OPEN ACCESS INTERNATIONAL JOURNAL OF SCIENCE & ENGINEERING

Real-Time Disease Detection in Lemon Leaves

Anusha B M [1], Harshitha C[1], Sai Akshaya A[1], Dr. Harish S.[2]
[1]UG scholar, Dept. of Electronics and Communication Engineering, R L Jalappa institute of Technology, Doddaballapur, Affiliated to VTU Belagavi. India
[a]anushagowdaanu38@gmail.com , [b]harshithachandrashekar511@gmail.com
[c]saiakshaya789@gmail.com
[2]Associate Professor, Department of Electronics and Communication Engineering
R L Jalappa Institute of Technology, Doddaballapur, Affiliated to VTU Belagavi, India
harishsrinivasaiah@gmail.com

Abstract: Lemon cultivation is an important component of horticultural agriculture, contributing significantly to both domestic consumption and commercial markets. However, lemon plants are highly vulnerable to a variety of leaf diseases such as citrus canker, anthracnose, leaf curl virus, spider mite infestation, and other fungal and bacterial infections. These diseases adversely affect leaf health, fruit quality, and overall yield, leading to substantial economic losses for farmers. Early and accurate detection of such diseases is critical for effective disease management and for minimizing the excessive use of pesticides. Traditional disease identification methods rely on visual inspection by experienced farmers or agricultural experts, which is time-consuming, subjective, and often impractical in large-scale or rural farming environments where expert access is limited. To address these challenges, this paper proposes a real-time lemon leaf disease detection system using digital image processing and artificial intelligence techniques. The system is designed to automatically capture, process, and analyze leaf images under real-field conditions. Live images of lemon leaves are acquired using a Raspberry Pi camera module, enabling continuous monitoring without manual intervention. The captured images are pre-processed using Open-CV and NumPy-based techniques such as resizing, noise removal, background elimination, color normalization, and contrast enhancement to improve image quality and ensure robustness against varying lighting and environmental conditions.

Following pre-processing, the system performs feature extraction to identify key visual indicators of disease, including color variation, texture irregularities, lesion patterns, and shape deformation. These features are then analyzed using a trained convolutional neural network (CNN) model optimized for real-time performance. Lightweight deep learning architectures and Tensor-Flow Lite are employed to ensure efficient execution on low-cost hardware platforms such as the Raspberry Pi, making the system suitable for field-level deployment. The model is capable of accurately detecting and classifying multiple lemon leaf diseases without the need for expert supervision.

Keywords: Lemon leaf, disease detection, real-time monitoring, image processing, CNN, Tensor-Flow Lite, smart agriculture, plant disease classification, feature extraction, Raspberry Pi, precision agriculture.

I. INTRODUCTION

Agriculture plays a vital role in the economic development of many countries, particularly in regions where farming is the primary source of livelihood. Among various horticultural crops, lemon is one of the most widely cultivated citrus fruits due to its high nutritional value, medicinal properties, and commercial demand. However, lemon cultivation is significantly affected by various leaf diseases that reduce plant health, crop yield, and fruit quality. Common lemon leaf diseases such as citrus canker, anthracnose, leaf miner, sooty mound, bacterial blight, and pest infestations can spread rapidly if not detected at an early stage. Therefore, timely identification and management of these diseases are essential to ensure sustainable agricultural productivity.

Traditionally, disease detection in lemon plants relies on manual inspection by farmers or agricultural experts. This method is time-consuming, subjective, and often inaccurate, especially when symptoms are subtle or appear similar across multiple diseases. Moreover, in rural areas, access to agricultural specialists is limited, making early diagnosis difficult. As a result, diseases often remain unnoticed until they cause severe damage, leading to excessive use of pesticides, increased costs, and environmental pollution. These challenges highlight the need for an automated, accurate, and real-time disease detection system that can assist farmers in identifying plant diseases efficiently.

Recent advancements in image processing and artificial intelligence have opened new possibilities for addressing these challenges. In particular, deep learning techniques such as (CNNs) have demonstrated remarkable performance in image classification tasks, including plant disease detection. CNNs are capable of automatically learning complex features such as color variations, texture patterns, and lesion shapes directly from leaf images, eliminating the need for manual feature extraction. This makes them highly suitable for recognizing multiple disease types with high accuracy.

Real-time disease detection systems combine image acquisition, preprocessing, model inference, and result visualization into a single integrated framework. In the context of lemon leaf disease detection, such systems capture live images of leaves using a camera, preprocess the images to match the trained model's input requirements, and then classify the leaf condition using a trained CNN model. The prediction result is displayed instantly along with a confidence score, enabling farmers to take immediate preventive or corrective actions. Real-time operation is particularly important in agricultural applications, as it allows continuous monitoring of plant health directly in the field.

With the growing demand for low-cost and portable solutions, embedded platforms such as the Raspberry Pi have gained popularity in agricultural automation. Raspberry Pi acts as a compact and energy-efficient computing unit capable of running machine learning models using frameworks like Tensor Flow Lite. Tensor Flow Lite is specifically designed for deploying deep learning models on resource-constrained devices, enabling fast inference without compromising accuracy. When combined with a camera module and a graphical user interface, Raspberry Pi-based systems offer an affordable and practical solution for real-time plant disease detection.

In this project, a real-time lemon leaf disease detection system is developed using image processing and deep learning techniques. The system captures images of lemon leaves through a camera module and preprocesses them using Open-CV and NumPy to ensure compatibility with the trained CNN model. A lightweight Tensor Flow Lite model is used to classify multiple lemon leaf conditions, including healthy and diseased states. The prediction output consists of the detected disease class and a confidence percentage, calculated as the maximum probability value produced by the model. This confidence score helps users understand the reliability of the prediction.

To make the system accessible to farmers with minimal technical knowledge, a user-friendly graphical interface is implemented using Tkinter. The interface displays the live camera feed, captured leaf image, detected disease name, confidence percentage, and relevant information such as causes, prevention, and treatment measures. This integrated approach not only assists in disease identification but also provides actionable guidance to improve crop management practices. By reducing dependency on expert intervention, the system empowers farmers to make informed decisions at the right time.

The significance of real-time disease detection lies in its ability to minimize crop losses, reduce excessive pesticide usage, and promote sustainable agriculture. Early detection allows targeted treatment, preventing the spread of diseases to healthy plants. Additionally, automated systems ensure consistent and objective diagnosis, overcoming the limitations of human observation. With the increasing availability of machine learning tools and low-cost hardware, such intelligent agricultural systems are becoming more feasible and impactful.

In conclusion, real-time disease detection in lemon leaves using deep learning and embedded systems represents a promising solution to modern agricultural challenges. This project integrates image processing, CNN-based classification, Tensor Flow Lite inference, and a graphical user interface to deliver an efficient and practical disease detection system. The subsequent chapters of this report discuss the literature review, system architecture, hardware and software requirements, model development, implementation details, and experimental results, providing a comprehensive understanding of the proposed approach and its effectiveness in real-world applications.

