- Sohini
- Milan
- Prachi
- Pratik

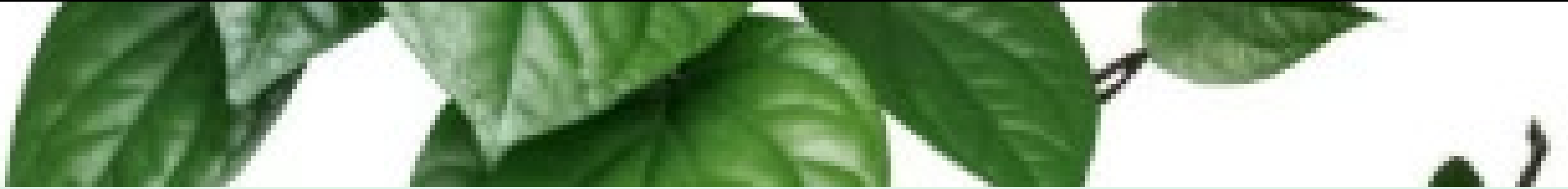
INTRODUCTION

PROBLEM STATEMENT

Accurate identification of plant species and detection of leaf diseases remains challenging due to ***fine-grained visual similarities***, ***environmental variations***, *and* ***background interference***. *Manual identification and disease diagnosis are* ***time-consuming***, ***expert-dependent***, *and* ***not scalable*** *for agricultural or educational use. Therefore, there is a need for an* ***automated deep learning–based system*** *capable of* ***classifying plant species***, ***detecting leaf diseases***, *and providing* ***confidence prediction scores***. *The system should be* ***accessible via mobile and web platforms***, *support* ***real-time image analysis***, *and enable* ***efficient decision-making*** *for farmers, researchers, and students to reduce dependency on expert intervention and minimize delays in treatment.*




Plant AI Workflow
Dataset Collection & Labeling
Leaf images are collected and labeled according to plant or disease class.
Image Preprocessing
Images are resized, normalized, and augmented for better learning.
Model Evaluation
Model performance and accuracy is evaluated.
Image Segmentation
The leaf region is extracted to remove background noise and focus on relevant features.
Model Training
Deep Neural Network for the classification of the leaf images.



PERFORMANCE

Segmentation Steps : To isolate the leaf region from complex backgrounds.

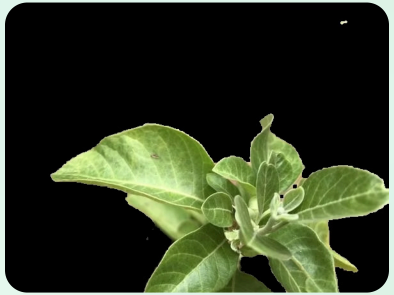
1. Resize and normalize the input image to make it model-ready.
2. The U-Net model predicts which pixels belong to the leaf.
3. Convert the model output into a binary mask
4. Apply the mask to the original image to remove the background.

Various Approaches for Segmentation

Original Image

Heatmap generated

Leaf Mask

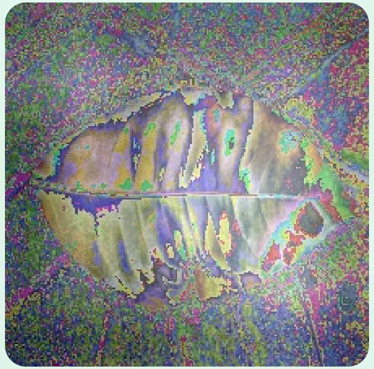


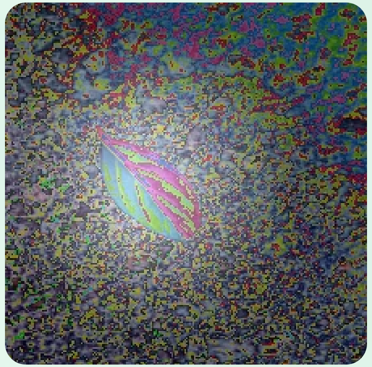
Leaf only image

Heatmap-based segmentation fails when pixel-level accuracy is required, as it provides only coarse and approximate localization.

