BRAUDE
College of Engineering, karmiel

Software Engineering Department
Braude College
Capstone Project Phase A

Lemon Health Monitoring

By:
David Kotler
Maxim Cherepanov

Advisor:
Dr. Naomi Unkelos Shpigel

Project Code:
25-2-D-12

Link to GitHub:
https://github.com/semaximche/Capstone-Project-Lemon-Health-Monitor

Table of Contents

Abstract **3**
1. Introduction **4**
2. Literature Review **5**
2.1 Lemon Plant Diseases 5
2.2 Visual Disease Identification 7
2.3 Machine learning models and evaluation 8
3. Research and Technology Review **8**
3.1 Meetings with agricultural farm manager Keren 8
3.2 Data Collection 9
3.3 Backend Technologies 10
3.3.1 Python 10
3.3.2 FastAPI 10
3.3.3 Google Auth 10
3.4 Frontend Technologies 11
3.4.1 React 11
3.4.2 Tailwind CSS 11
3.5 Machine Learning Technologies 11
3.5.1 YOLOv11 11
3.5.2 EfficientNetV2 12
3.5.3 OpenAI API 12
3.6 Algorithm 13
4. System Design **14**
4.1 Requirements 15
4.2 Architecture Diagram 17
4.3 User Diagram 18
4.4 Activity Diagram 19
4.5 Objects Diagram 20
5. Expected Achievements **21**
5.1 Challenges 21
5.2 Evaluation 21
6. Testing **22**
6.1 Testing Approach 22
6.1.1 Model Testing 22
6.1.2 Backend API Testing 22
6.1.3 Frontend Testing 22
6.2 Test Cases 23
Academic References **26**
Technological References **27**

Abstract

Crop loss caused by pests, diseases, and environmental stress remains a critical challenge in global agriculture, threatening both food security and farmer livelihoods. **Smart agriculture technologies**, integrating artificial intelligence (AI) and Internet of Things (IoT) solutions, offer new opportunities for early detection and intervention.

Lemon trees, which are a common fruit tree especially in Israel, suffer from many diseases that easily spread between trees and cause major agricultural damages. Early detection of these diseases could prevent crop loss and help maintain healthier trees and orchards. We believe early detection using AI visual models could greatly help early detections and early treatments that would prevent further losses.

This project proposes a **lemon health monitoring system** that leverages computer vision and natural language AI to provide real-time analysis of lemon plant conditions. Using smartphone and IoT cameras, the system captures images of lemon leaves and fruits, which are processed through a pipeline combining **YOLOv11** for object detection, **EfficientNetV2** for disease classification, and a large language model for generating natural, actionable treatment recommendations. The backend is implemented in **Python** with **FastAPI**, while the frontend is built with **React** and **Tailwind CSS** to ensure accessibility and ease of use.

For our system we decided to focus on identifying several symptoms and diseases for lemon trees. We base our detection on earlier works creating a dataset for identifying **Anthracnose**, **Bacterial Blight**, **Citrus Canker**, **Curl Virus**, **Deficiency Leaf**, **Dry Leaf**, **Healthy Leaf**, **Sooty Mold**, and **Spider Mites**. After identification, our system will provide user-friendly written advice and recommendations on what the farmers should do next in order to treat and manage their lemon orchards better.

Our goal is to achieve highly accurate and fast plant health analysis, with a user-centric design that empowers farmers regardless of their technical background. This work aims to demonstrate a scalable, modular, and practical solution for lemon disease monitoring that can be extended to other crops in the future, contributing to sustainable smart farming practices.

1. Introduction

With rising global populations and increased demand for food, the agricultural industry has been shifting in recent years to what's known as **smart agriculture** in an effort to cut costs and increase efficiency while maintaining the sustainability of the world's food supplies [1]. This method allows for the application and integration of advanced technologies such as sensors, robotics and automated systems utilising computer assistance in order to collect data, automate tasks and help farmers make informed decisions managing the farm.

Industry 4.0 principles and in particular, smart agriculture offers an opportunity to enhance global food security by improving resource availability, accessibility, and utilization. New technologies, such as Artificial Intelligence (AI), the Internet of Things (IoT), and Blockchain, have the potential to make significant contributions to the agricultural sector [2].

In lemon trees, literature shows crop losses are critical issues impacting today's global food security, it is estimated that pathogens, animals and weeds collectively contribute to crop yield loss of around 20% to 40% [3]. Lemon specific diseases such as Citrus Canker are projected to impact 50% of all lemon trees [4]. There are not many smart solutions to this problem and we believe that by combining artificial intelligence and camera sensors we could provide early detection and treatment plans and as a result, decrease the crop yield loss suffered in lemon trees.

Our proposed project will allow us to **monitor** and **provide analysis** for **lemon plants** using personal smartphone cameras and dedicated IoT cameras installed for the purpose of monitoring health in real time. The system will detect potential issues such as dehydration, deficiency, pests and diseases. Dehydration and deficiency in particular are considered a common problem leading to crop loss and good indicators for lemon health [5] even if no disease is visually identifiable such as with the disease **Mal Secco** [6] and others that can be detected early by looking at dehydration and deficiency but need further inspections to confirm. The health monitoring system will be connected to an easy to use web interface with a combination of natural language AI models ensuring accessibility even to potential users who are not adept at using advanced technological systems.

As part of our project, we plan to build a backend server capable of analyzing broad images of lemon trees, isolate the leaves from the trees, and individually diagnose deficiencies and diseases in the plant. Using a backend server will allow us to efficiently and quickly provide results to our end users and would help us solve potential problems with users using mobile devices that aren't capable of running AI models on their own.