1.1. Importance of plant disease detection

The importance of plant disease detection in a Real-Time Lemon Leaf Disease Detection system lies in protecting crop health, improving productivity, and promoting sustainable agriculture. Lemon plants are highly susceptible to diseases such as citrus canker, powdery mildew, and leaf miner, which reduce fruit yield and quality, directly affecting farmers' income and the agricultural economy. Early and accurate disease detection is essential to minimize crop losses, control disease spread, and enable effective farm management.

Traditional disease detection relies on manual observation by farmers or experts, which is time-consuming, subjective, and often inaccurate. Diseases are usually identified only after visible symptoms appear, by which time they may have spread widely. This method also encourages excessive or improper pesticide use, increasing costs and causing environmental and health risks. Hence, a technology-driven approach is necessary for timely and precise disease identification.

The integration of image processing and machine learning using a Raspberry Pi Camera enables continuous real-time monitoring of lemon leaves. The system captures images, extracts feature such as color, texture, and spots, and uses trained models to classify leaves as healthy or diseased. This automated approach ensures accurate, consistent, and rapid results.

By enabling early detection, the system helps farmers take timely corrective actions, optimize resource usage, reduce chemical dependency, and improve yields. Overall, real-time plant disease detection supports smart farming, food security, and sustainable agricultural development through affordable and innovative technology.

1.1 Importance of Real Time Image Based Detection

In this paper, real-time lemon leaf disease detection is achieved by integrating a Raspberry Pi with a camera module that captures live images of leaves directly from the agricultural field. These images are processed using advanced software tools such as Open-CV and Tensor-Flow to extract important visual features, including color variations, texture patterns, and leaf shape. A trained Convolutional Neural Network (CNN) model then analyzes these features to accurately classify the leaf as healthy or diseased. The entire process is completed within seconds, enabling instant diagnosis and immediate response. This real-time capability allows early detection of disease symptoms, helping to prevent their spread to nearby plants and reducing potential crop loss.

Beyond speed, the system significantly improves accuracy, efficiency, and consistency in disease diagnosis. Unlike manual observation, which may vary due to human error, experience, or environmental conditions, image-based analysis provides uniform and objective results. Automation eliminates guesswork and ensures reliable detection under varying field conditions such as lighting or weather. Additionally, the system reduces dependence on agricultural experts, making advanced disease monitoring accessible to small and marginal farmers. By supporting precision agriculture, the real-time detection approach optimizes resource usage, minimizes unnecessary chemical application, and promotes sustainable and efficient farming practices.

2. Literature Survey

A literature survey is a comprehensive overview of all the significant research and publications related to a specific topic or field of study. It involves identifying, analysing, and summarizing existing scholarly articles, books, reports, and other sources to understand the current state of knowledge, key themes, and research gaps. The purpose of a literature survey is to provide a solid foundation for new research by highlighting what has already been studied, the methodologies used, and the conclusions drawn. It helps researchers avoid duplication, refine their research questions, and build upon previous work to contribute new insights to the field.

The authors of [1] presents a robust multi-stage plant disease detection framework combining the Segment Anything Model (SAM) with Fully Convolutional Data Description (FCDD) for accurate leaf isolation from complex field images. The ensemble improves disease classification accuracy, generalization, and explain ability under real-world agricultural conditions.

The journals of [2] Field Plant dataset introduces real-field plant disease images collected under diverse environmental conditions. With expert annotations across multiple crops and diseases, it addresses the generalization gap of lab-based datasets and supports robust evaluation of deep learning models for real-world plant disease detection.

The writer of [3] review surveys AI-driven plant disease detection techniques using computer vision, deep learning, and transfer learning. It highlights CNN architectures, explainable AI, UAV-based imaging, and IOT integration, identifying challenges such as dataset scarcity, generalization to field conditions, and the need for lightweight deployable models.

The reporter of [4] systematic review analyzes self-supervised learning methods for plant disease detection, showing their effectiveness in reducing labeling requirements while achieving high accuracy. The study highlights SSL's robustness, faster convergence, and potential for real-field deployment, while addressing challenges like domain shift and computational complexity.

The authors of [5] work presents an attention-augmented residual network integrated with Faster R-CNN for precise plant disease detection. Attention modules and GAN-based data augmentation enhance feature localization and accuracy, achieving near-human performance while improving robustness against background noise and limited training data.

The authors of [6] is RAI-Net architecture combines residual learning, channel attention, and inception modules for tomato disease classification. The lightweight yet expressive model captures multi-scale disease features efficiently, achieving high accuracy and interpretability through Grad-CAM, making it suitable for mobile and edge-based agricultural applications.

The journals of [7] study proposes a hybrid framework that fuses deep CNN features with Local Binary Pattern texture descriptors for multi-class plant disease classification. Feature fusion improves discriminative power and robustness, achieving superior accuracy compared to standalone CNN or handcrafted feature-based methods.

The writer of [8] provides a comprehensive comparison of machine learning and deep learning models for plant disease classification and detection. It demonstrates the superiority of CNN-based and object detection frameworks over traditional ML methods, emphasizing transfer learning, GAN augmentation, and hyperspectral imaging for precision agriculture.

The authors of [9] evaluates deep CNN architectures for rice plant disease classification. By comparing Res-Net, VGG, Inception, and Dense-Net models, the study demonstrates that deep learning significantly outperforms traditional approaches, enabling accurate and scalable automated rice disease diagnosis.

The reporter of [10] introduces an optimal fuzzy deep neural network for plant disease detection using UAV-based remote sensing data. The fusion of fuzzy logic and deep learning enhances classification accuracy and uncertainty handling, enabling effective large-scale crop monitoring under variable environmental conditions.

Summary of literature survey

This literature survey of the project "Real Time Disease Detection in Lemon Leaves," synthesizes research on leveraging computer vision and deep learning to automate plant disease diagnostics. This automation is essential for addressing foliar diseases like citrus canker and greasy spot to improve yield and reduce pesticide misuse.

The core of the research involves advanced deep Convolutional Neural Networks (CNNs) and transfer learning, with models like EfficientNetB0 and DenseNet-121 demonstrating high classification accuracy (up to 99.69% on controlled data). A major challenge addressed is the poor performance of models trained on laboratory data when applied to complex, real-world field images. Solutions include creating new field-specific datasets (like Field Plant) and developing sophisticated pre-processing pipelines. One such pipeline integrates the Segment Anything Model (SAM) for object segmentation with Fully Convolutional Data Description (FCDD) to accurately discriminate leaf regions from complex backgrounds, leading to over 10% improvement in classification accuracy.

Research is also moving towards lightweight models (like MobileNetV2 and ShuffleNetv2) suitable for deployment on edge devices, drones (UAVs), and mobile applications for real-time monitoring. Crucially, the integration of Explainable Artificial Intelligence (XAI) techniques, such as LIME, enhances transparency by showing users which part of a leaf image influenced the prediction, building trust in the AI-driven diagnostic system.