HSV-based segmentation fails under varying lighting and colour conditions because it relies on fixed thresholds rather than learned features (green objects get wrongly segmented as leaves)

Various Approaches for Segmentation



Grad-CAM fails for segmentation because it does not provide pixel-level object masks. **Grad-CAM explains *where* the model looks, not *what* the object is.** It highlights discriminative regions, not the entire leaf. It visualizes the regions that contribute most to a CNN's prediction by generating a class-specific heatmap.

HSV and Grad-CAM were evaluated as baseline approaches but were replaced by **U-Net** due to their inability to perform precise leaf segmentation.

YOLO vs U-NET

We worked on both the models. Since accurate **leaf–background separation was prioritized** over real-time speed, U-Net was selected as the final segmentation model. **U-Net = Precision + Clean masks**

YOLO-Seg = Speed + Coarser masks

YOLO

- Real-time leaf detection
- Multiple leaves per image
- Mobile or edge deployment
- YOLO-Seg requires **larger datasets** for stable training.

U-Net

- High-precision leaf boundaries
- Clean background removal
- Offline or batch processing
- U-Net performs effectively with **smaller annotated datasets**.

In future work, the system can be extended by switching from U-Net–based segmentation to YOLO-based segmentation to support real-time leaf detection, especially for mobile and edge deployment.

However, these techniques failed for segmenting leaves with diseases. Here's why -

Method	Why it struggled in Disease part
HSV	Disease colors overlap with background hues, yellow - brown spots considered as background
Grad-CAM / Attention maps	Highlight regions, not pixel-accurate segmentation
U-Net	It just classifies through pixels, if there is a different pixel inside leaf, it considers that as backround.
SAM	Sometimes over-segments or misses diseased leaf - segments background, make leaf vanish
SAM - CENTER	Could only segment if leaf boundaries were totally visible inside the image

HSV Based

SAM (Segment Anything Model)

SAM failed example

SAM - center

GrabCut - Disease perfect method

GrabCut removes the background by learning what the leaf looks like and repeatedly refining the cut until only the leaf (with disease) remains.

Instead of classifying pixels once, GrabCut **learns and improves** the segmentation over multiple iterations.

Leaves are one big connected object

- GrabCut prefers **one solid object**
- It avoids breaking the leaf into pieces
- Disease areas remain connected to the leaf body

MODEL TRAINING

Why Our First Ideas Fell Short

We didn't just stumble upon our breakthrough. We explored many paths, testing different approaches to find what truly worked for identifying plant. Here's what we discovered didn't quite hit the mark:

Simple CNNs

These classic networks just couldn't handle the tricky details of plant. We needed something much smarter to spot the subtle differences between many plant types.

Training Deep Neural Network From Scratch

Building a deep network neural network from the ground up proved far too demanding. It devoured computational power and required an ocean of labeled data—resources we simply didn't have for our project.

YOLO for Classification

While YOLO excels at finding objects, it wasn't the right tool for classification. It struggled with the fine-grained distinctions needed to accurately classify similar leaf.

MODEL TRAINING

Challenges in Achieving Higher Accuracy

As we refine our models, several key challenges have emerged that impact the overall accuracy and precision of our plant identification system:

Dataset Limitations

Our current dataset predominantly features plant images with similar characteristics, which restricts the model's ability to accurately identify adult plants or those from diverse geographical regions.

Lighting Variation Challenges

Inconsistent lighting conditions significantly hinder image segmentation, potentially leading to inaccuracies in feature extraction and erroneous diagnoses.

Computational Constraints

Training a robust model from scratch on extensive datasets demands substantial hardware resources and is a highly time-consuming process, posing a considerable development hurdle.

Model Overfitting

The model currently exhibits signs of overfitting, performing exceptionally well on familiar data but struggling to maintain accuracy when presented with random or novel images.

