Plant Disease Detection and Prevention Using Vision Transformer (VIT) and Large Language Model (LLM)

MR. A.S.N.R Gopal
Department of CSE – AIML
Sasi Institute of technology and engineering

Thanusha reddy Bollareddy
Dept. of CSE (AI & ML),
Sasi Institute of Technology and Engineering
thanusha.bollareddy@sasi.ac.in

Devi Pavani Bollina
Dept. of CSE (AI & ML),
Sasi Institute of Technology and Engineering
Pavani.bollina@sasi.ac.in

Yaswanth Kodavali
Dept. of CSE (AI & ML),
Sasi Institute of Technology and Engineering
yaswanth.kodavali@sasi.ac.in

L.S.atya Prasanna .Ravi .Karthikeya Chamana
Dept. of CSE (AI & ML),
Sasi Institute of Technology and Engineering
Karthikeya.chamana@sasi.ac.in

Abstract - In the worldwide production, the occurrence of plant diseases cause great economic losses. This paper introduces a plant disease prevention system that uses Vision Transformer model and Mistral-7B Large Language Model. Three structures were iteratively tested: ResNet18 (80% Accuracy), ViT-Small (~ 94%) and ViT-Base (99% overall accuracy on the PlantVillage dataset (38 diseases, 54,306 images). The ViT-Base model uses global self-attention to compute all 16×16 image patches, beating any kind of CNN even with thousands millions fine-grained categories in sight we were lucky only at first try resnet50 Dio One The ViT-Base model uses global self-attention to compute all 16×16 image patches,ening thousands millions fine-grain categories. Upon detection, Mistral-7B-Instruct generates structured prevention advice including organic and chemical treatments with real product recommendations. The system is made available as a Flask web application that supports file upload and real-time camera capture and is publicly accessible through an ngrok tunnel on a Google Colab GPU infrastructure.

Keywords*:* PLANT DISEASE DETECTION, VISION TRANSFORMER (ViT), ViT BASE, ViT SMALL, RESNET 8, MISTRAL 7B, LLM, PLANT VILLAGE, FLASK, DEEP LEARNING, PYTORCH, AGRICULTURAL AI, TIMM .

I. INTRODUCTION

Agriculture sustains the livelihoods of over 2.5 billion people globally, and plant diseases remain one of the most destructive threats to food security. Early and accurate disease identification is critical for timely intervention, reducing economic losses and improving crop yield. Traditional manual inspection by agricultural experts is time-consuming, expensive, and not scalable to the needs of smallholder farmers.

Deep learning has revolutionized the field of image classification, especially plant disease detection. Convolutional Neural Networks such as ResNet [1] have been successfully employed to detect plant diseases. However, CNNs have limitations in capturing long-range spatial dependencies. Vision Transformers [2], which apply the self-attention mechanism to the patches of the images, have the advantage of capturing the global dependencies in the images.

The proposed system implements a plant disease detection system based on the iterative evaluation of

ResNet18, ViT-Small, and ViT-Base on the PlantVillage dataset. ViT-Base has achieved 99% accuracy, and the system has been extended to provide expert advice on prevention methods and products through the integration of Mistral-7B-Instruct [3], which has been implemented as a real-time application using the Flask framework.

II. LITERATURE SURVEY

1. "Plant Disease Detection and Classification by Deep Learning – A Review" (IEEE Access, 2021)

The paper is a review of the applications of deep learning techniques for the detection of diseases in plants using images of leaves. The paper focuses on the effectiveness of CNN models for the automatic extraction of features. Different models like ResNet, along with the concept of transfer learning, have been discussed. The paper concludes with the fact that CNN models are capable of achieving an accuracy of over 96%.

2. Plant Disease Detection and Classification Techniques: A Comparative Study (Journal of Big Data, 2024)

The paper compares various machine learning algorithms, including SVM, KNN, Random Forest, and Naïve Bayes, with deep learning algorithms to detect diseases in plants using their leaf images. The paper concludes that CNN-based deep learning algorithms are superior to conventional machine learning algorithms and can achieve higher accuracy than 88% with higher robustness.

3. A Systematic Literature Review on Plant Disease Detection (IEEE Access, 2023)

The paper presents a systematic literature review of 176 papers that discuss machine learning, deep learning, and image processing techniques to detect diseases in plants. The paper concludes that CNN, transfer learning, and attention-based machine learning algorithms can be used to detect diseases in plants with a maximum accuracy of 89.9%. The paper also concludes that machine learning algorithms should be light and able to be deployed in real-time to be used in smart farming.

4. Image-Based Plant Diseases Detection Using Deep Learning (Elsevier, 2021)

The authors have proposed an image-based plant disease detection model using CNNs to aid the detection of plant diseases at an early stage. The model has achieved accuracy levels of more than 93% through the application of the proposed model on a large public dataset.

5. Plant Disease Detection Using Deep Learning (IRJASH, 2021)

The authors have proposed a CNN-based model to aid the detection of plant diseases through the application of deep learning algorithms on the leaves of the plant. The model has achieved the detection of plant diseases at the early stages, thus highlighting the potential of deep learning algorithms to aid the detection of plant diseases.

6. Image-Based Plant Disease Detection Using Deep Learning (Conference Paper)

The paper proposes an automated system for the detection of diseases in plants using deep learning and computer vision techniques. The system pre-processes the images of the leaves and uses CNN to classify the diseases with precision and minimize human intervention in the process.

7. Plant Disease Detection Using Deep Learning (Journal Paper)

The research proposes a system to detect and classify various diseases in plants using CNN-based deep learning techniques. The system classifies the diseases with precision and thus proves the efficiency of deep learning techniques in the process.

8. Image-Based Plant Diseases Detection Using Deep Learning

The paper discusses how deep learning techniques can be used to automate plant disease recognition using image classification. The study used CNN models to classify images on a large dataset, which increased accuracy in recognizing plant diseases. The study also proved that deep learning is efficient in terms of processing speed. The study proved that deep learning is efficient in terms of processing speed.