While significant progress has been made toward robust, scalable, and interpretable systems, challenges remain in handling environmental variations, occlusion, and the scarcity of large, field-annotated datasets. The overall trend in the literature favors hybrid models and continued adaptation to transform traditional farming into data-driven precision agriculture.

2.1. Problem Statement

Agricultural productivity is strongly affected by crop health, with leaf diseases being a major cause of yield and quality loss. Conventional manual disease identification is time-consuming, error-prone, and often inaccessible in rural regions due to limited expert availability, leading to delayed diagnosis and economic losses. To address this, the proposed work develops a real-time plant leaf disease detection system using computer vision and artificial intelligence. Live leaf images are captured and preprocessed using a Raspberry Pi, then analyzed on a laptop using Open-CV- and NumPy-based algorithms to automatically detect and classify diseases such as anthracnose, bacterial blight, citrus canker, and leaf curl virus. The system also provides disease descriptions, causes, and recommended remedies through a Tkinter-based graphical interface, offering a user-friendly and practical solution for precision agriculture.

2.2. Objectives

The paper aims to build a real-time system that captures and pre-processes leaf images, detects and classifies diseases using AI and Open-CV, and displays results through a Tkinter interface. It provides disease details and remedies, reduces manual dependency, supports farmers, and improves timely diagnosis for better crop health.

- **To develop a real-time leaf image acquisition system:** Design and implement a setup using a Raspberry Pi camera to capture live images of lemon leaves in varying environmental conditions. The objective is to ensure clear, high-resolution inputs for reliable disease detection.
- **To pre-process and enhance leaf images for accurate analysis:** Use Open CV and Numpy to perform pre-processing steps such as noise removal, resizing, segmentation, and contrast enhancement. This prepares the captured images for efficient feature extraction and classification.
- **To provide detailed disease information and suitable remedies:** Integrate a database that displays disease symptoms, causes, severity, and recommended cure or preventive measures, helping farmers take timely action.
- **To support early diagnosis for improving crop productivity:** Enable farmers to identify leaf diseases at an early stage, reducing crop loss, improving yield quality, and promoting sustainable precision agriculture.

3. Block diagram

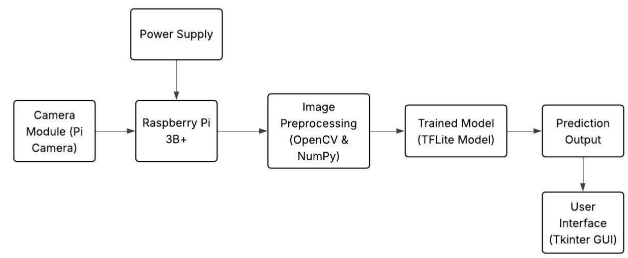


Figure-1: Block diagram of proposed methodology

The fig. 1 shows the working process of a real-time lemon leaf disease detection system that uses a Raspberry Pi camera for image acquisition and a deep learning model for classification. The system is designed to automatically capture images of lemon leaves, analyse them, and identify whether they are healthy or affected by any disease. Each component in the block diagram plays a vital role in ensuring smooth data flow and accurate prediction results.

The Power Supply is the fundamental unit that provides the necessary operating voltage and current to all components in the system. A stable and reliable power source ensures continuous operation of the Raspberry Pi board and the connected camera module. Without a proper power supply, the system may experience interruptions or inaccurate data processing. Therefore, it serves as the backbone of the hardware setup.

The Camera Module (Pi Camera) is responsible for capturing real-time images of lemon leaves. It acts as the primary input device in the system. The camera continuously monitors the plant and takes images that are sent to the Raspberry Pi for further analysis.

The quality and resolution of these captured images play an important role in ensuring accurate detection since high-quality images allow for better feature extraction and classification by the trained model. The Raspberry Pi (Model 3B+) functions as the central processing unit of the project. It controls the camera operations, performs pre-processing on the captured images, and executes the machine learning model for disease detection. The Raspberry Pi is chosen because of its compact size, low power consumption, and ability to handle real-time image processing efficiently. It serves as the bridge between the hardware components (camera and power supply) and the software components (image processing and prediction modules).

The next stage is Image Processing, which involves cleaning and preparing the captured images for model prediction. Libraries like Open-CV and NumPy are used for this purpose. Image processing includes steps such as resizing, noise reduction, segmentation, and feature extraction. These operations enhance the image quality and isolate the regions of interest, making it easier for the trained model to recognize patterns associated with specific diseases. Proper image pre-processing significantly improves the overall accuracy of the detection system.

After pre-processing, the processed image is fed into the Trained Model (TFLite Model). This model is a lightweight version of a Tensor Flow deep learning network, optimized to run efficiently on the Raspberry Pi. The model has been trained using a dataset of lemon leaf images containing both healthy and diseased samples. It classifies the input image into categories based on the patterns it has learned during training. The model outputs a prediction indicating whether the leaf is healthy or infected with a particular disease, such as leaf spot or blight.

The Prediction Output is then passed to the User Interface (Tkinter GUI), which presents the results in a simple and user-friendly manner. The interface displays the captured image, the predicted disease type, and the confidence percentage of the model's output. This helps farmers or researchers easily identify the condition of the leaf and take necessary actions in real time. The system efficiently integrates hardware and software components to achieve automated, real-time detection of lemon leaf diseases. From capturing the image to displaying the prediction results, each module plays a crucial role in improving agricultural productivity and reducing manual inspection efforts.

Raspberry Pi 3B+ (Main Controller)

Figure-2: Raspberry Pi 3B+ (Main Controller)

In a real-time leaf-disease detection system, the Raspberry Pi 3B+ acts as the edge-computing hub the small, low-cost "brain" that ties together sensing, computation, and action. A camera module connected to the Pi captures live images of leaves (for example, lemon-tree leaves) in the field. The Pi then pre-processes those images (resizing, normalization) and feeds them into a lightweight or optimized deep-learning model (e.g. a CNN or quantized model) deployed locally. Because the Pi supports deployment frameworks like Tensor Flow Lite or edge-optimized CNNs, the inference (disease classification) can happen directly on-device without needing internet or powerful GPU servers. Once the model predicts whether a leaf is healthy or diseased (and possibly the disease type), the Pi can immediately trigger downstream actions for example, log the result, alert the user, or activate other hardware (like a spray pump, irrigation control, or notification system) via its GPIO pins.

Because Raspberry Pi 3B+ is compact, power-efficient, and relatively inexpensive, it makes an autonomous, portable disease-detection unit feasible — suitable for small farms or remote areas where internet/cloud connectivity may be unreliable. Its use of on-device processing ensures low latency, real-time detection, enabling early diagnosis and timely intervention, which is especially valuable for perishable crops like lemon.

Camera control (image capture), Image pre-processing using Open-CV & NumPy, Loading and running the Tensor Flow Lite model, Displaying output on the Tkinter GUI. Controls all operations and performs disease detection using the trained model.