Similar Leaf Structures

Plants possessing highly similar leaf structures, such as papaya and castor or neem and curry, frequently cause the model to 'hallucinate' or confuse identifications, resulting in incorrect classifications.

Final Approach

Gathering Rich Dataset:

We built a massive, diverse dataset from global sources like iNaturalist and PlantNet.

Data augmentation:-

We digitally transform images (rotate, mirror, adjust contrast) to mimic real-world conditions.

Accelerated Learning with Transfer Learning:

This approach dramatically speeds up development and helps our model generalize effectively.

Precision Training & Prevention of Overfitting:

We custom-trained our model with incredible precision to prevent it from memorizing data.

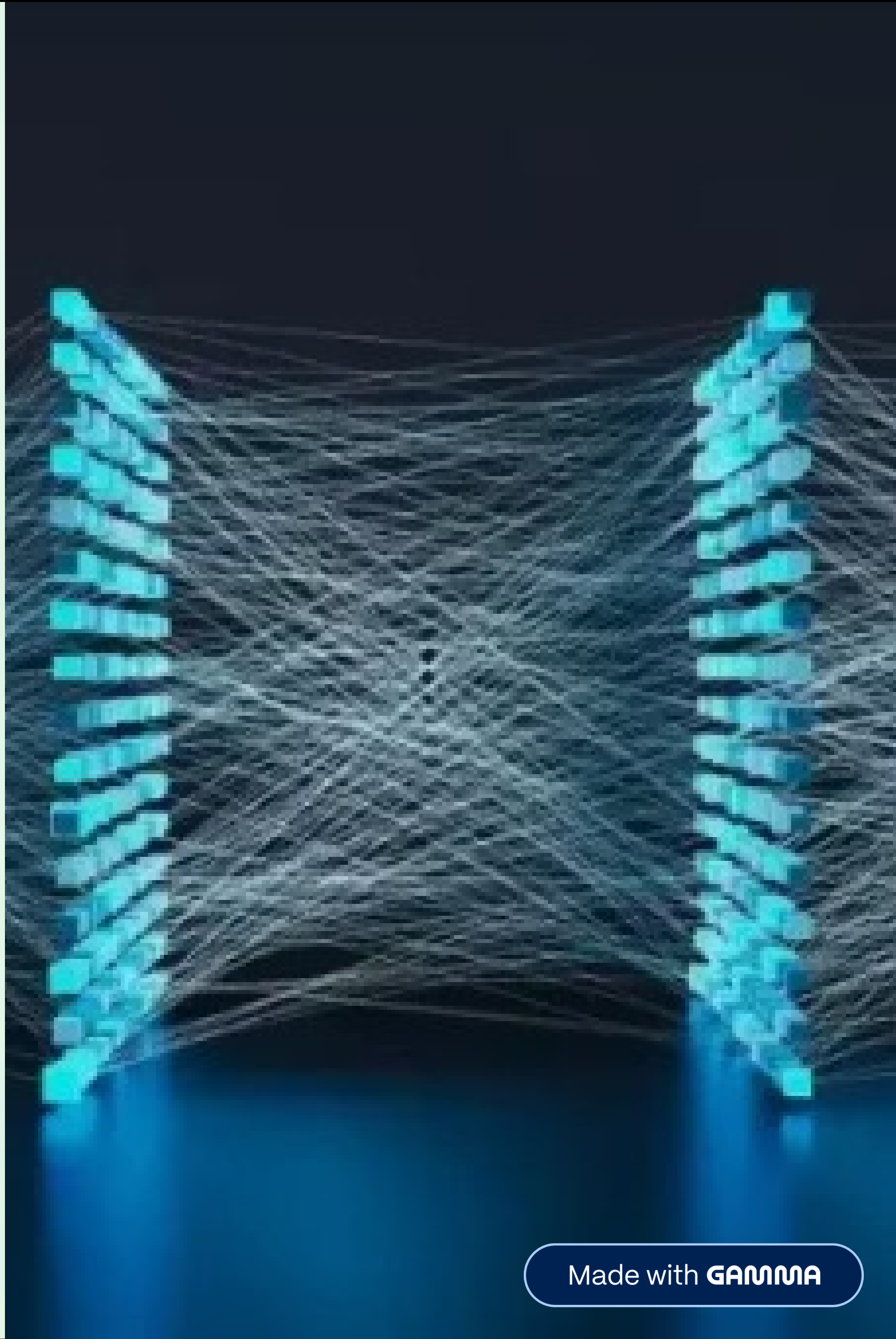
Testing:

After rigorous testing against unseen photos from the web and actual field scenarios, our trained model is now validated and ready.

MODEL TRAINING

Why ResNet18 for Classification?

- **Optimal Balance:** Selected for its ideal blend of performance and computational efficiency, crucial for accurate plant identification across diverse platforms.
- **Transfer Learning:** Leverages pre-trained ImageNet weights for rapid development and enhanced generalization, even with limited domain-specific data.
- **Residual Connections:** Effectively mitigates the vanishing gradient problem, enabling deeper networks and more precise fine-grained classification.
- **Fine-Grained Capabilities:** Facilitates accurate classification, distinguishing subtle leaf variations and adapting to varying environmental conditions.



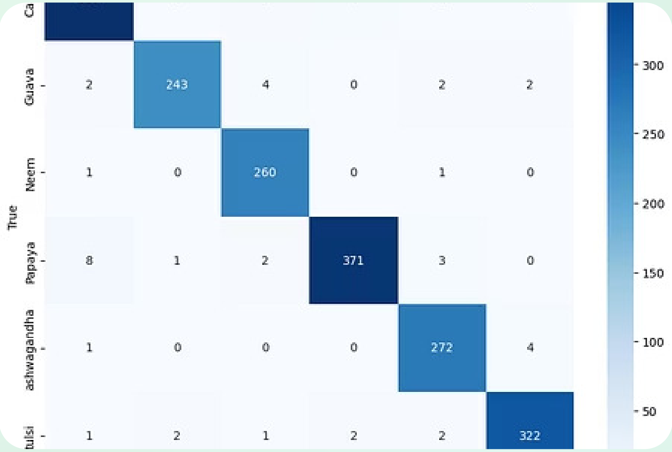
RESULTS

```
Top-5 Accuracy : 100.00%

Weighted Precision : 0.9759
Weighted Recall    : 0.9757
Weighted F1-score  : 0.9757

Detailed Classification Report:
              precision    recall  f1-score   support

      Castor       0.97      0.98      0.97       387
       Guava       0.99      0.96      0.97       253
        Neem       0.97      0.99      0.98       262
      Papaya       0.98      0.96      0.97       385
 ashwagandha       0.97      0.98      0.97       277
       tulsi       0.98      0.98      0.98       330

    accuracy                           0.98      1894
```

Other Parameters

```
Per-Class Accuracy:
Castor               : 98.19%
Guava                : 96.05%
Neem                 : 99.24%
Papaya               : 96.36%
ashwagandha          : 98.19%
tulsi                : 97.58%


Average Confidence: 0.9705
```

Accuracy



Confusion Matrices

MODEL TRAINING

Leaf Disease Detection

Objective: Extend the system beyond plant species recognition to analyze leaf health automatically.

Plant Identification

Recognize plant species from leaf image

Leaf Segmentation

Isolate affected leaf regions using image processing

Disease Classification

Analyze patterns using ResNet 18 model

Confidence Output

Display disease type and severity score

Key Benefits

- Helps farmers detect crop diseases early
- Supports agricultural researchers in plant pathology studies
- Enables users to take preventive action before crop loss
- Provides real-time health monitoring for plants

MODEL TRAINING

Disease Classes Used

Apple — Cedar Apple Rust

Apple — Black Rot

Potato — Healthy

Tomato — Yellow Leaf Curl Virus

Maize — Common Rust

Squash — Powdery Mildew

MODEL TRAINING

MobileNetV2 vs ResNet18 — Technical Comparison for Leaf Disease Detection

A detailed evaluation of deep learning architectures for fine-grained leaf disease classification.