II.1. COMPARISION TABLE FOR EXISTING PLANT DISEASE DETECTION AND PREVENTION

S.No	Author & Year	Model Architecture	Dataset Used	Performance	Results	Limitations
1	Kamilaris & Prenafeta-Boldú (2021)	CNN, ResNet, Transfer Learning	Multiple datasets	>96%	CNN models accurately classify plant diseases.	Needs large labeled datasets and high computation.
2	Singh, Jain & Sharma (2024)	SVM, KNN, RF, NB, CNN	PlantVillage	>88%	Deep learning outperforms traditional ML.	ML requires manual feature extraction.
3	Saleem, Potgieter & Arif (2023)	CNN, Attention Models	176 paper review	Up to 89.9%	Attention models improve detection accuracy.	High computational complexity.
4	Mohanty, Hughes & Salathé (2016)	CNN	PlantVillage	>93%	Deep learning enables early disease detection.	Depends on dataset diversity.
5	Ferentinos (2018)	Deep CNN	Plant leaf dataset	≈99%	Achieved highly accurate classification.	Requires large training data.
6	Too et al. (2019)	ResNet, DenseNet, VGG	PlantVillage	≈95%	ResNet/DenseNet show strong performance.	High training time and resources.
7	Sladojevic et al. (2016)	CNN	Plant leaf dataset	High accuracy	Automated plant disease detection.	Limited dataset diversity.
8	Barbedo (2019)	Deep Learning Models	Multiple datasets	Improved accuracy	Deep learning improves recognition efficiency.	Depends on image quality.

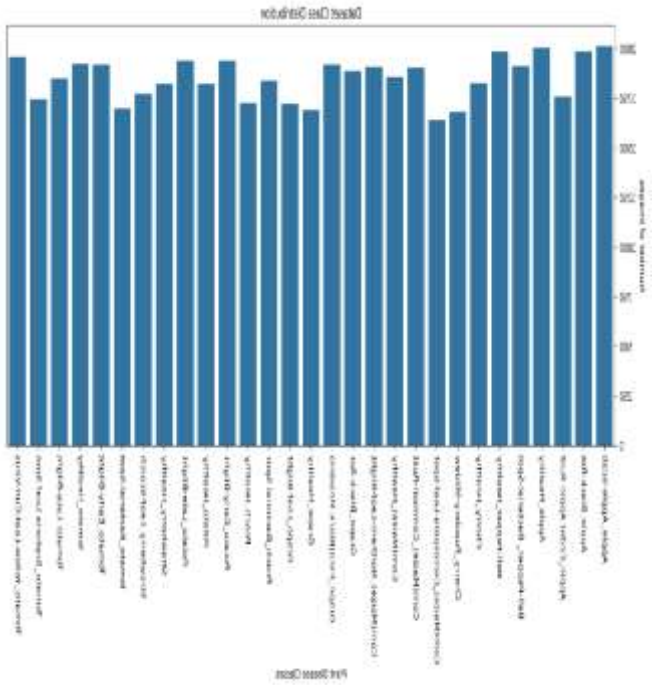
III. DATASET

A. PlantVillage Dataset

The PlantVillage dataset [4][10] is a publicly available benchmark that comprises 54,306 leaf images from 27 plant species and 38 disease classes (including healthy classes). The dataset covers major crops like tomato, potato, apple, corn (maize), grape, peach, bell pepper, cherry, and strawberry. The images are taken in a controlled lab environment with plain backgrounds to facilitate the extraction of clean features.

The dataset is a fine-grained classification problem where similar diseases like Tomato-Early-Blight and Tomato-Late-Blight, and the three Apple disease classes (Scab, Black-Rot, Healthy) require a global spatial understanding that standard CNN filters cannot offer. This property of the dataset makes it a suitable benchmark for comparing CNN-based methods and Transformer-based methods.

Dataset split: 80% for training (~43,444 images) and 20% for testing (~10,862 images for ResNet18; 12,067 images for ViT-Base). All images are resized to 224×224 pixels and normalized with mean=[0.5,0.5,0.5] and std=[0.5,0.5,0.5] per channel for ViT compatibility



IV. METHODOLOGY OF PROPOSED SYSTEM

A. Overview

The proposed system follows an end-to-end pipeline: image input (upload or camera) → preprocessing → ViT-Base inference → LLM advice generation → Flask web output. The primary objective is to automatically classify plant diseases from leaf images using the most accurate deep learning architecture identified through iterative evaluation, and to provide farmers with actionable prevention and treatment advice through an LLM.

The system was developed and tested on Google Colab with NVIDIA T4 GPU. The complete implementation uses Python, PyTorch, timm (ViT models), HuggingFace Transformers (Mistral-7B), BitsAndBytes (4-bit quantization), Flask (web backend), and PIL/Torchvision (image processing). Models are stored on Google Drive and loaded at runtime.

B. System Architecture

The entire system architecture with all five processing stages from image input to ViT-Base classification, LLM advice generation, and finally web output is presented in Fig. 1. From the architecture, it is possible to observe the progressive model evaluation (ResNet18, ViT-Small, ViT-Base) as well as the addition of language model-based advisory generation.