Additionally, because all processing happens locally, data privacy and security are better controlled, and farmers can use the system without depending on cloud infrastructure. Raspberry Pi 3B+ in such a project captures leaf images, runs disease-detection model locally, outputs classification (healthy/diseased + disease type), optionally actuates hardware or alerts user, delivering a cost-effective, portable, real-time, on-site disease-monitoring solution for agriculture.

3.1.2 Pi Camera Module

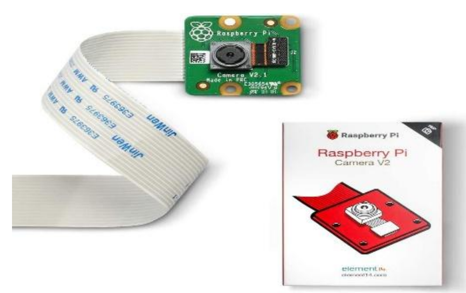


Figure-3: Pi Camera Module

In a real-time lemon (or other plant) leaf disease detection system, the Raspberry Pi camera module acts as the primary visual sensor, continuously capturing leaf images directly in the field. Connected via the Pi's CSI interface, the camera provides real-time RGB image frames at adequate resolution for accurate analysis. These images are immediately processed on the Raspberry Pi using computer vision techniques and a pre-trained deep learning model, such as a CNN deployed with Tensor Flow Lite. Since all processing is performed locally, the system operates without cloud dependency, ensuring low latency and offline functionality. Before prediction, the captured images undergo pre-processing using Open-CV and NumPy. This includes resizing to 224 × 224 pixels, converting to RGB format, and normalizing pixel values between 0 and 1 to match the model's training conditions. The trained model then classifies the leaf as healthy or diseased. Based on the result, the Pi can trigger alerts or control actuators through GPIO pins. This compact, low-cost setup enables efficient, real-time, and field-deployable disease monitoring.

3.1.3 Tkinter GUI (Graphical User Interface)

In a real-time lemon-leaf disease detection setup, once a deep-learning model is trained (on a dataset of lemon-leaf images: healthy + diseased under various conditions), converting it to a TFLite file makes it lightweight and efficient enough to run on an embedded system like Raspberry Pi 3B+. The .tflite model becomes the core intelligence of the system: after a leaf image is captured (e.g. by a connected camera module) and pre-processed (resized, normalized, colour-corrected), the Raspberry Pi loads the TFLite model using the TFLite runtime (or a TFLite Interpreter in Python), sets the pre-processed image as input tensor and invokes inference.

During inference, the model processes the image through a series of convolutional layers (depth wise and pointwise convolutions, as per MobileNetV2's lightweight architecture) that extract hierarchical feature representations — leaf veins, lesion patterns, colour changes, texture irregularities associated with different diseases — while keeping computational load low enough for the Pi's CPU. The output is a probability distribution (or confidence scores) over the pre-defined classes (e.g. "Canker", "Healthy", "Rust", etc.). The index of the highest-confidence class is taken as the predicted label.

Because TFLite models are optimized — often quantized — they require far less memory and CPU/GPU than full-scale models. This allows the Raspberry Pi to run classification quickly (often within fractions of a second to a few seconds per image) — enabling near real-time detection. For example, similar deployments on Pi for leaf-image classification have shown inference times acceptable for on-site operations.

Once the Pi receives the prediction, the system can immediately act: display the result, log it, alert the user (e.g. via buzzer, message), or even trigger automated responses (spraying, irrigation, etc.) via Pi's GPIO pins — depending on how the larger system is configured.

In essence, the trained TFLite model transforms raw image data (from the camera) into semantic, actionable information (disease/no-disease, disease type) in real time. This makes the entire system portable, low-cost, on-site, and autonomous — no need for cloud access or powerful hardware. The model is the "brain" of the system: it encodes disease knowledge, lets the Pi "understand" leaf appearances, and enables fast classification so that farmers or automated agents can respond immediately to detected plant diseases.

Disease Information & Class Mapping

In a real-time lemon-leaf disease detection system, once the core model classifies an input leaf image and outputs a class index (for example, "3" or "7"), the system must translate that numeric index into a human-readable disease name. It maps each class index (e.g., 0, 1, 2, …) to a meaningful label (e.g., "Canker", "Healthy", "Leaf rust", etc.). This mapping enables the system to present results to users in a way they can understand (rather than arbitrary integer codes).

But classification alone is often not enough for a farmer or user — knowing just "disease: Canker" or "disease: Healthy" doesn't always help decide what to do next. Therefore, the system also includes a richer knowledge base stored in something like disease_info.json, which for each disease name stores additional details: typical symptoms (leaf spots, yellowing, wilting), likely causes (fungal, bacterial, environmental), recommended prevention measures (good drainage, avoid overhead irrigation, crop rotation), and treatment advice (appropriate fungicide or pesticide, dosage, timing, sanitation steps). Upon detection, the system uses the disease name (from the class mapping) as a key to retrieve the corresponding detailed information from disease_info.json.

This function — mapping class → disease name → disease information — significantly enhances the utility of the system. Instead of simply telling "the leaf is diseased," the system becomes a lightweight expert assistant: it not only diagnoses but also educates the user on what disease it is, what visible symptoms to watch for, how to prevent its spread, and what remedial actions to take. This is especially valuable for farmers or users who may not have formal training in plant pathology.

Moreover, bundling this mapping in JSON files decouples model logic (the classification network) from domain knowledge (disease descriptions). That makes it easier to update or extend: if a new disease class is added or if treatment recommendations change, you just update the JSON — no need to re-train the model. Finally, in a real-time, field-deployable setup (e.g. using a camera module + edge-device like Raspberry Pi), this mapping + information retrieval happens immediately (after model inference), making the system actionable on the spot. The user sees not just "Canker" — but "Canker: corky lesions on bark and leaf margins; avoid waterlogging; treat with fungicide X at Y concentration," thus enabling timely and informed intervention rather than just detection.

ESP32 Wi-Fi Camera

In a real-time lemon-leaf disease detection system, Tkinter serves as the graphical user interface (GUI) layer that ties together all components — camera feed, image capture, model inference, and user output — into a cohesive, user-friendly application. As the built-in GUI toolkit for Python, Tkinter allows you to create windows, buttons, labels, canvases, and other widgets easily and cross-platform.

When the system starts, Tkinter opens a main window. One part of this window displays the live camera feed from the connected camera (e.g. a webcam or a Raspberry Pi camera module). Under the hood, a video-capture library such as Open-CV reads frames continuously from the camera, converts each frame into a format suitable for Tkinter (e.g. using PIL/Pillow to convert an Open-CV image array to an Image Tk object), and updates a Label or Canvas widget in the Tkinter window at regular intervals (e.g. every 10–30 ms). This creates the effect of a "live video stream" embedded inside the GUI. The GUI also includes control buttons — for example:

- Live: start/stop the live feed,
- Capture: freeze or save a captured frame,
- Predict: take the captured/ current frame and run the trained disease-detection model on it,
- Clear: reset the display or clear the last result,
- Exit: close the application safely.