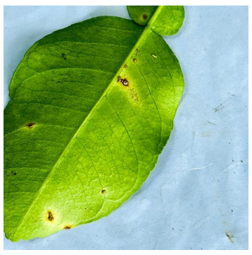
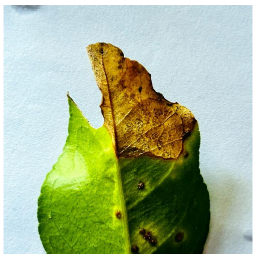
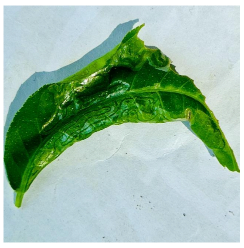
2. Literature Review

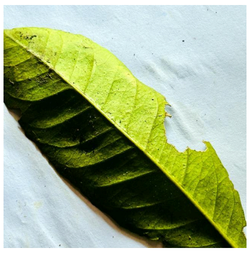
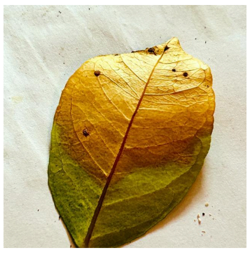
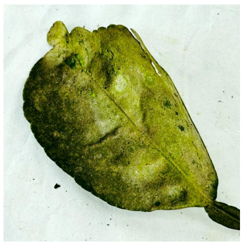
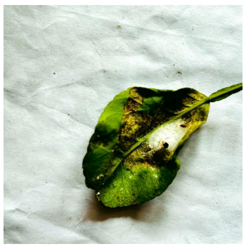
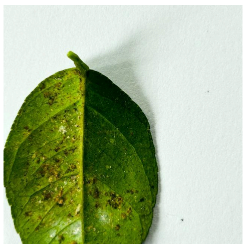
2.1 Lemon Plant Diseases

The most basic problems general to any agricultural crop and citrus plants in particular are **dehydration** and **nutritional deficiencies**, both of which can be identified visually through the leaves. Dehydration in particular can lead to a crop loss of between 10% and 20% depending on the water stress the plants go through. [7]

Given proper watering and nutritional care, diseases both **bacterial** and **fungal** [8] pose major obstacles for efficient crop production, some are visually identifiable such as **Sooty Mold Fungi** [9] and **Bacterial Blight** [10]. Other diseases are only externally identifiable with symptoms of dehydration and deficiencies that are not consistent with lack of watering or soil fertilization.

For this project we decided to focus mainly on the following diseases:

Disease	**Description**	**Visualization**
Anthracnose	Anthracnose primarily colonizes damaged and aging plant tissue. It thrives on dead wood in the canopy, spreading over short distances via heavy dew, rain splashes, or overhead irrigation. Typical symptoms include circular, flattened spots that are light tan in color, often bordered by a distinct purple edge. [10]	
Citrus Canker	The bacteria Xanthomonas is the cause of the illness known as citrus canker, which affects citrus species. Lesions on the leaves, stems, and fruit of citrus trees, such as grapefruit, oranges, and limes, are caused by infection. A fruit afflicted with canker is safe to eat but too ugly to be marketed since it severely reduces the vigor of citrus trees, even though it poses no threat to humans. In ideal circumstances, citrus canker mostly causes rind blemishing and leaf spots, but it can also result in fruit drop, defoliation, and shoot dieback. [4]	
Curl Virus	Citrus plants can curl their leaves due to virus infection, inadequate irrigation, and uneven temperature. Heavy leaf miner infestations, mealy bugs, aphids, and whiteflies can induce leaf curl. Viral infections and rising temperatures can also cause curling. However, container cultivation can control this issue. [11]	

Deficiency diseases	High-alkaline soil can cause deficiency diseases in citrus plants, like chlorosis, affecting iron, magnesium, and zinc absorption. Testing soil pH is crucial, and acidic fertilizers can remedy chlorosis. Nitrogen deficiency can cause yellowing leaves, and foliar sprays or Epsom salts can help. [5]	
Dry leaf (dehydration)	Dry leaf lemon refers to leaves that have dried out naturally or through a drying process. These leaves lose their moisture content but retain a subtle citrus aroma and flavor, often used in teas, culinary dishes, and herbal remedies. Dry lemon leaves are known for their mild, fragrant qualities, adding a delicate lemony note to infusions and spice blends. [5]	
Sooty Mold	Sooty mold on lemon leaves is a fungal growth that appears as a black, powdery coating on the leaf surface. It usually develops when the leaves are covered with honeydew, a sticky substance excreted by pests like aphids or whiteflies. While the mold itself doesn't directly harm the plant, it can block sunlight, interfering with photosynthesis and weakening the lemon tree over time. Proper pest control and regular leaf cleaning can help prevent sooty mold formation. [9]	
Spider Mites	Spider mite infestation affects lemon leaves, causing yellowing, speckling, and stippling. In severe cases, they produce webbing. Untreated damage can weaken the tree, reduce fruit yield, and increase susceptibility to diseases. Control measures include regular leaf washing, insecticidal soap, or natural predators. [10]	
Bacterial Blight	Bacterial blight in lemon leaves is a destructive disease caused by Xanthomonas axonopodis, causing dark, water-soaked lesions that enlarge, turn brown, and reduce fruit quality. Management involves pruning, improving air circulation, avoiding overhead irrigation, and using copper-based bactericides. [10]	
Healthy leaf	A healthy lemon leaf, vibrant green, glossy, smooth, and firm, supports photosynthesis and fruit growth.	

When it comes to other lemon tree diseases, many of them are not uniquely identifiable other than with dehydration and deficiency symptoms. For example, one common disease in Israel is the **Mal Secco** disease [6] that is not diagnosable without physically looking inside tree branches. A best case scenario would be an early detection of dehydration, followed by identifying the disease itself and then treating the orchard and fully removing infected branches and trees.

2.2 Visual Disease Identification

After going through the literature of the various diseases lemon trees have, for our project we looked at various plant diseases identification methods and past results. We plan on building up on earlier literature for our lemon health monitoring system to provide the most accurate diagnosis and treatments recommendation for our users.