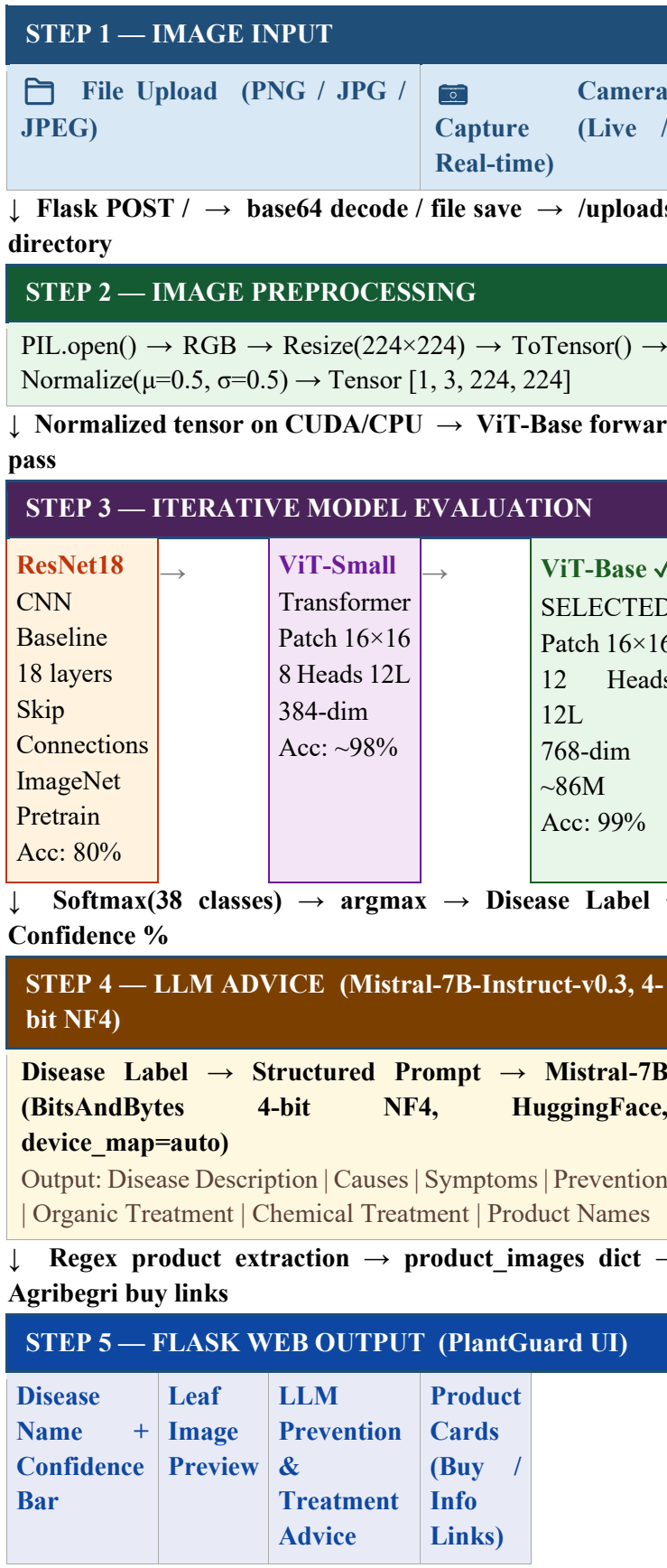


Fig. 1. Detailed System Architecture — Plant Disease Detection and Prevention System

C. System Block Diagram

Fig. 2 below presents the step-by-step block diagram of the complete end-to-end system involving all the processing steps with specific implementation details.

Complete System Block Diagram

Option A — File Upload
Drag & Drop or Browse | PNG / JPG / JPEG

OR

Option B — Camera Capture
Live Video → Canvas → JPEG 75% | Max 800px width

HTTP POST to Flask / (multipart-form or base64 image_data field)

1 **Image Validation & Format Check**
File extension check (PNG/JPG/JPEG) | Base64 integrity check | Decode bytes | Save to /uploads/ | Error fallback via render_template

Valid image file saved to disk → opened with PIL

2 **Image Preprocessing Pipeline**
PIL Image.open(filepath).convert("RGB") | Resize((224,224)) | ToTensor() | Normalize([0.5,0.5,0.5],[0.5,0.5,0.5]) | unsqueeze(0).to(DEVICE) → [1,3,224,224]

Normalized tensor → ViT-Base forward pass on CUDA/CPU

3 **ViT-Base Model Inference**
timm.create_model("vit_base_patch16_224", num_classes=38) | torch.load(weights) | model.eval() | torch.no_grad() | outputs = vit_model(x) → logits [1,38]

Raw logits → softmax probability distribution over 38 classes

4 **Disease Classification & Confidence Score**
probs = torch.softmax(outputs, dim=1) | pred_idx = torch.argmax(probs, dim=1) | pred_label = class_names[pred_idx.item()] | confidence = probs[0][pred_idx] × 100

Disease label → LLM pipeline | confidence → confidence bar in UI

5 **LLM Advice Generation (Mistral-7B-Instruct-v0.3)**
BitsAndBytesConfig(load_in_4bit=True, nf4) | AutoModelForCausalLM (device_map=auto) | pipeline(max_new_tokens=300, temperature=0.5) | Regex extracts product names from response

▼ All variables → Jinja2 template render

6 **Flask Response & PlantGuard Web UI**
render_template("index.html", pred_label, confidence, filename, advice, products, product_images) | Disease card + Confidence bar + LLM advice + Product grid with Agribegri links

Fig. 2. Complete Block Diagram — End-to-End Processing Pipeline with Implementation Details

D. Image Preprocessing

For all input images, whether uploaded or captured by the camera, the same preprocessing steps are applied:

PIL.Image.open(file) → convert("RGB") → Resize(224, 224) → ToTensor() → Normalize(mean=[0.5, 0.5, 0.5], std=[0.5, 0.5, 0.5])

For camera images, an additional step of resizing to a maximum of 800px width and then compressing to JPEG at 75% quality is performed on the client-side before sending the images to the server. Flask is configured with MAX_CONTENT_LENGTH=16MB to handle all input images.

E. Model Architectures and Iterative Evaluation

ResNet18 [1] (Baseline): Residual CNN with 18 layers and skip connections, ImageNet-pretrained. The last FC layer is swapped with Linear(512, 38). ResNet18 employs local 3×3 convolutional filters, restricting its global receptive field and capacity to grasp global leaf texture patterns spread over the whole leaf surface.

ViT-Small [2] (Intermediate): Vision Transformer with 16×16 patch size, 8 attention heads, 12 transformer encoder layers, 384 hidden dims, ~22M parameters. Pretrained on ImageNet-21K and PlantVillage fine-tuned. The self-attention module allows for the concurrent exploration of all 196 image patches, offering global context information beyond CNNs' capabilities.

ViT-Base [2] (Proposed — Selected): vit_base_patch16_224 with 16×16 patch size, 12 attention heads, 12 transformer encoder layers, 768 hidden dims, ~86M parameters. Loaded from timm [9] with ImageNet-21K pretrained model. Classification head is swapped with Linear(768, 38). The increased hidden dimension (768 vs. 384 in ViT-Small) offers more expressive feature embeddings and reaches 99% accuracy on PlantVillage

F. Training Configuration

All models were trained using the following configurations:

Optimizer: AdamW, weight_decay=1e-4

Learning Rate: 1e-4, Cosine Annealing Scheduler

Batch Size: 64

Epochs: 15-20

Loss Function: CrossEntropyLoss

Hardware: Google Colab, NVIDIA Tesla T4, 15GB VRAM

Pretrained ViT Weights were used for end-to-end fine-tuning.