Once "Predict" is triggered, the application feeds the current image (after any pre-processing: resizing, normalization etc.) to the model (for instance a TF-Lite model) and obtains the predicted class (disease or healthy). The GUI then displays: the captured leaf image, the predicted disease name (mapping from class index → disease name), and optionally additional information — confidence/accuracy percentage, symptoms, causes, prevention and treatment suggestions (using a lookup from a JSON or database). This design offers several practical benefits:

- User-friendly interaction: Farmers or users don't need to write code — they just click buttons.
- Real-time, on-site detection: Because the GUI runs on the device (e.g. a Raspberry Pi) with live camera feed + model inference, detection happens instantly — enabling timely intervention.
- Integrated workflow: Capture → Prediction → Information display — all in one window — streamlines the diagnostic process.
- Extend ability: Additional features (logging results, saving images, alerting, hardware control for spraying, etc.) can be easily added via buttons or additional Tkinter widgets.

In short, Tkinter functions as the front-end interface that makes the disease-detection system interactive, accessible, and usable transforming raw camera + model output into a complete, practical application for real-world farming use.

Output Display

In the system, after the model predicts a class for a leaf image (say "Canker" or "Healthy"), the output display presents the result in a clear and meaningful way: it shows the detected disease name, the confidence score (accuracy percentage) of the prediction, and recommended preventive or treatment measures (drawn from a knowledge base).

This final presentation converts raw model output class indices and probabilities into actionable information that a user can understand and use. Because the GUI (built with a toolkit like Tkinter in Python) shows the captured or live image side-by-side with the predicted result and additional guidance, a farmer doesn't need technical knowledge to interpret the output. Studies and applications in plant disease detection have shown that integrating detection with recommendation, via user-friendly interfaces, helps farmers make timely decisions on treatment and prevention. For example, the system can display: "Detected disease: Leaf spot, Confidence: 95% Recommendation: Remove infected leaves, apply fungicide XYZ, ensure proper drainage." Thus, the user immediately knows what is wrong and what to do next — no manual inspection or expert consultation required. Such integration significantly lowers the barrier for adoption.

Beyond just diagnosis, the output display acts as a digital assistant summarizing not only what but also why and how to respond. This level of guidance is particularly useful for small-scale farmers or those without ready access to agronomists. Indeed, some apps and research systems highlight that delivering both diagnosis and treatment advice boosts usability and trust among end users.

Finally, presenting the confidence score helps users gauge the reliability of the prediction. If the score is low, a farmer might decide to take a second image or consult further, avoiding blind trust in uncertain predictions. This transparency helps build confidence in the system and ensures more prudent decision-making.

In short: the Output Display is the bridge between AI inference and practical agriculture. It transforms technical outputs into understandable, actionable, and user-centric information enabling timely disease detection, informed decision-making, and effective crop management.

4. Hardware and Software Used

The hardware requirements of the proposed lemon leaf disease detection system are designed to ensure efficient image acquisition, processing, and result visualization. At the core of the system is the Raspberry Pi 3 Model B+, which functions as a compact and affordable mini-computer. It is responsible for executing the disease detection program, processing captured leaf images, running the trained machine learning model, and displaying the output results. Due to its sufficient processing power and GPIO support, the Raspberry Pi 3 Model B+ is well suited for real-time agricultural monitoring applications.

To capture high-quality images of lemon plant leaves, a Raspberry Pi Camera Module is used. This camera is directly connected to the Raspberry Pi through the CSI interface, enabling fast and reliable image transfer. It allows the system to acquire clear, real-time images under natural lighting conditions, which are essential for accurate disease detection and classification. The captured images are immediately sent to the processing unit for analysis.

A micro SD card serves as the primary storage component of the system. It stores the operating system, Python scripts, trained disease detection model, and all required software libraries. The SD card enables smooth booting of the Raspberry Pi and provides sufficient memory for storing images and program data.

Power is supplied using a stable 5V, 3A power adapter, which ensures uninterrupted operation of the Raspberry Pi and the connected camera module. A reliable power source is critical to prevent system crashes or data loss during image processing tasks.

For user interaction and result visualization, an HDMI display monitor is connected to the Raspberry Pi. The monitor displays the graphical user interface, live camera feed, detected disease name, and accuracy percentage. Additionally, a keyboard and mouse are used to control the system, execute programs, and interact with the interface, making the system user-friendly and practical for real-time agricultural use.

The software requirements of the Lemon Leaf Disease Detection System are selected to support efficient image processing, machine learning inference, and user interaction on a Raspberry Pi platform. The system runs on Raspberry Pi OS, which is a Linux-based operating system specifically optimized for Raspberry Pi hardware. It provides built-in support for essential features such as GPIO control, camera interfacing, networking, and package management, creating a stable environment for running Python-based applications.

The core application is developed using the Python programming language, which controls the camera module, processes captured images, loads the trained machine learning model, and displays prediction results on the graphical user interface. Python is chosen due to its simplicity, readability, and extensive support for image processing and machine learning libraries. For executing the trained deep learning model efficiently on resource-constrained devices**,** Tensor Flow Lite is used. This lightweight framework enables fast and low-power inference, allowing the system to accurately classify lemon leaves as healthy or diseased in real time.

Image processing tasks are handled using the Open-CV (Open-Source Computer Vision Library). Open-CV is used for operations such as image resizing, cropping, filtering, and normalization, ensuring that the captured images are properly prepared before being passed to the trained model. For building a user-friendly interface, Tkinter, a standard Python GUI library, is employed. The GUI allows users to capture images, view the live camera feed, and display disease prediction results along with accuracy percentages.

Additionally, Num-Py is used for numerical computations and for converting image data into array formats compatible with the machine learning model. Code development and execution are carried out using the Thorny IDE or the Raspberry Pi terminal, which provide a simple environment for writing, running, and debugging Python programs. Other supporting libraries such as Pillow (PIL) for image handling, json for managing class labels and output data, and os/sys for file and system operations are also utilized. Together, these software components enable automatic detection and classification of multiple lemon leaf diseases while displaying detailed disease information, causes, prevention, and treatment measures through an interactive GUI.

5. Results & Discussions

The expected output of the real-time lemon leaf disease detection system using a Raspberry Pi camera is to provide accurate and instant identification of lemon leaf health status. The process begins when the Raspberry Pi camera captures live images of lemon leaves in the field. These images are processed using Open-CV and NumPy to enhance clarity, remove noise, and prepare them for analysis. The pre-processed image is then sent to a trained Tensor Flow Lite (TFLite) model, which detects and classifies whether the leaf is healthy or affected by a specific disease such as citrus canker, sooty Mold, or leaf spot. The prediction results are displayed on a Tkinter-based graphical user interface (GUI), making it simple for users to view and understand the disease status. Thus, the system's expected output is a clear, real-time, and reliable diagnosis that helps farmers take immediate action, reduce crop losses, and maintain healthy lemon plantations efficiently.