Earlier works like **Harakannanavar et al.** [12] have already shown 99.09% accuracy in identifying tomato leaf disorders using common machine learning techniques such as K-means clustering and a CNN model. **Islam et al.** [13] have shown a 98.98% accuracy against the **PlantVillage** dataset which comprises a variety of plant diseases such as corn, potato, tomato and fruit plants such as apples. The comprehensive lemon leaf image dataset research from **Siam et al.** [14] that we are basing our project on has shown an accuracy 98.56% when predicting close up images of leaves. This shows that it is possible to identify in a highly accurate manner diseases in some of the most common agricultural products in the world and in particular in lemon trees. It is important to note that from these studies the main challenge is how comparably small the datasets are.

For more literature about **AI detection of lemon tree diseases**, there have been several attempts to build models that would detect diseases. Another such research is the detection of various citrus leaf diseases using a c**omputer vision model.** [15] the same research has been able to obtain **accuracy results up to 94.34%**. But, it is important to note that the same research has used lemon leaves placed on blank backgrounds and images of high quality not reflective of how a smartphone camera would capture leaves on a tree in an actual orchard.

This only further emphasises the need to **first isolate the leaves** and then run a model to detect diseases on them. We believe this is possible by first running the image through a model that will sample and provide **bounding boxes** for where the leaves are located in the image, before passing on to a disease classification CNN.

Following our literature overview of diseases, we understand the importance of not only detecting separate disease symptoms but also the importance of constant real time monitoring of plants in order to get early warning signs for diseases and problems that might not be visually detectable from the outside.

2.3 Machine learning models and evaluation

In earlier research shown in section 2.3 Plant health monitoring and disease detection models, very simple neural network models proved efficient in identifying unhealthy plants and categorizing diseases. Specifically, even small CNN models have proven to be effective, these also have the potential to run on a mobile device and could be used both online and offline.

With that, we understand the reality that an automatic camera on a robot or even stationary would have a difficult time identifying where exactly the leaves are located and therefore we have decided to utilise third party models such as **YOLOv11** [16] in order to first identify where the plants are located in the image and what type of plants are we looking at. This model can isolate leaves and features that we would like to pipeline to optimized CNNs to identify and categorize diseases. The diseases could further be pipelined into an **LLM** such as OpenAI or **Llama** [17] in order to generate a natural text explaining what the disease is, how it should be treated and offer farmers insights into what decisions they should make in regards to managing their farm.

The models we build should be evaluated according to standard evaluation metrics such as F1, accuracy, precision and recall and in comparison to **deep learning models** that wouldn't be able to run on a simple mobile device to access the accuracy and usability of our system in the field..

3. Research and Technology Review

3.1 Meetings with agricultural farm manager Keren

As part of refining our project to better address real-world agricultural needs, we met with Keren, a manager for an agricultural farm who provided invaluable practical insights into how smart agriculture and our system in particular could be optimized for field use.

After brainstorming several ideas including attaching cameras to a robotic arm, combining sensor data with weather forecast stations and building centralised dashboards (which is a part of a separate capstone project) we have decided on making a generalized solution for taking photographs of plants and assessing their health status.

In further meetings with Karen we have looked at and considered what plants in the farm we could run our health monitoring camera on and decided that lemons would be most suitable due to their universal importance in human nutrition, availability of past research and susceptibility to various easily recognizable diseases.

3.2 Data Collection

The first practical task for the lemon health monitoring camera is collection of data to train and test the system on. We have identified three datasets, one for lemon leaves, the second for lemon fruit, the third for generic leaf detection.

1. **Lemon Leaf Disease Dataset -** https://www.sciencedirect.com/science/article/pii/S235234092401206X - The largest available dataset targeting lemon leaves, includes healthy leaves, dehydrated, dry and deficient leaves and a variety of common diseases easily identifiable visually.
2. **Dataset for Classification of Citrus Diseases -** https://www.kaggle.com/datasets/jonathansilva2020/dataset-for-classification-of-citrus-diseases - Less comprehensive dataset for lemon fruits identifying black spots and citrus canker in lemon fruits.
3. **Leaf Detection Dataset -** https://www.kaggle.com/datasets/alexo98/leaf-detection - Comprehensive dataset for general detection of leaves in plants.

We plan to first use a model trained on the leaf detection dataset to identify leaves and isolate them, in order to be sent to a second model that will detect the exact symptoms the leaf is showing.

It is important to note these datasets include both healthy and unhealthy plants enabling us to first verify the plant type before pipelining to models that would detect the exact disease the plant has. In the future this could be used to branch out and provide in depth analysis of different plants for our system.

In addition to the existing datasets, we have access to the agricultural farm where we could source our own dataset for more lemon fruits and leaves. Collection of this dataset would include taking samples of leaves and photographing leaves separately on a blank background.

Further in the research phase of our project we will cover data preprocessing in order to augment the datasets to fit a unified form for ease of use and increased efficiency of the training and testing phases of the models that we would use.

3.3 Backend Technologies

3.3.1 Python

Python [18] is a high-level, interpreted programming language. It supports multiple programming paradigms, including object-oriented, functional, and procedural styles, and comes with a vast standard library that simplifies tasks like data processing, web development, automation, and scientific computing. Python will serve as the main programming language for the backend. It will be used to implement the machine learning pipeline, including training and deploying YOLOv11 [16] for object detection and EfficientNetV2 for disease classification.

3.3.2 FastAPI

FastAPI [20] is a modern, high-performance web framework for building **APIs with Python**, designed to be easy to use while providing production-ready speed and features. FastAPI will act as the gateway between the frontend and backend. It will expose endpoints for uploading images, triggering the pipeline, retrieving diagnostic results, and accessing historical records.