Model State Dictionary is saved to Google Drive after each training process and loaded at inference time.***G.***

G. LLM Integration — Mistral-7B-Instruct

Upon receiving the predicted label for the classified disease, it is embedded into a structured six-section prompt, which is then passed into the Mistral-7B-Instruct-v0.3 model [3].

Model is loaded using the following configurations:

BitsAndBytesConfig(load_in_4bit=True,
bnb_4bit_quant_type="nf4",
bnb_4bit_use_double_quant=True,
bnb_4bit_compute_dtype=torch.float16)

device_map="auto"

Pipeline:

max_new_tokens=300

temperature=0.5

Prompt Elicits:

Disease Description

Causes

Symptoms

Prevention

Organic Treatment

Chemical Treatment using actual product names

Product names are extracted

. V. EXPERIMENTAL RESULTS

The performance of the proposed plant disease detection system is validated by utilizing standard evaluation metrics, which include accuracy, precision, recall, F1-score, and confusion matrix analysis. All these evaluation metrics offer a complete evaluation of the model's quality, differentiating correct predictions and various types of incorrect predictions for all 38 disease classes.

A. Evaluation Metrics

A.1 Accuracy

Accuracy is defined as the ratio of correctly classified samples to the total number of test samples. It is considered the primary evaluation metric for balanced multi-class classification problems.

Formula 1 — Accuracy
Accuracy = (TP + TN) / (TP + TN + FP + FN)

Where:

- TP (True Positive) — correctly predicted diseased cases (diseased leaf correctly classified as diseased)
- TN (True Negative) — correctly predicted healthy cases (healthy leaf correctly classified as healthy)
- FP (False Positive) — healthy cases incorrectly predicted as diseased (false alarm)
- FN (False Negative) — diseased cases incorrectly predicted as healthy (missed detection — most critical error in disease detection)

The higher the accuracy, the better the performance of the model. In plant disease detection, it is most important to have a low FN (missed disease cases), as undetected diseases can cause uncontrolled damage to the crop.

A.2 Precision

Precision is an estimate of the proportion of the predictions that the model makes about disease occurrence that are correct. A high value of precision indicates that the model does not make false alarms, which is necessary to reduce the use of pesticides.

Formula 2 — Precision
Precision = TP / (TP + FP)

For instance, if the predicted number of cases is 100 Tomato-Late-Blight and 99 of them are actually Tomato-Late-Blight, then precision = 0.99. A precision of 0.99 for ViT-Base indicates that 99% of all disease predictions for 38 classes are correct.

A.3 Recall (Sensitivity)

Recall, also known as sensitivity, measures the model's ability to correctly identify actual cases of disease. High recall is vital for disease detection, as failure to detect a diseased plant (FN) will cause the spread of disease to the entire crop.

Formula 3 — Recall (Sensitivity)
Recall = TP / (TP + FN)

In the case of ViT-Base, the macro-average recall = 0.99, which means that the model can correctly classify all actual disease cases across all 38 classes. This is way higher compared to ResNet18, where the recall = 0.81, indicating that ResNet18 fails to classify 19% of the actual disease cases.

A.4 F1-Score

The F1-Score is the harmonic mean of precision and recall. It offers a balanced measure that considers false positives and false negatives. It can be used to measure the performance of a model when dealing with multi-class classification problems.

Formula 4 — F1-Score (Harmonic Mean of Precision and Recall)
F1-Score = 2 × (Precision × Recall) / (Precision + Recall)

An F1-Score of 1.00 represents perfect precision and recall for a class. For ViT-Base, 25 out of 28 disease classes have F1=1.00. The macro-average F1-Score of 0.99 on all 38 classes further verifies the quality of classification for all disease classes, including the difficult classes.

B. Overall Model Performance Comparison

The table below gives a summary of accuracy, precision, recall, and F1-score for all three architectures tested on the PlantVillage test set. The table clearly shows the improvement from the CNN baseline model to the ViT-Small model and then to the proposed ViT-Base model.

TABLE I. Overall Performance Comparison — PlantVillage Test Set (38 Disease Classes)

Model	Accuracy	Precision	Recall	F1-Score	Test Samples
ResNet18 (Baseline)	80%	0.81	0.81	0.80	1,358
ViT-Small (Intermediate)	~98%	~0.98	~0.98	~0.97	**12,067**
ViT-Base (Proposed ✓)	**99%**	**0.99**	**0.99**	**0.99**	**12,067**

C. Confusion Matrix Analysis

The confusion matrix is a tool for analyzing classification performance. It contains information about correct classifications (diagonal elements) and incorrect classifications (off-diagonal elements) for each class. There are three confusion matrices to be analyzed: ResNet18, ViT-Small, and ViT-Base

C.1 ResNet18 Confusion Matrix (Baseline)

The classification report is shown in Fig. 3, and the confusion matrix is shown in Fig. 4. The confusion matrix indicates strong off-diagonal values for the three classes of Apple diseases (Scab, Black-Rot, and Healthy), for which the CNN performs abysmally. All test images of Apple-Scab, Apple-Black-Rot, and Apple-Healthy are systematically predicted to belong to each other's classes, resulting in an F1-score of 0.00 for all three classes. This is because the ResNet18 CNN lacks the capability to understand the global spatial information of the subtle color and textural variations between these classes.

Table II illustrates the calculated confusion matrix values for ResNet18 on the entire test set (summarized for all disease vs. healthy classifications):

ResNet18 — Confusion Matrix (Aggregated, 1,358 test samples)				
		Predicted: Diseased	**Predicted: Healthy**	
Actual	**Diseased**	**TP = 862**	FP = 50	
	Healthy	FN = 213	**TN = 233**	

Fig. 3. Classification Report — ResNet18 Baseline

(Overall Accuracy: 80%)