MobileNetV2

- Lightweight architecture designed for mobile inference
- Depthwise separable convolutions reduce computation
- Lower feature extraction depth → struggles with subtle lesions
- Difficult to differentiate fungal vs viral disease patterns
- Lower confidence & unstable predictions across disease classes

ResNet18

- Deeper residual layers enable richer feature extraction
- Strong capability for fine-grained classification
- Better at identifying lesion texture, color, and pattern variations
- Higher per-class confidence and stable predictions
- Suitable for transfer learning with smaller datasets

Decision: MobileNetV2 rejected — Failed to successfully test images from web. ResNet18 selected as the final model for reliable leaf disease detection.

MODEL TRAINING

Why ResNet18 Performs Better

Residual Connections

ResNet18's architectural design, particularly its residual connections, enables the training of much deeper networks without encountering the vanishing gradient problem, facilitating more profound learning.

Fine-Grained Feature Learning

This model excels at identifying subtle micro-lesions and minute discoloration patterns on plant leaves, which are critical for early and accurate disease detection.

Transfer Learning Advantage

By leveraging pre-trained weights from large datasets like ImageNet, ResNet18 significantly boosts performance and generalisation, especially when working with limited domain-specific plant disease data.

Enhanced Per-Class Accuracy

It achieves superior per-class accuracy in disease classification, meaning it can reliably distinguish between various diseases, even those with similar visual symptoms.

Robustness to Variations

ResNet18 demonstrates greater robustness against environmental variables such as varying lighting conditions, shadows, and image noise, ensuring consistent performance in diverse real-world scenarios.

MODEL TRAINING

Key Challenges in Disease Detection

Visual Similarity

Many diseases exhibit similar symptoms, making differentiation challenging even for human experts.

Limited Datasets

Availability of diverse, high-quality real-world diseased plant images is often scarce.

Environmental Variations

Changes in lighting conditions and camera angles can significantly alter image appearance.

Class Imbalance

Some diseases are more prevalent, leading to an unequal distribution of samples in training data.

Overlapping Textures

Complex or overlapping lesion textures can confuse classifiers, hindering accurate identification.

RESULTS & ROADMAP

Results & Future Improvements

Our journey has yielded significant advancements in plant disease detection, and we're continually charting a course for even greater impact.

Key Achievements with ResNet18

Stable Confidence Scores

ResNet18 consistently provides highly reliable and stable confidence scores, ensuring trustworthy disease identification for users.

Enhanced Detection of Rust & Viral Lesions

The model demonstrates superior capabilities in identifying subtle rust and complex viral lesions, crucial for early and effective intervention.

Outperformance in Fine-Grain Detection

Significantly outperforms MobileNetV2 in distinguishing minute, fine-grain disease patterns, leading to more accurate diagnoses.

RESULTS & ROADMAP

Future Research & Development

Expand Real Disease Data Collection

Prioritise gathering more diverse, real-world disease imagery to further bolster model robustness and generalisation.

Support Multi-Disease per Leaf

Develop advanced algorithms to simultaneously identify and classify multiple co-occurring diseases on a single plant leaf.

Integrate Pre-Classification Segmentation

Implement precise image segmentation as a preliminary step to isolate affected regions, enhancing classification accuracy.