3.3.3 Google Auth

Google Auth [21] is an **authentication service** provided by Google that allows users to securely sign in to applications using their Google account credentials. It simplifies user management by handling secure login, token generation, and session management while supporting industry standards like OAuth 2.0. Google Auth will be integrated into the frontend and backend to allow users to log in securely with their existing Google accounts, and it will be used to authenticate the user session between each action.

3.4 Frontend Technologies

3.4.1 React

React [22] is a popular open-source **JavaScript** library developed by Meta for building dynamic and interactive user interfaces. It uses a component-based architecture that allows developers to create reusable UI elements and efficiently manage state, while its virtual DOM ensures fast rendering and updates. React will be used to develop the web interface through which the users interact with the system.

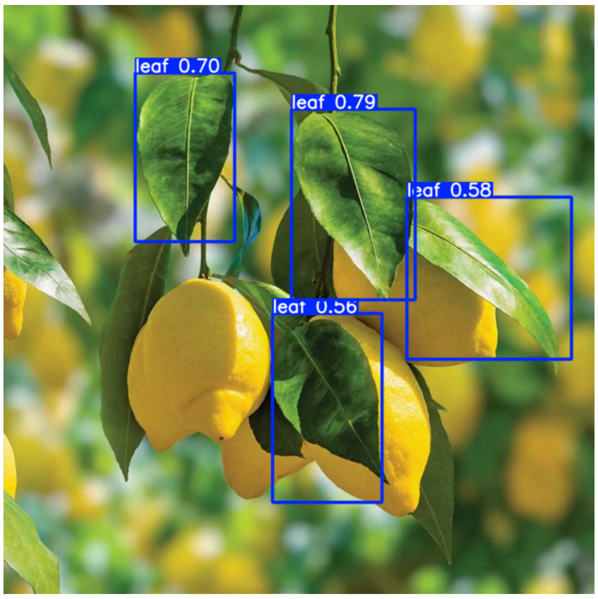
3.4.2 Tailwind CSS

Tailwind CSS [23] is a utility-first CSS framework that provides a wide range of low-level classes to build custom, responsive user interfaces without writing traditional CSS. Instead of relying on predesigned components, it enables developers to compose styles directly in their markup, giving full design flexibility while maintaining consistency and speed. Tailwind CSS will be used to style the frontend, ensuring that the interface is clean, modern, responsive and user-friendly.

3.5 Machine Learning Technologies

3.5.1 YOLOv11

YOLOv11 [16] is a **computer vision model**, widely regarded as “state of the art” in the field. designed as a fast and efficient single-stage model that directly predicts bounding boxes and class labels in one pass. Our system uses YOLOv11 for the initial sampling of leaves to scan for diseases. In our preliminary testing the model has a tendency to identify fruits (possibly because of apples images in the dataset) and as such we would need to train it to differentiate from lemon fruits and lemon leaves, and then sample only leaves for diseases.



Example sampling of leaves using YOLOv11.

3.5.2 EfficientNetV2

EfficientNetV2 [19] is a family of **convolutional neural networks** developed by Google that improves upon the original EfficientNet by offering faster training, better parameter efficiency, and higher accuracy on image classification tasks. Designed to be both lightweight and powerful, EfficientNetV2 is well-suited for a wide range of applications, from deployment on mobile and edge devices to large-scale image recognition in the cloud. EfficientNetV2 will classify the detected leaves and fruits into categories such as healthy, dehydrated, deficient, or diseased.

3.5.3 OpenAI API

The **OpenAI API** is a cloud-based service that provides access to advanced language and multimodal models, enabling developers to **integrate natural language** understanding, generation, and reasoning into their applications. It supports a wide range of tasks such as conversation, text summarization, code generation, and data analysis, with additional capabilities for working with images and other modalities. The OpenAI API will generate natural-language feedback for users based on classification results. Instead of simply reporting “leaf dehydrated,” the system will provide actionable recommendations including what further information could help diagnose an exact disease, and what should be done to begin treating the plants.

3.6 Algorithm

When our system receives a photo it will run it through an analysis that includes three phases.

The first step is sending the photo through the pretrained **YOLO** model that will create bounding boxes over the individual leaves detected.