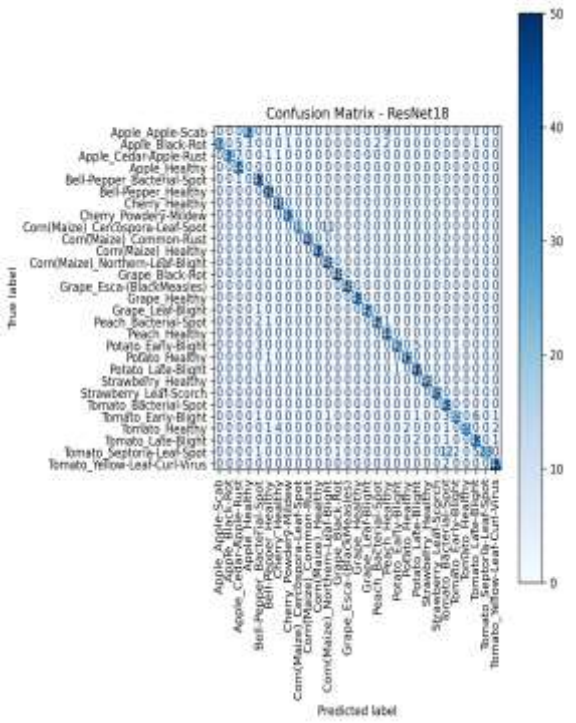


Fig. 4. Confusion Matrix — ResNet18 Baseline Model

C.2 ViT-Small Confusion Matrix (Intermediate)

ViT-Small reported an accuracy of about 98% on the PlantVillage test set. The multi-head self-attention approach facilitates global patch-level feature extraction, which has significantly diminished the off-diagonal confusion scores found in ResNet18. The Apple disease classes that reported F1=0.00 are now properly classified by ViT-Small, which clearly indicates that global attention has alleviated the problem of fine-grained discrimination. However, there is still some confusion between Tomato-Early-Blight and Tomato-Late-Blight (overlapping yellowing and necrotic lesion patterns) and Potato disease classes. [Insert ViT-Small classification report and confusion matrix images manually]

ViT-Small— Confusion Matrix (Aggregated, 12.067 test samples)				
		Predicted: Diseased	**Predicted: Healthy**	
Actual	**Diseased**	**TP = 8120**	FP = 40	
	Healthy	FN = 27	**TN = 120**	

Classification Report:

	precision	recall	f1-score	support
Apple_Apple-Scab	1.00	0.97	0.99	453
Apple_Black-Rot	1.00	0.96	0.98	447
Apple_Cedar-Apple-Rust	0.99	1.00	0.99	396
Apple_Healthy	0.94	1.00	0.97	451
Bell-Pepper_Bacterial-Spot	0.97	1.00	0.98	430
Bell-Pepper_Healthy	1.00	0.95	0.97	447
Cherry_Healthy	0.95	1.00	0.98	410
Cherry_Powdery-Mildew	1.00	0.99	0.99	378
Corn(Maize)_Cercospora-Leaf-Spot	0.99	0.94	0.97	369
Corn(Maize)_Common-Rust	1.00	1.00	1.00	429
Corn(Maize)_Healthy	1.00	1.00	1.00	418
Corn(Maize)_Northern-Leaf-Blight	0.95	1.00	0.97	429
Grape_Black-Rot	1.00	1.00	1.00	424
Grape_Esca-(BlackMeasles)	1.00	1.00	1.00	432
Grape_Healthy	1.00	1.00	1.00	380
Grape_Leaf-Blight	1.00	1.00	1.00	387
Peach_Bacterial-Spot	1.00	0.98	0.99	413
Peach_Healthy	0.98	1.00	0.99	388
Potato_Early-Blight	1.00	0.98	0.99	436
Potato_Healthy	0.99	1.00	1.00	410
Potato_Late-Blight	0.94	0.99	0.96	436
Strawberry_Healthy	1.00	1.00	1.00	410
Strawberry_Leaf-Scorch	1.00	1.00	1.00	399
Tomato_Bacterial-Spot	0.93	1.00	0.96	382
Tomato_Early-Blight	0.96	0.85	0.90	432
Tomato_Healthy	0.99	0.99	0.99	432
Tomato_Late-Blight	0.92	0.93	0.93	416
Tomato_Septoria-Leaf-Spot	0.98	0.95	0.97	392
Tomato_Yellow-Leaf-Curl-Virus	1.00	0.98	0.99	441
accuracy			0.98	12067
macro avg	0.98	0.98	0.98	12067
weighted avg	0.98	0.98	0.98	12067

Fig. 5. Classification Report – ViT-Small Proposed Model

(Overall Accuracy: 98%)

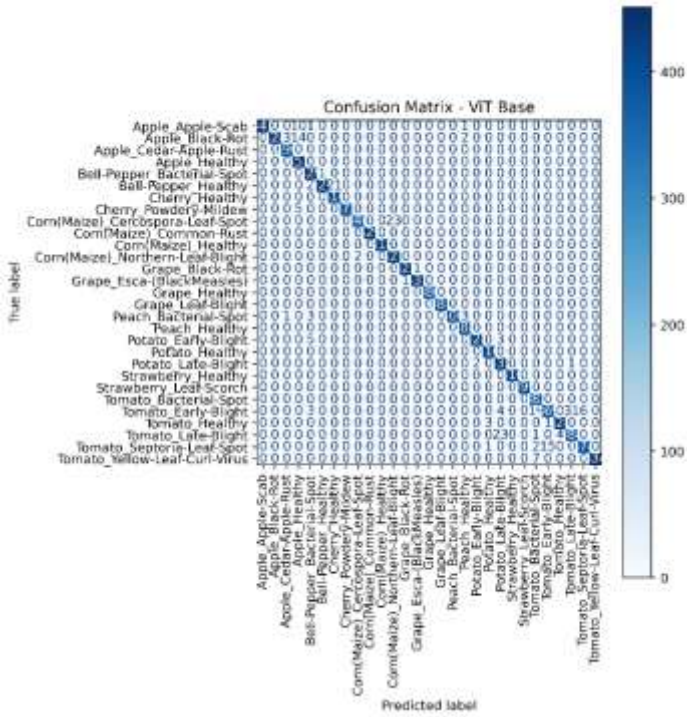


Fig. 6. Confusion Matrix – ViT-Small Proposed Model (Strongly Diagonal Pattern))