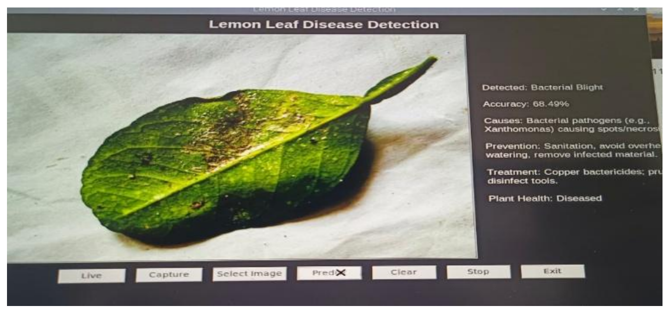


Figure -4: Bacterial Blight Disease Detection

The figure 4, displays a Lemon Leaf Disease Detection system analyzing a diseased leaf. The system identifies the condition as Bacterial Blight with 68.49% accuracy. It highlights possible bacterial causes, suggests prevention and treatment methods, and indicates the plant health status as diseased.

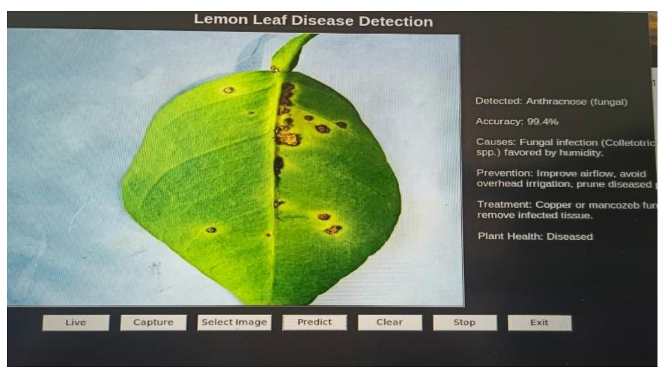


Figure-5: Anthracnose Disease Detection

The figure 5, shows a Lemon Leaf Disease Detection system identifying Anthracnose, a fungal disease, with 99.4% accuracy. Visible brown lesions confirm infection caused by Colletotrichum species. The system provides causes, prevention measures, treatment options, and marks the plant health status as diseased.

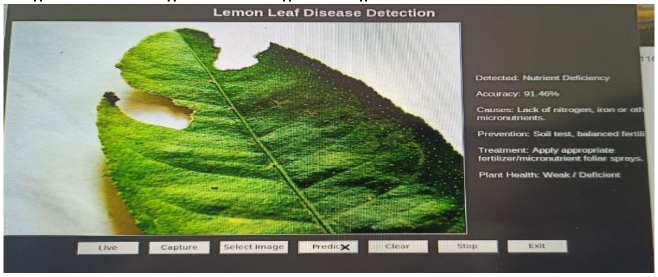


Figure-6: Nutrient Deficiency Disease Detection

The figure 6, presents a Lemon Leaf Disease Detection system identifying Nutrient Deficiency with 91.46% accuracy. The leaf shows discoloration and weakness caused by lack of essential nutrients like nitrogen or iron. The system suggests soil testing, balanced fertilization, and micronutrient sprays, indicating weak plant health.

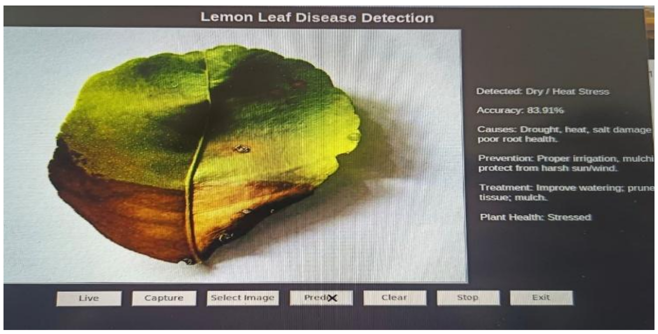


Figure-7: Dry/ Heat Stress Disease Detection

The figure 7, shows the graphical user interface of a Lemon Leaf Disease Detection System. On the left side, a captured image of a lemon leaf is displayed for analysis. The right side presents the detection results, including the identified disease type, confidence or accuracy level, and additional information such as causes, prevention methods, and treatment measures. In this example, the system detects leaf dry or heat stress, indicating that the plant is under environmental stress. The bottom section contains control buttons like live view, capture, select image, predict, clear, stop, and exit, allowing easy interaction. Overall, the interface provides clear, user-friendly real-time disease diagnosis support.

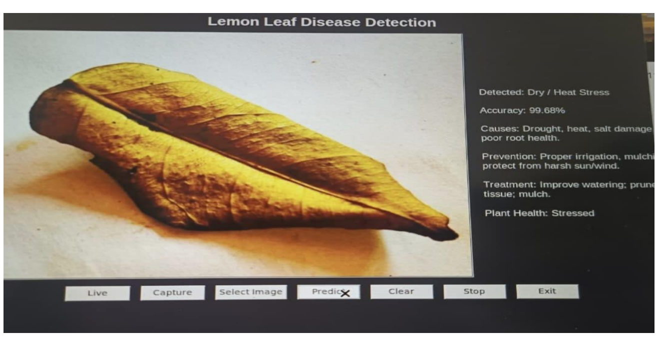


Figure-8: Dry/ Heat Stress Disease Detection

The figure 8, illustrates a Lemon Leaf Disease Detection system diagnosing Dry/Heat Stress with 99.68% accuracy. The leaf appears dried and discolored due to drought, high temperature, or poor root conditions. The system suggests proper irrigation, mulching, and protection from harsh environmental stress, indicating stressed plant health.

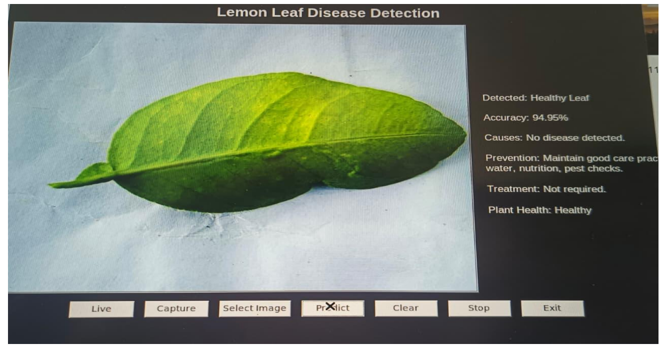


Figure-9: Lemon Healthy Leaf Disease Detection

The figure 9, a Lemon Leaf Disease Detection system classifying the leaf as Healthy with 94.95% accuracy. No disease symptoms are observed. The system confirms good plant health and recommends maintaining proper care practicessuch as adequate watering, nutrition, and regular pest monitoring.