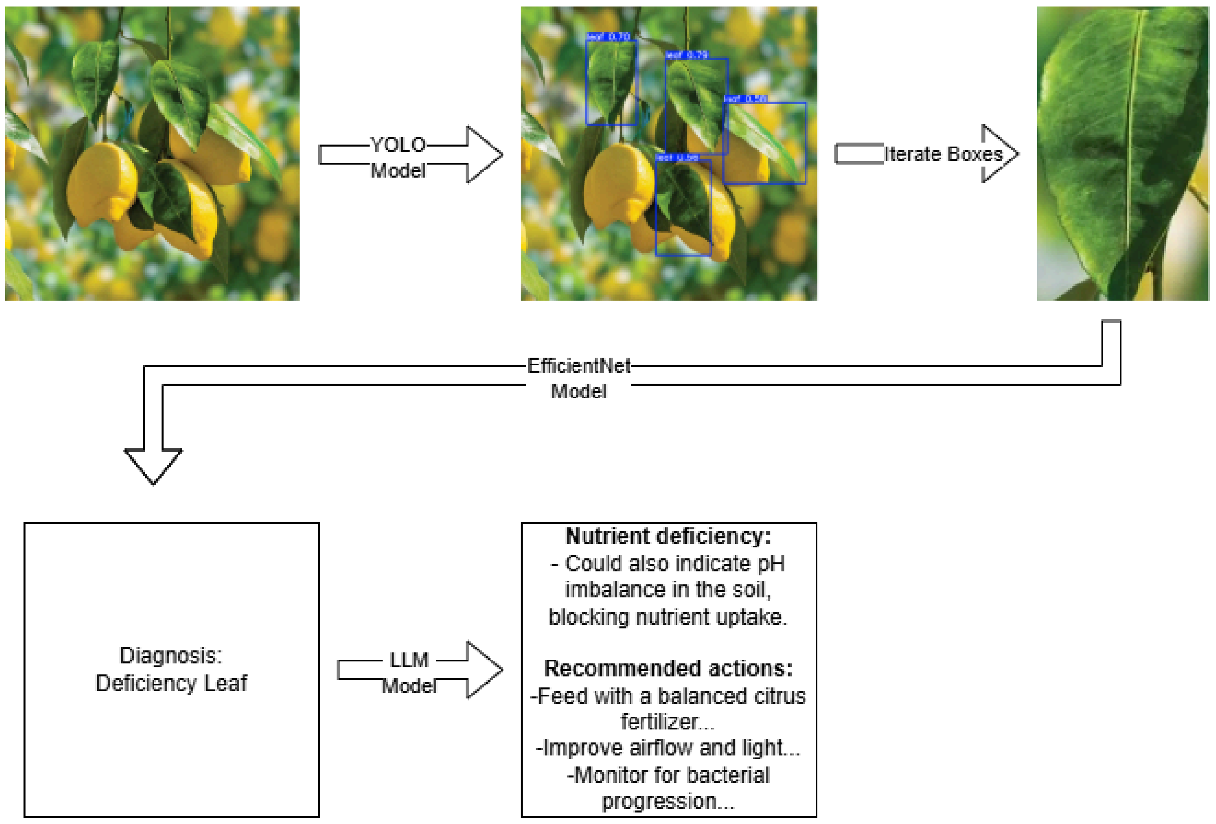
Next, we iterate over each bounding box and run it through the pretrained **EfficientNet** CNN model to identify the exact visual symptoms. The full list of symptoms that we will identify are: Anthracnose, Bacterial Blight, Citrus Canker, Curl Virus, Deficiency Leaf, Dry Leaf, Healthy Leaf, Sooty Mould, and Spider Mites. Which we append to a list of diagnoses identified for the tree.

In the final step, we take our list of diagnosed symptoms and build them into a combined LLM prompt to get written step by step recommendations and treatments to send to our end user.

```
result = yolomodel.predict('image.jpg')

for box in result.boxes:
     diagnosis += efficientnetmodel.predict(box)

prompt = createPrompt(diagnosis)
recommendationsForUser = sendPrompt(prompt)
```



Nutrient deficiency:
- Could also indicate pH imbalance in the soil, blocking nutrient uptake.

Recommended actions:
-Feed with a balanced citrus fertilizer...
-Improve airflow and light...
-Monitor for bacterial progression...

4. System Design

1. **Identification and Diagnosis Pipeline**
    a. Computer Vision Model such as **YOLOv11** to get bounding boxes for leaves and fruits of lemon trees. The bound image will be sent down a pipeline to the next part. As part of the training for leaves and fruits, we could also include identification of various common pests and harmful bugs.
    b. CNN model such as **EfficientNetv2** or **MobileNet** [24] to train on datasets and categorize leaves and fruits to symptoms, dry leaves, dehydrating leaves, deficient leaves.
    c. Large language model such as OpenAI or llama to be trained on lemon tree diagnosis and provide the user with simple actionable plans to treat their orchard. The LLM should **keep in context previous analysis** to account for users who are already properly watering and fertilizing to find undetectable diseases.

2. **Python backend server based on FastAPI**
    a. On demand run the Identification and Diagnosis Model to get recommendation and actionable plants for orchards.
    b. Real time monitoring running the model without the user input with IoT cameras according to pre configured intervals.
    c. Connect to IoT cameras from **Home Assistant** and mobile device cameras for picture inputs.
    d. View and remember past diagnosis and recommendations in LLM context.
    e. Exposed API to allow for connection to other smart agriculture dashboards
    f. Account management using **Google Auth.**

3. **Frontend User Interface based on React**
    a. User Centered Design based on simplicity.
    b. Accessibility to users who are not used to computers and mobile apps.
    c. Access to historical diagnosis and results from real time monitoring.
    d. Diagnosis functions using both smartphone cameras and IoT cameras on demand, and on regular intervals.
    e. Viewing past records function and model analysis on trends to get cues on diseases that aren’t visually identifiable. (For example, consistent dehydration despite proper watering)
    f. Configure monitoring settings such as intervals and LLM context, and interface settings such as language.

4. **Data Storage**
    a. Object store tools like (Firebase object store , amazon S3 or Minio) - store images of lemons.
    b. Relational DB - store user accounts and settings, orchard settings, previous diagnosis logs.