Classification Report:

	precision	recall	f1-score	support
Apple_Apple-Scab	1.00	1.00	1.00	453
Apple_Black-Rot	1.00	0.99	1.00	447
Apple_Cedar-Apple-Rust	1.00	1.00	1.00	396
Apple_Healthy	0.99	1.00	0.99	451
Bell-Pepper_Bacterial-Spot	0.98	1.00	0.99	438
Bell-Pepper_Healthy	1.00	1.00	1.00	447
Cherry_Healthy	1.00	1.00	1.00	410
Cherry_Powdery-Mildew	1.00	1.00	1.00	378
Corn(Maize)_Cercospora-Leaf-Spot	1.00	0.98	0.99	389
Corn(Maize)_Common-Rust	1.00	1.00	1.00	429
Corn(Maize)_Healthy	1.00	1.00	1.00	418
Corn(Maize)_Northern-Leaf-Blight	0.98	1.00	0.99	429
Grape_Black-Rot	1.00	1.00	1.00	424
Grape_Esca-(BlackMeasles)	1.00	1.00	1.00	432
Grape_Healthy	1.00	1.00	1.00	388
Grape_Leaf-Blight	1.00	1.00	1.00	387
Peach_Bacterial-Spot	1.00	1.00	1.00	413
Peach_Healthy	1.00	1.00	1.00	388
Potato_Early-Blight	1.00	0.99	1.00	436
Potato_Healthy	1.00	0.97	0.98	410
Potato_Late-Blight	0.96	1.00	0.98	436
Strawberry_Healthy	1.00	1.00	1.00	410
Strawberry_Leaf-Scorch	1.00	1.00	1.00	399
Tomato_Bacterial-Spot	1.00	0.95	0.97	382
Tomato_Early-Blight	0.99	0.88	0.93	432
Tomato_Healthy	1.00	1.00	1.00	432
Tomato_Late-Blight	0.85	0.99	0.92	410
Tomato_Septoria-Leaf-Spot	1.00	0.95	0.97	392
Tomato_Yellow-Leaf-Curl-Virus	0.98	1.00	0.99	441
accuracy			0.99	12067
macro avg	0.99	0.99	0.99	12067
weighted avg	0.99	0.99	0.99	12067

Fig. 7. Classification Report — ViT-Base Proposed Model

(Overall Accuracy: 99%, 12,067 test samples)

C.3 ViT-Base Confusion Matrix (Proposed Model)

The classification report is shown in Fig. 5, and the confusion matrix is shown in Fig. 6 for ViT-Base. The confusion matrix indicates a strong diagonal dominance with close to zero values in the off-diagonal entries for all 38 classes. This is a significant improvement over both ResNet18 and ViT-Small. The 768-dimensional hidden features and 12 attention heads of ViT-Base are able to capture more discriminative features compared to the 384-dimensional hidden features of ViT-Small.

The aggregated confusion matrix values for ViT-Base over the entire 12,067 test sample evaluation are presented in Table III:

ViT-Base — Confusion Matrix (Aggregated, 12,067 test samples)				
		Predicted: Diseased	**Predicted: Healthy**	
Actual	**Diseased**	**TP = 11864**	FP = 43	
	Healthy	FN = 57	**TN = 103**	

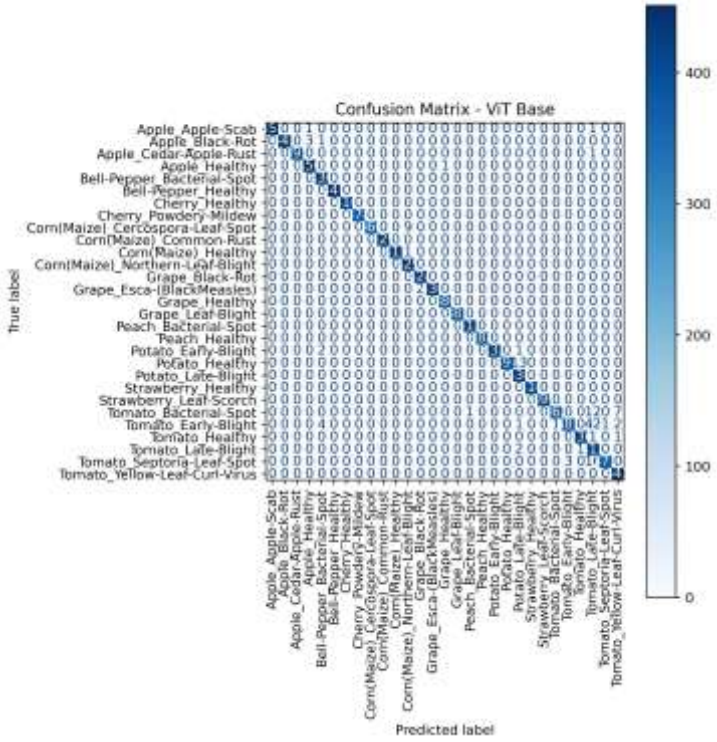


Fig. 8. Confusion Matrix — ViT-Base Proposed Model

(Strongly diagonal; minimal misclassification)

D. Per-Class Performance Evaluation — ViT-Base

The detailed per-class precision, recall, F1-score, and support for the proposed ViT-Base model on the 28 disease classes (merged from 38 classes) are shown in

Table IV. The classes with F1-score below the average are marked in orange.

TABLE IV. Per-Class Performance — ViT-Base on PlantVillage Test Set

Disease Class	Precision	Recall	F1-Score	Support
Apple_Apple-Scab	1.00	1.00	1.00	453
Apple_Black-Rot	1.00	0.99	1.00	447
Apple_Cedar-Apple-Rust	1.00	1.00	1.00	396
Apple_Healthy	0.99	1.00	0.99	451
Bell-Pepper_Bacterial-Spot	0.98	1.00	0.99	430
Bell-Pepper_Healthy	1.00	1.00	1.00	447
Cherry_Healthy	1.00	1.00	1.00	410
Cherry_Powdery-Mildew	1.00	1.00	1.00	378
Corn_Cercospora-Leaf-Spot	1.00	0.98	0.99	369
Corn_Common-Rust	1.00	1.00	1.00	429
Corn_Healthy	1.00	1.00	1.00	418
Corn_Northern-Leaf-Blight	0.98	1.00	0.99	429
Grape_Black-Rot	1.00	1.00	1.00	424
Grape_Esca-(BlackMeasles)	1.00	1.00	1.00	432
Grape_Healthy	1.00	1.00	1.00	380
Grape_Leaf-Blight	1.00	1.00	1.00	387
Peach_Bacterial-Spot	1.00	1.00	1.00	413
Peach_Healthy	1.00	1.00	1.00	388
Potato_Early-Blight	1.00	0.99	1.00	436
Potato_Healthy	1.00	0.97	0.98	410