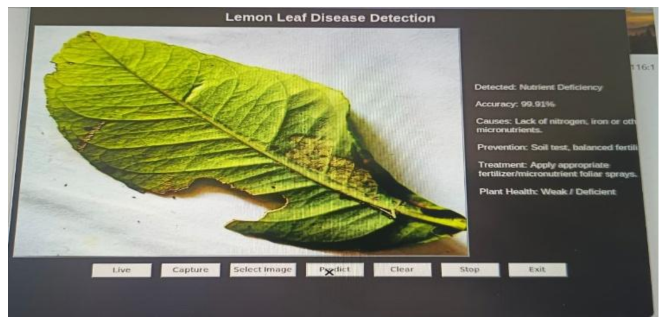


Figure-10: Nutrient Deficiency Disease Detection

The figure 10, shows a Lemon Leaf Disease Detection system diagnosing Nutrient Deficiency with 99.91% accuracy. The leaf displays discoloration and damage due to insufficient nitrogen, iron, or micronutrients. The system recommends soil testing, balanced fertilization, and micronutrient sprays, indicating weak and deficient plant health.

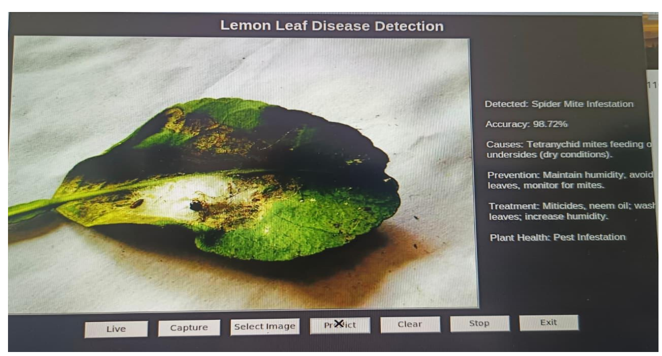


Figure-11: Spider Mite Infestation Disease Detection

The figure 11, detects Spider Mite infestation on a lemon leaf with 98.72% accuracy. The disease is caused by mites feeding on leaf undersides, common in dry conditions. Prevention includes maintaining humidity and monitoring leaves. Treatment involves neem oil, matricides, washing leaves, and improving plant health.

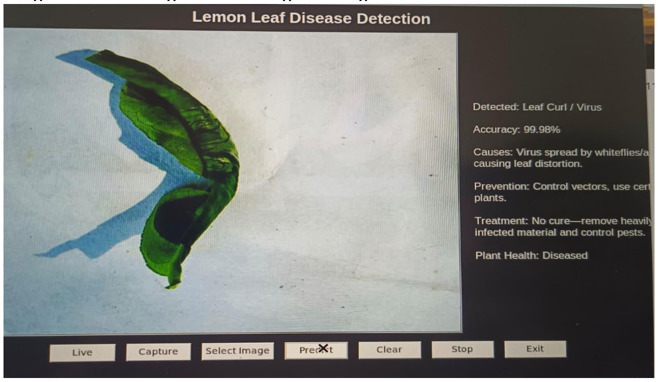
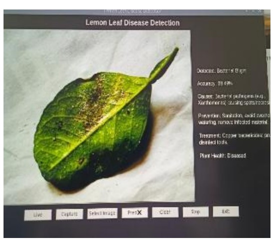
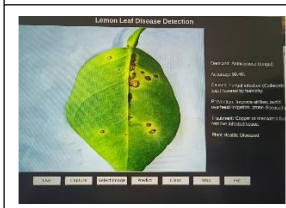
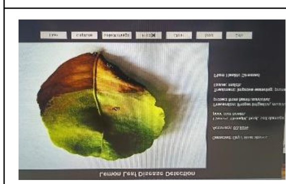


Figure -12: Curl/Virus Disease Detection

The figure 12, identifies Leaf Curl Virus in a lemon leaf with 99.98% accuracy. The disease is caused by viruses spread through whiteflies, leading to leaf curling and distortion. Prevention includes controlling insect vectors and using healthy plants. There is no direct cure; infected leaves should be removed and pests controlled.

Leaf image	**Disease caused**	**Accuarcy**
	Curl/Virus Disease	99.98%
	Bacterial Blight Disease	68.49%
	Anthracnose Disease	99.4%
	Nutrient Deficiency Disease	91.46%
	Leaf Dry/ Heat Stress Disease	83.91%
	Leaf Dry/ Heat Stress Disease	99.68%
	Healthy Leaf Disease	94.95%
	Healthy Leaf Disease	99.91%
	Nutrient Deficiency Disease	99.91%
	Spider Mite Infestation Disease	98.72%

Formula for Calculating Accuracy:

$$\text{Accuracy (Confidence \%)} = \max\ (\text{Model Output Probabilities}) \times 100$$

Code for Finding Accuracy:

```
ouptut= interpreter.get,tenso{ output_details
idx = int(np.argmx(output)
conf = float(np.max(output)) 100
```

Accuracy(Confidence %) = max(Model Output Probabilities) x100**defines how the confidence score is calculated. This formula indicates that the highest probability value produced by the model for a given input image is selected and converted into a percentage to represent prediction confidence. The lower part of the image shows the corresponding** Python code implementation **using a Tensor Flow Lite interpreter. First, the output tensor is extracted from the trained model using interpreter get tensor( ), which contains probability values for all disease classes. Next, np.argmax(output) identifies the index of the class with the highest probability, which represents the predicted disease category. Finally, np.max(output) retrieves the maximum probability value from the output tensor, and multiplying it by 100 converts it into a confidence percentage. Together, the formula and code demonstrate the direct relationship between theoretical computation and real-time system execution. This approach ensures that**

users receive not only the predicted disease name but also a meaningful confidence score, helping farmers and researchers assess the reliability of the detection result and make informed decisions regarding crop management.