4.1 Requirements

5.0 FR Requirements

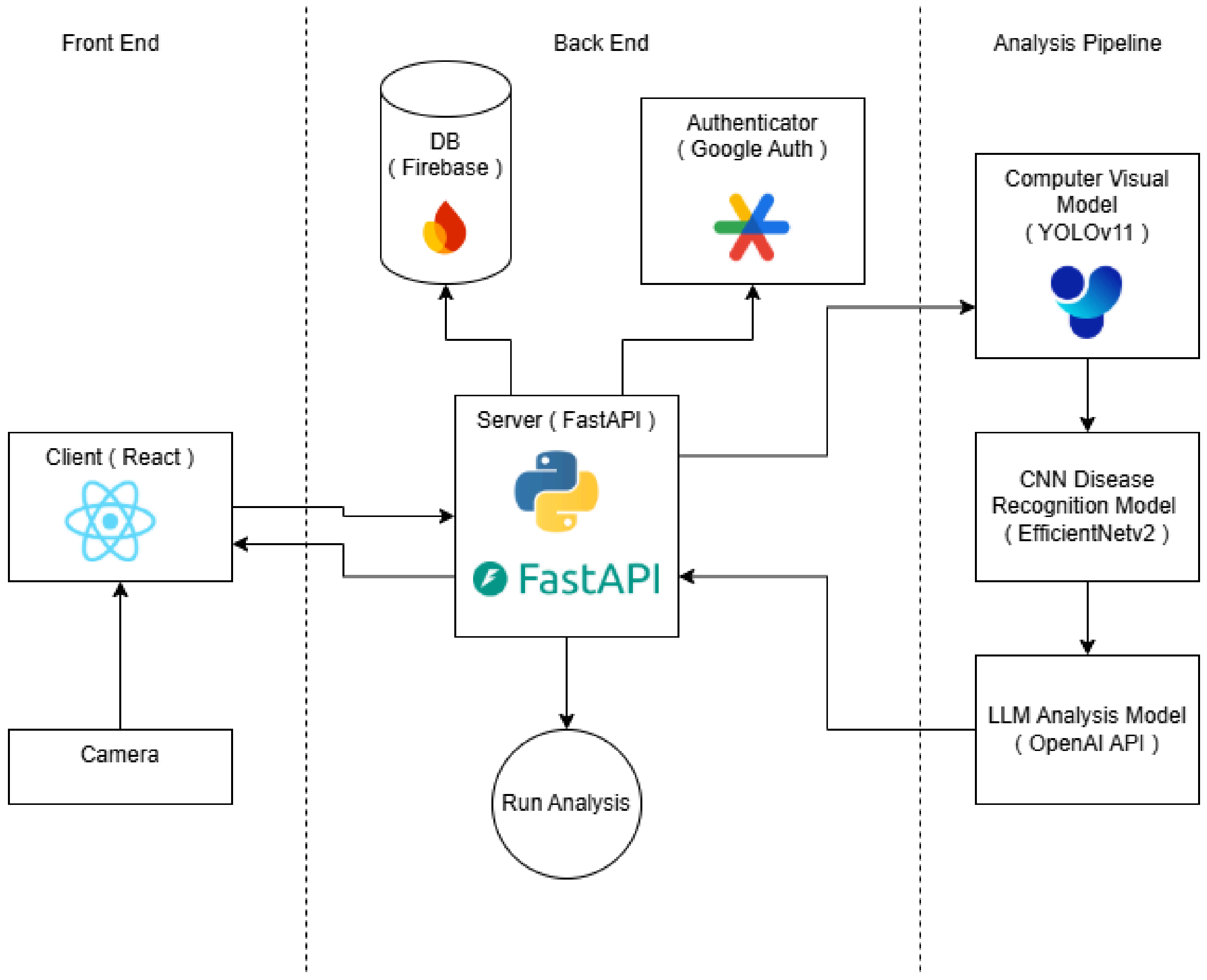
No.	FR Requirement	Description
1	Connect and get image data from smartphone cameras	*The system will allow the user to send smartphone pictures to the identification pipeline*
2	Connect and get image data from IoT connected cameras	*The system will connect and get image data from external IoT cameras using Home Assistant*
3	Upload image for analysis	*The system will allow the user to upload images from their device for analysis*
4	Run identification pipeline on image data	*The system will run and get results from the identification pipeline on image data collected*
5	Provide identification pipeline results	*The system will show users the results of the identification pipeline*
6	Run interval monitoring	*The system will capture image data and run models without user input on pre-determined intervals*
7	Save past results	*The system will save past results from the pipeline to be available for future analysis and viewing*
8	Create and connect accounts	*The system will allow users to create and connect their accounts to the system*
9	Exposed API	
10	Show and change user settings	*The system will allow to show and change user settings including language*
11	Show past results	*The system will allow to fetch data from past results of the user and display them*

The system will allow all its function to be run through an external API for connection to external dashboards

5.0 NFR Requirements

No.	NFR Requirement	Description
1	Model Performance	*The pipeline including all models should finish running within 5 seconds of the user submitting new image data*
2	Model Accuracy	*The pipeline and models shall provide accuracy scores of at least 90% in identifying diseases and symptoms.*
3	Model Scalability	*The pipeline models shall be scalable to more plants and diseases in the future.*
4	Interface Performance	*The user web interface should load and display within 5 seconds on modern smartphones and computers.*
5	Interface Usability	*The user web interface should be accessible even to none computer literate users, SUS score of at least 8*
6	Interface Configurability	*The user shall be able to decide on flexible monitoring intervals and run on demand analysis for image data.*
7	Interface Responsiveness	*The interface shall run on both common tall smartphone displays and common wide computer displays.*
8	Device Compatibility	*The interface shall be accessible and compatible with any device capable of running a full chrome/firefox/edge web browser*
9	Code Documentation	*The system code will be fully documented for ease of maintaining and use*
10	Code modularity and maintainability	*The system code will be modular and maintainable to add additional models and plants to the identification pipeline and the user interface*