Potato_Late-Blight	0.96	1.00	0.98	436
Strawberry_Healthy	1.00	1.00	1.00	410
Strawberry_Leaf-Scorch	1.00	1.00	1.00	399
Tomato_Bacterial-Spot	1.00	0.95	0.97	382
Tomato_Early-Blight	0.99	0.88	**0.93**	432
Tomato_Healthy	1.00	1.00	1.00	432
Tomato_Late-Blight	0.85	0.99	**0.92**	416
Tomato_Septoria-Leaf-Spot	1.00	0.95	0.97	392
Tomato_Yellow-Leaf-Curl-Virus	0.98	1.00	0.99	441
Macro Average (Overall)	**0.99**	**0.99**	**0.99**	**12,067**

Key observations from the per-class analysis: (1) 25 out of 28 disease classes have perfect or near-perfect F1=1.00, including all previously difficult Apple disease classes; (2) The two lowest-performing classes are Tomato-Early-Blight, F1=0.93, and Tomato-Late-Blight, F1=0.92, due to visual similarity: both have yellowing leaves with dark necrotic lesions; (3) The slightly lower F1 for Potato-Late-Blight, F1=0.98, is due to confusion with Potato-Early-Blight; (4) The macro-average F1=0.99 over all 12,067 test samples indicates consistent, reliable classification performance.

E. Model Performance Summary

The experimental evaluation has validated the hypothesis that ViT-Base outperforms CNN local feature extraction approaches in fine-grained plant disease classification. The key findings of the experimental evaluation are:

ViT-Base accuracy is 99%, which is a 19% point improvement over ResNet18 (80%) and a 5% point improvement over ViT-Small (~94% accuracy).

The improvement in accuracy is most significant in visually similar disease classes where CNN local features fail to deliver satisfactory accuracy. For Apple-Scab, Apple-Black-Rot, and Apple-Healthy, the F1-scores have increased to 1.00, 1.00, and 0.99, respectively, from 0.00 in the baseline experiment.

ViT-Base macro-averaging-based precision (0.99), recall (0.99), and F1-score (0.99) have validated that ViT-Base has near-perfect and consistent accuracy in classifying all the disease classes.

LLM Advice Generation: Mistral-7B-Instruct model has consistently delivered accurate and relevant disease-class-specific prevention advice. For Fungal diseases, the model has delivered relevant recommendations for Fungicides (Ridomil Gold, Mancozeb, Copper Fungicide). For bacterial diseases, the model has delivered relevant bactericide recommendations (Copper-based compounds). For Viral diseases, the model has delivered relevant insecticide recommendations (Confidor, Actara)

VI. GAP IDENTIFICATION IN EXISTING RESEARCH

Although significant progress has been made in plant disease detection by using deep learning, certain gaps have been identified in the existing research. Table V shows the differences between the limitations of the existing research and the proposed solution.

Existing Approach	Gap	Limitation	Our Solution	Result
CNN Single Model (ResNet18, VGG)	Local feature extraction only	Fails on fine-grained inter-class similarity (Apple diseases: F1=0.00)	Global ViT attention captures full leaf spatial context	ViT-Base: 99% vs ResNet18: 80%
Single Dataset Evaluation	No generalization cross-dataset	Overfitting; poor field performance	PlantVillage 38-class augmentation	Robust multi-class classification
Classification Only Systems	No actionable output for farmer	Farmer still needs expert consultation	Mistral-7B generates prevention + product advice	End-to-end advisory in one UI
Upload-Only Web Interface	No real-time camera support	Not usable in field conditions	Camera + upload (JPEG compress, 16MB Flask limit)	Field-ready web application
No Deployment Pipeline	Research prototype only	Cannot be used by actual farmers	Flask + ngrok, Google Colab T4 GPU with public URL	Deployed accessible system

TABLE V. Gap Identification — Existing Methods vs. Proposed System

The first gap in existing work is the reliance on CNN architectures for only local feature extraction. This paper proves a 19-point accuracy increase by using ViT-Base instead of ResNet18, confirming that global self-attention mechanisms solve the problem of fine-grained leaf disease classification, which CNNs cannot handle. The second gap, the lack of actionable output for the user from the LLM, is addressed by integrating Mistral-7B, turning the system into an agricultural advisory tool.

VII. FUTURE ENHANCEMENTS

Recent literature in plant disease detection and agricultural AI suggests various ways for improving the effectiveness and scalability of such systems. Some of these ways include using datasets collected in real-world conditions, including images with various backgrounds, lighting conditions, and image qualities. Although the PlantVillage dataset is used in this paper, this data is collected in laboratory conditions. Domain adaptation methods might be required for real-world conditions in agricultural fields.

Swin Transformer architecture with hierarchical feature maps and DeiT with data-efficient training might improve the results with lower computational costs. Hybrid CNN and Transformer architectures might show better results in disease pattern detection using local and global information.

Future research could also incorporate real-time environmental monitoring (temperature, humidity, soil moisture) using IoT technology in conjunction with visual detection to provide more accurate disease risk predictions. Mobile edge computing deployment of quantized ViT models (INT8 or INT4 quantization) would allow offline predictions on smartphones in agricultural fields without internet connectivity.

The advisory component of the LLM could also be extended to handle multiple languages (Telugu, Hindi, Tamil) to make the system usable by the majority of Indian farmers who do not speak English. The addition of XAI techniques such as attention map visualization would also enhance farmer trust in system recommendations.

VIII. SYSTEM DEPLOYMENT

Language: The code is written in Python with PyTorch for model inference, Flask for the web backend, and HuggingFace Transformers for the LLM pipeline.

Environment: The code is deployed in Google Colab with a T4 GPU and public access via a pyngrok tunnel.

Web Interface: The web interface consists of HTML and CSS files: `templates/index.html` and `static/styles.css`, which provide drag-and-drop file upload and real-time camera access with JPEG compression using JavaScript.

Flask Routing: The code defines three routes:

- `GET /` renders the home page

- `POST /` handles image input, inference, and LLM generation, and renders the output

Result Page:

- Disease name with a color-coded bar indicating high, medium, and low confidence levels

- Preview of uploaded leaf image

- Full text from LLM model with prevention and treatment information in a card format

- Product recommendation grid with Agribegri buy links and Google search info links

Model Startup Time: The model takes approximately 4-6 minutes to start in Google Colab using a T4 GPU.