Two-Stage Pipeline for Farming Tools

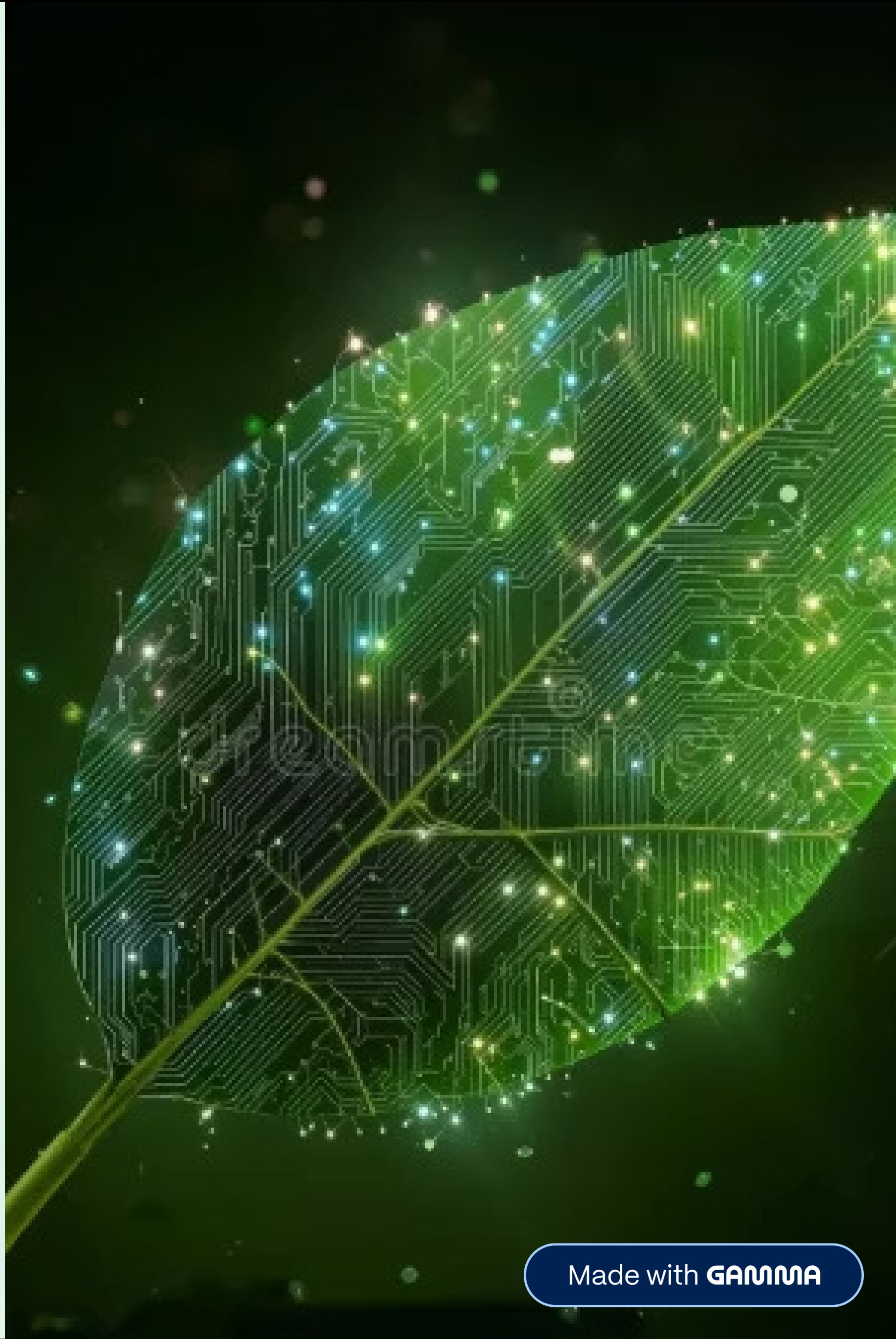
Design a robust two-stage pipeline for seamless integration with automated farming equipment and robotics for proactive crop management.

Enable Mobile/On-Device Inference

Optimise the model for lightweight deployment, allowing for efficient, real-time disease detection directly on mobile devices without constant server communication.