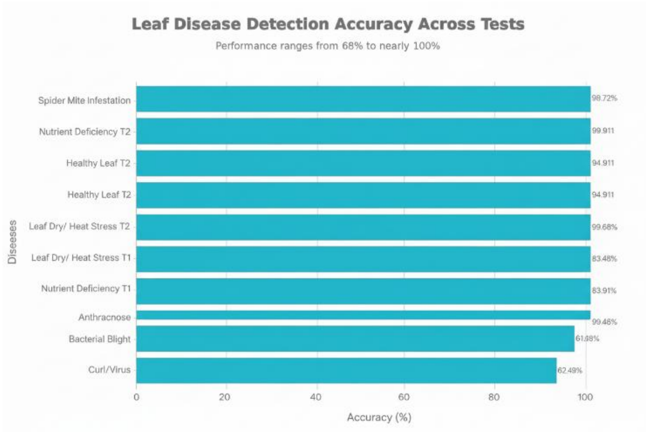


Fig-12: Leaf Disease Detection Accuracy Across Tests

The Fig-12, presents a horizontal bar chart titled "Leaf Disease Detection Accuracy Across Tests", which illustrates the performance of a leaf disease detection model across multiple disease categories and testing conditions. The x-axis represents accuracy in percentage, while the y-axis lists various leaf disease classes and test scenarios (T1 and T2). Overall, the chart highlights that the model performs with high reliability, with accuracy values ranging from approximately 68% to nearly 100%, as indicated in the subtitle. From the chart, diseases such as Curl Virus, Anthracnose, Healthy Leaf T2, Nutrient Deficiency T2, and Leaf Dry/Heat Stress T2 achieve accuracies close to 99–100%, demonstrating excellent classification capability. This suggests that the model is highly effective in identifying clear visual patterns associated with these conditions, particularly under the second testing setup (T2), which may represent improved lighting, better image quality, or enhanced pre-processing. Moderately high performance is observed for Healthy Leaf T1, Nutrient Deficiency T1, and Leaf Dry/Heat Stress T1, with accuracy values ranging between 82% and 94%. The slight drop compared to T2 results indicates that earlier test conditions or more challenging image variations affected prediction confidence. This highlights the importance of consistent data quality and adequate training samples. The lowest accuracy is observed for Bacterial Blight, which falls near 68–70%. This suggests that the visual symptoms of bacterial blight may overlap with other diseases or appear less distinct, making classification more challenging for the model. It also indicates an area where additional training data or feature enhancement could improve performance. In summary, the chart demonstrates that the proposed leaf disease detection system performs exceptionally well for most disease classes, especially under optimized test conditions. While a few categories show relatively lower accuracy, the overall results confirm the robustness and effectiveness of the model for real-world agricultural disease monitoring and decision support applications.

Conclusion

The Real-Time Lemon Leaf Disease Detection system demonstrates the effective integration of image processing, artificial intelligence, and embedded systems to address one of the major challenges in modern agriculture—early and accurate identification of plant diseases. Lemon cultivation is highly vulnerable to a variety of leaf diseases caused by bacterial, viral, fungal, and pest-related factors, which directly impact crop yield, fruit quality, and farmers' income. Traditional disease detection methods based on manual inspection are time-consuming, subjective, and often unreliable, especially in large farming areas or regions with limited access to agricultural experts. The proposed real-time detection system successfully overcomes these limitations by providing an automated, efficient, and reliable solution for monitoring lemon leaf health.

By utilizing a Raspberry Pi camera for live image acquisition and applying image pre-processing techniques such as noise removal, segmentation, and contrast enhancement, the system ensures high-quality inputs for analysis. The incorporation of machine learning and deep learning models, particularly convolutional neural networks, enables accurate classification of lemon leaf conditions into healthy or diseased categories. Diseases such as leaf curl virus, spider mite infestation, citrus canker, and other common infections can be detected with high accuracy in real time. The ability of the system to display disease type, confidence level, causes, prevention methods, and treatment recommendations through a user-friendly graphical interface further enhances its practical usefulness for farmers.

One of the most significant strengths of this project lies in its real-time capability. Immediate detection allows farmers to take timely preventive or corrective measures, thereby reducing the spread of disease and minimizing crop loss. Early diagnosis not only improves yield and fruit quality but also reduces unnecessary pesticide usage, promoting environmentally sustainable farming practices. The system supports precision agriculture by enabling data-driven decision-making and efficient resource utilization. Additionally, the use of low-cost and energy-efficient hardware such as Raspberry Pi makes the solution affordable and accessible to small-scale and marginal farmers, which is crucial for widespread adoption.

The project also highlights the potential of scalable and adaptable agricultural technologies. With further enhancements, the system can be expanded to support cloud-based data storage, remote monitoring, and mobile notifications, enabling farmers to access disease information anytime and anywhere. Integration with Internet of Things (IOT) platforms and weather data can further improve disease prediction accuracy and help in developing preventive strategies. Moreover, the dataset collected through continuous monitoring can contribute to long-term agricultural research and the development of more robust predictive models.

The Real-Time Lemon Leaf Disease Detection system represents a significant step toward smart and sustainable agriculture. It bridges the gap between traditional farming practices and modern technological solutions by providing an intelligent, automated, and user-friendly disease detection mechanism. The successful application of image processing and AI in this project demonstrates how emerging technologies can transform agricultural practices, empower farmers, and contribute to food security. By enabling early detection, reducing crop losses, and supporting eco-friendly farming, this system not only improves lemon cultivation but also serves as a strong foundation for future advancements in plant disease detection and precision agriculture.

References

[1].E. Moupojou, F. Retraint, H. Tapamo, M. Nkenlifack, C. Kacfah and A. Tagne, "Segment Anything Model and Fully Convolutional Data Description for Plant Multi-Disease Detection on Field Images," in IEEE Access, vol. 12, pp. 102592-102605, 2024, doi: 10.1109/ACCESS.2024.3433495.

[2].E. Moupojou et al., "FieldPlant: A Dataset of Field Plant Images for Plant Disease Detection and Classification With Deep Learning," in IEEE Access, vol. 11, pp. 35398-35410, 2023, doi: 10.1109/ACCESS.2023.3263042.

[3].V. Balafas, E. Karantoumanis, M. Louta and N. Ploskas, "Machine Learning and Deep Learning for Plant Disease Classification and Detection," in IEEE Access, vol. 11, pp. 114352-114377, 2023, doi: 10.1109/ACCESS.2023.3324722.

[4].S. Babu, M. Maravarman and R. Pitchai, "Detection of Rice Plant Disease Using Deep Learning Techniques," in Journal of Mobile Multimedia, vol. 18, no. 3, pp. 757-770, May 2022, doi: 10.13052/jmm1550-4646.18314.

[5].N. Nigar, H. Muhammad Faisal, M. Umer, O. Oki and J. Manappattukunnel Lukose, "Improving Plant Disease Classification With Deep-Learning-Based Prediction Model Using Explainable Artificial Intelligence," in IEEE Access, vol. 12, pp. 100005-100014, 2024, doi: 10.1109/ACCESS.2024.3428553.

[6].M. Pajany, S. Venkatraman, U. Sakthi, M. Sujatha and M. K. Ishak, "Optimal Fuzzy Deep Neural Networks-Based Plant Disease Detection and Classification on UAV-Based Remote Sensed Data," in IEEE Access, vol. 12, pp. 162131-162144, 2024, doi: 10.1109/ACCESS.2024.3488751.

[7].D. S. Joseph, P. M. Pawar and K. Chakradeo, "Real-Time Plant Disease Dataset Development and Detection of Plant Disease Using Deep Learning," in IEEE Access, vol. 12, pp. 16310-16333, 2024, doi: 10.1109/ACCESS.2024.3358333.

[8].A. Bhargava, A. Shukla, O. P. Goswami, M. H. Alsharif, P. Uthansakul and M. Uthansakul, "Plant Leaf Disease Detection, Classification, and Diagnosis Using Computer Vision and Artificial Intelligence: A Review," in IEEE Access, vol. 12, pp. 37443-37469, 2024, doi: 10.1109/ACCESS.2024.3373001.