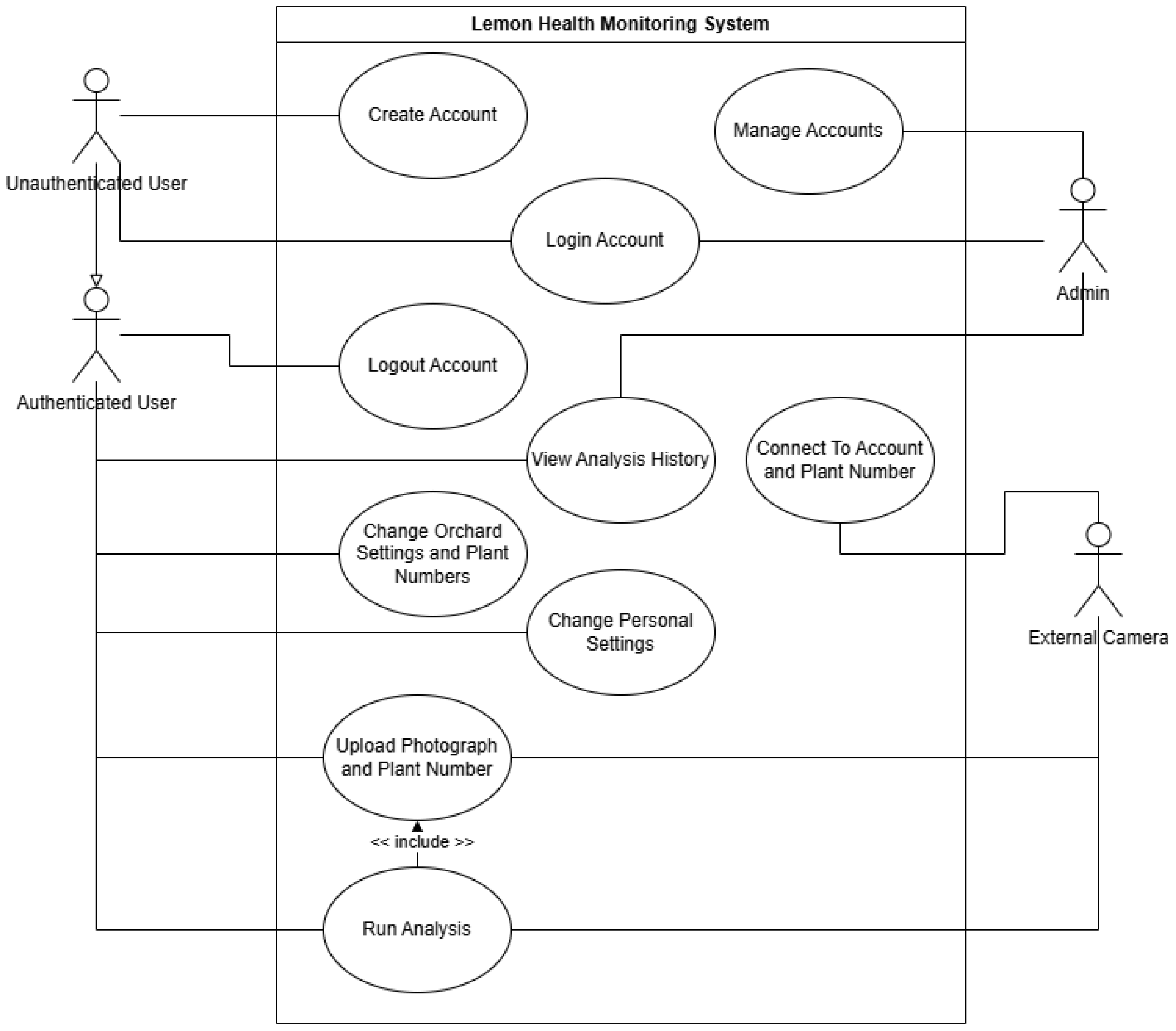
4.2 Architecture Diagram



Front End
Back End
Analysis Pipeline
DB (Firebase)
Authenticator (Google Auth)
Computer Visual Model (YOLOv11)
Server (FastAPI)
FastAPI
Client (React)
CNN Disease Recognition Model (EfficientNetv2)
Camera
LLM Analysis Model (OpenAI API)
Run Analysis

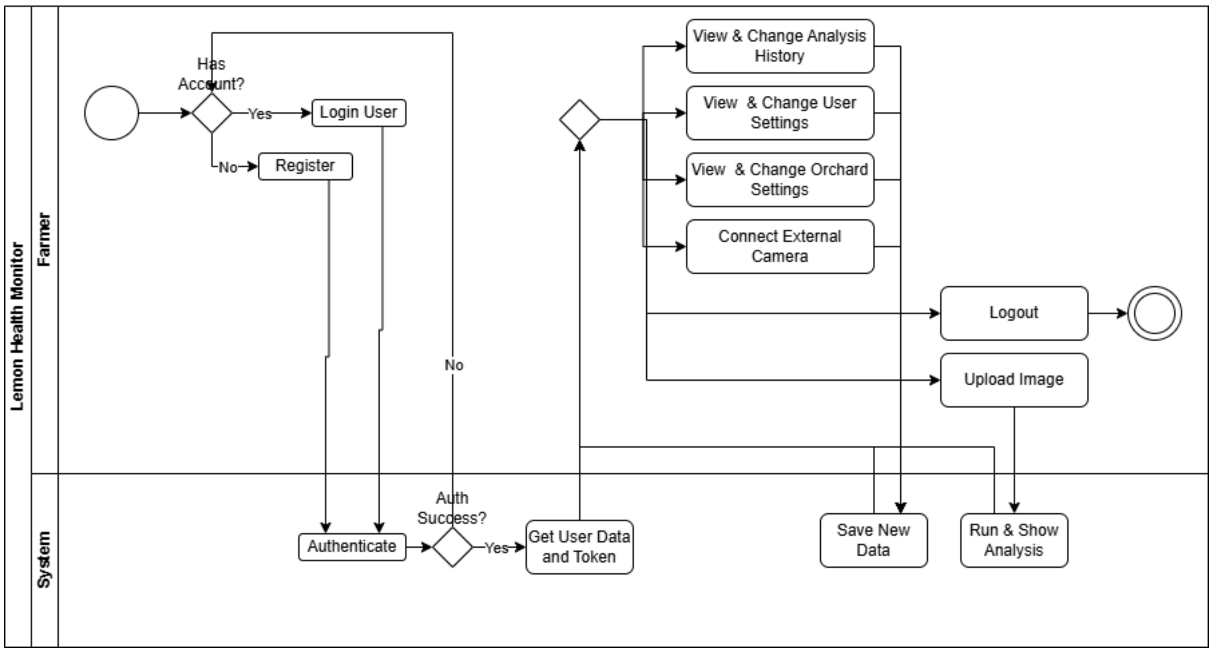
Frontend - Backend architecture design with the analysis pipeline highlighted as a separate section to show a brief of the technologies used.