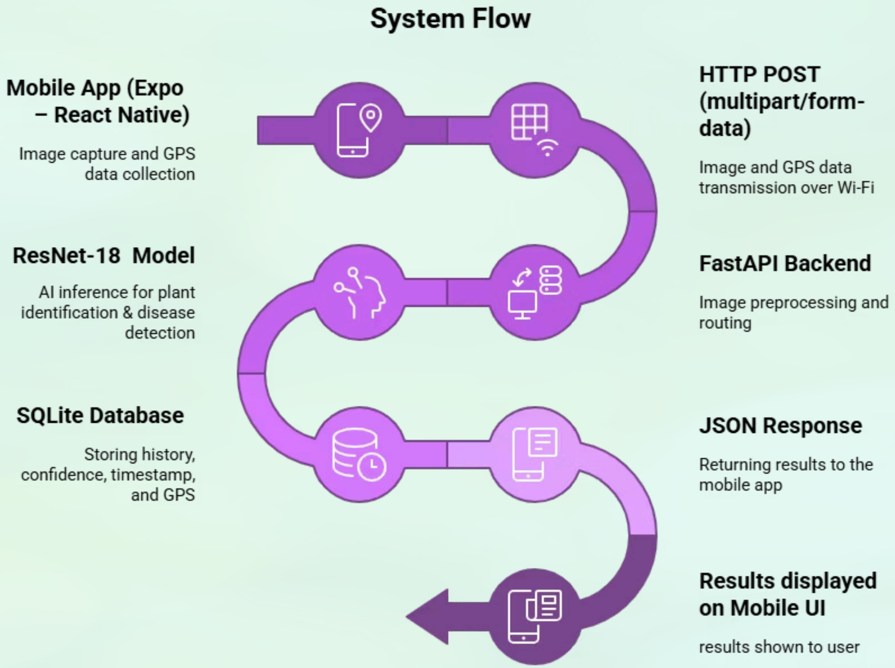
MOBILE APP & BACKEND INTEGRATION FOR PLANT &DISEASE IDENTIFICATION

- Developed a cross-platform mobile application using **React Native (Expo)**.
- Implemented a **FastAPI backend** for AI model inference.
- Integrated **plant identification and plant disease detection models** with the mobile app.
- Enabled **camera capture and gallery image selection** for real-time analysis.
- Implemented **image upload using multipart/form-data** and API communication over local Wi-Fi.
- Added **prediction history storage** using SQLite database.
- Implemented **user feedback system** for prediction validation.
- Integrated **GPS-based location tagging** for each prediction.
- Optimized backend using **image compression** to reduce storage and improve performance.