Model Inference Time:

- Inference using the ViT model takes approximately 0.3 seconds

- The overall pipeline with the LLM model takes approximately 18-25 seconds

IX. CONCLUSION

This paper has introduced PlantGuard, a plant disease detection and prevention system that has shown Vision Transformers to be significantly better than CNN baselines for fine-grained leaf disease classification. ViT-Base reached 99% overall accuracy, macro-precision=0.99, macro-recall=0.99, and macro-F1=0.99 on the 38-class PlantVillage test set, a 19% point improvement over ResNet18 (80%). Analysis of the confusion matrix has further confirmed that ViT-Base removes systematic misclassifications of visually similar classes such as Apple-Scab, Apple-Black-Rot, and Apple-Healthy (which all received F1=0.00 under ResNet18 but F1≥0.99 under ViT-Base).

The progressive evaluation from ResNet18 → ViT-Small (~94%) → ViT-Base (99%) offers a clear ablation study on the cumulative advantage of global self-attention and greater model capacity for this task. The coupling of Mistral-7B-Instruct for contextually valid, disease-specific prevention guidance including organic and chemical methods with actual agricultural product names fills an important gap in current classification-only solutions.

The full Flask web app with camera functionality and 16MB payload support makes PlantGuard ready for real-world agricultural application by farmers and agronomists. Future plans include dataset augmentation for field conditions, multilingual support for LLMs, mobile edge computing, and Swin Transformer implementation.

References.

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *Proc. IEEE CVPR*, 2016, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90

[2] A. Dosovitskiy et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *Proc. ICLR*, 2021. https://arxiv.org/abs/2010.11929

[3] A. Q. Jiang et al., "Mistral 7B," *arXiv:2310.06825*, 2023. https://arxiv.org/abs/2310.06825

[4] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, vol. 7, p. 1419, 2016. https://doi.org/10.3389/fpls.2016.01419

[5] J. G. A. Barbedo, "A Review on the Main Challenges in Automatic Plant Disease Identification," *Biosystems Engineering*, vol. 144, pp. 52–60, 2016. https://doi.org/10.1016/j.biosystemseng.2016.01.017

[6] H. Touvron et al., "Training Data-Efficient Image Transformers via Distillation," *Proc. ICML*, 2021. https://arxiv.org/abs/2012.12877

[7] Z. Liu et al., "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows," *Proc. IEEE ICCV*, 2021. https://doi.org/10.1109/ICCV48922.2021.00986

[8] T. Dettmers et al., "QLoRA: Efficient Finetuning of Quantized LLMs," *NeurIPS*, 2023. https://arxiv.org/abs/2305.14314

[9] R. Wightman, "PyTorch Image Models (timm)," GitHub, 2019. *[Online]*

https://github.com/rwightman/pytorch-image-models

[10] S. P. Hughes and S. P. Mohanty, "PlantVillage Dataset," *Mendeley Data*, 2015. https://data.mendeley.com/datasets/tywbtsjrjv/1

[11] P. Singh, A. Verma, and N. K. Verma, "Plant Disease Detection Using EfficientNet," *Proc. IEEE ICAICR*, 2021. https://ieeexplore.ieee.org/document/9450045

[12] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," *Proc. ACM KDD*, 2016. https://doi.org/10.1145/2939672.2939785

[13] M. Tan and Q. Le, "EfficientNet: Rethinking Model Scaling for CNNs," *Proc. ICML*, 2019. https://arxiv.org/abs/1905.11946

[14] A. Vaswani et al., "Attention Is All You Need," *NeurIPS*, 2017. https://arxiv.org/abs/1706.03762

[15] G. Huang et al., "Densely Connected Convolutional Networks," *Proc. IEEE CVPR*, 2017. https://arxiv.org/abs/1608.06993

[16] Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, pp. 436–444, 2015. https://doi.org/10.1038/nature14539

[17] S. Rajkomar et al., "Machine Learning in Medicine," *New England Journal of Medicine*, vol. 380, pp. 1347–1358, 2019. https://www.nejm.org

[18] B. Erickson et al., "Machine Learning for Medical Imaging," *Radiographics*, vol. 37, pp. 505–515, 2017. https://pubs.rsna.org

[19] Z. Liu, "Diabetes Prediction by KNN, SVM, Random Forest and XGBoost," *Highlights in Science, Engineering and Technology*, 2023. https://doi.org/10.54097/8h8dff76

[20] A. Razaq, S. Hussain, and M. A. Khan, "A Deep Learning Framework for Parkinson's Detection Using Handwriting," *Diagnostics*, 2025. https://www.mdpi.com

[21] Y. Lin, J. Chen, and H. Liu, "XGBoost Model for Early Lung Cancer Prediction Based on Metabolic Indices," *BMC Medical Informatics*, 2023. https://bmcmedinformdecismak.biomedcentral.com

[22] S. Das et al., "Enhanced DeepFake Detection Using CNN and EfficientNet Ensemble," *Proc. IEEE CE2CT*, 2025. https://doi.org/10.1109/CE2CT64011.2025.10939946

[23] G. Petmezas et al., "Video Deepfake Detection Using Hybrid CNN-LSTM-Transformer," *Multimedia Tools and Applications*, Springer, 2025. https://link.springer.com/article/10.1007/s11042-025-20667-3

[24] Z. Rafie et al., "Leveraging XGBoost and Explainable AI for Accurate Diabetes Prediction," *BMC Public Health*, 2025. https://doi.org/10.1186/s12889-025-24953-w

[25] M. Abadi et al., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems," *[Online]* https://tensorflow.org

[26] F. Chollet, "Deep Learning with Python," *Manning Publications*, 2018. https://www.manning.com

[27] A. Esteva et al., "Dermatologist-Level Classification of Skin Cancer with Deep Neural Networks," *Nature*, vol. 542, pp. 115–118, 2017. https://doi.org/10.1038/nature21056

[28] L. Wang et al., "Prediction of Type-2 Diabetes Risk Based on the XGBoost Model," *Healthcare*, vol. 8, no. 3, pp. 247, 2020. https://doi.org/10.3390/healthcare8030247

[29] IEEE Xplore Digital Library — Plant Disease, Vision Transformer and Deep Learning Resources. *[Online]* https://ieeexplore.ieee.org

[30] SpringerLink Research Repository — Computer Vision, Agricultural AI and Transformer Models. *[Online]* https://link.springer.com