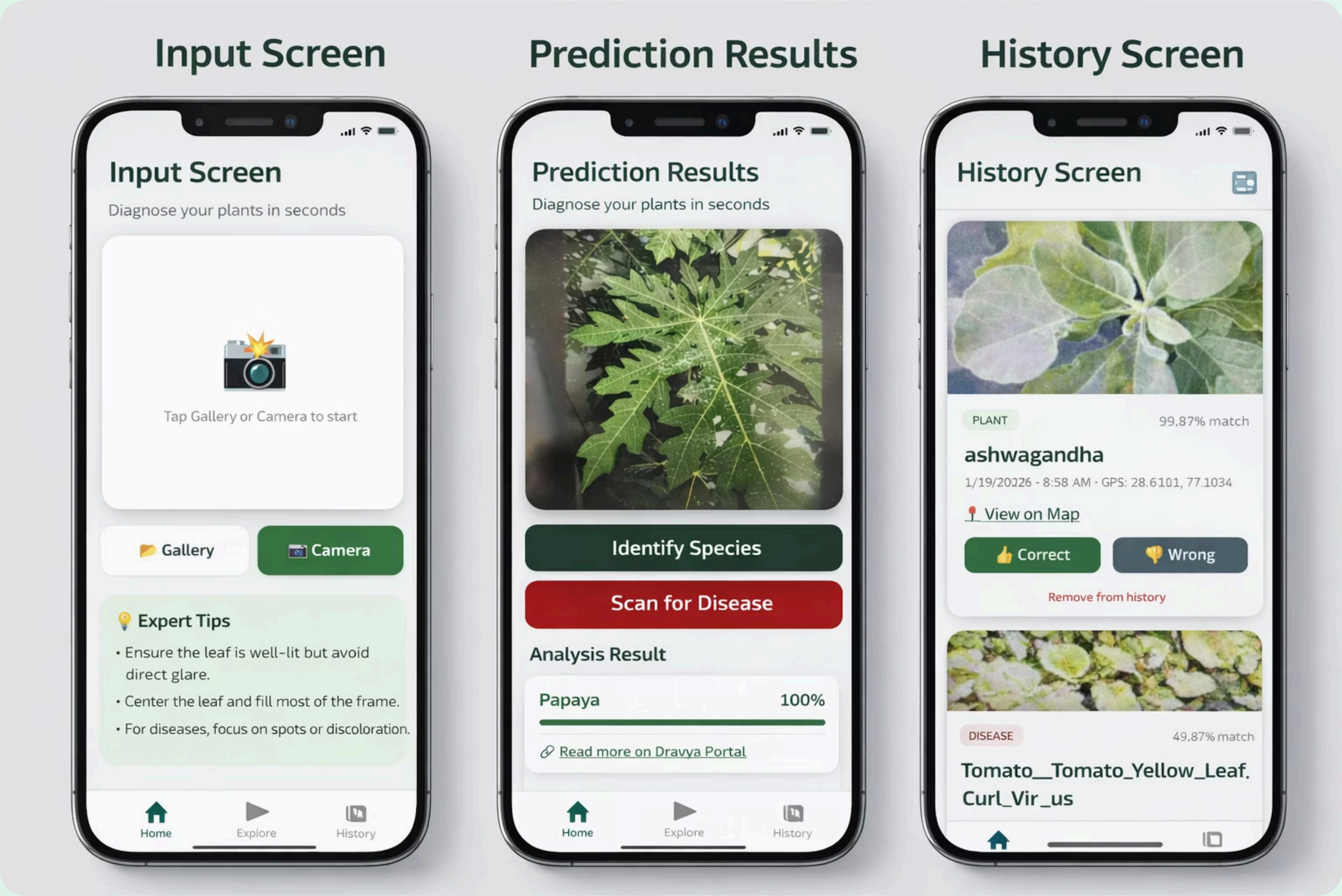


System Flow (How It Works)

- User captures or selects a leaf image using the mobile
- Mobile app sends the image to backend via **HTTP POST request** over local Wi-Fi.
- Top predictions with confidence scores are generated.
- “Read More” option redirects users to **CCRAS–DRAVYA portal** for verified plant information.
- Backend stores prediction details including:
  - Image
  - Prediction type (plant / disease)
  - Confidence
  - Timestamp
  - GPS coordinates
- Mobile app displays prediction results with confidence.
- Prediction history can be viewed, refreshed, or deleted by the user.



Live Demo (Mobile Interface)



CONCLUSION

Future Scope and Impact

- **Expand plant species and disease coverage**
  Extend the system to support a wider variety of plant species and multiple disease classes beyond the current dataset.
- **Improve model accuracy with real-world data**
  Train the AI model on diverse, real-world images captured under varying lighting conditions, angles, backgrounds, and camera qualities to improve robustness and generalization.
- **Cloud deployment for scalability and public access**
  Deploy the FastAPI backend on cloud infrastructure to enable scalable inference, faster response times, and secure public access for multiple users.
- **Offline on-device inference**
  Enable offline predictions by deploying optimized trained models directly on the mobile device using frameworks such as **TensorFlow Lite (TFLite)** or **ONNX**, reducing dependency on network connectivity.
- **Multi-part plant identification**
  Extend identification beyond leaf images by allowing users to capture and analyze **stem, flower, or fruit images**, improving plant recognition accuracy when leaf-based prediction is insufficient.


THANK
YOU!