4.3 User Diagram



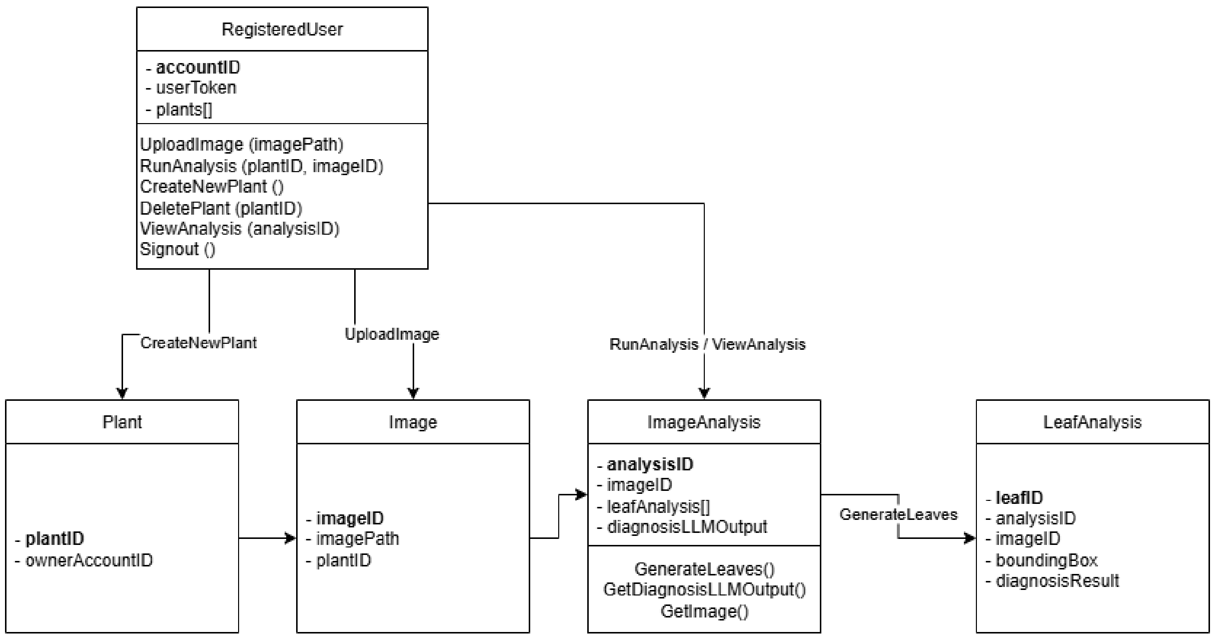
Use case diagram including from top to bottom: account management, personal and orchard settings, uploading of photographs and running analysis, and connecting external cameras through API.

4.4 Activity Diagram



Activity diagram showing user authentication, basic user operations and running an analysis on an uploaded image through the algorithm.

4.5 Objects Diagram



The most important entity of the system that facilitates the core functionality is the ImageAnalysis that will contain the image analysed and a list of all the leaves found by the YOLOv11 model and their individual leaf disease detection model result.

The user will be able to create and remove specific plants by number and upload images to the plant directly through the web interface, or upload automatically through the API.

5. Expected Achievements

a) **Accurate plant analysis -** As a main goal for the project we expect to provide accurate (>90%) analysis of the lemon plant, whether it is simply dehydrated or deficient or whether it has underlying diseases to take care of.

b) **Performative plant analysis -** We aim to deliver results in **under 5 seconds** per image. This will show how efficient our pipeline is and provide the user immediate feedback on what they should do next.

c) **Modular and scalable system -** Both in the sense that the system should be able to support **100 + concurrent users** without performance loss. enable **easy expansion** to include **other crops** (at least 2 more corps).

d) **User centric system -** We aim to make our system as simple as possible for the end user to allow even users without previous technological background to simply scan their plants and get feedback on its status and recommendation on what they should do next. We aim to achieve **SUS usability score ≥ 80 and mobile and desktop accessibility.**

5.1 Challenges

a) **Getting high quality images -** In order to get accurate results it is required the user gets high quality photographs of their lemon trees, as such it would be a challenge to guide the user on getting quality images that would provide the most accurate results.

b) **Training models to get results even on lower quality images -** As a secondary challenge, we would like to make our models work on even lower quality images

5.2 Evaluation

Due to the inherent focus on **artificial intelligence** and machine learning in this project, on the algorithm back-end, we can take standard machine learning metrics to test the viability of our system, this includes **accuracy**, **precision**, **recall** and **F1 scores**.

On the user front-end, we will evaluate the user system usability using a **SUS questionnaire** to assess how accessible the system is to our customer base, mainly farmers and agricultural workers.

A results based oriented metric such as crop yield could potentially be used to assess our system but would require long term usage and data collection on yearly crop yield, before and after implementing our system, and would be susceptible to external effects that may increase productivity or decrease regardless of our system's implementation.

6. Testing

6.1 Testing Approach

Our testing approach will combine both technical validation of the machine learning models and practical validation of the complete system.

6.1.1 Model Testing

Each model in the pipeline (YOLOv11 for detection, EfficientNetV2 for classification) will be evaluated on labeled datasets using standard performance metrics (**accuracy**, **precision**, **recall**, **F1-score**). Stress tests will be performed with varying image qualities (e.g., low resolution, poor lighting) to assess robustness.

6.1.2 Backend API Testing

Unit and integration tests will verify that the FastAPI backend correctly handles image input, executes the analysis pipeline, and returns the correct results. The automated tests will ensure the API remains stable as features expand.

6.1.3 Frontend Testing

Usability and responsiveness of the web interface will be tested across devices. User feedback will provide us with both textual assessments of how using the interface felt to the user and a quantitative metric using SUS to grade our frontend effectiveness as a user-centric design.

6.2 Test Cases

Test	Precondition	Expected Result	Description
#1 UserAuthenticationNewSignup	User has not previous account	Account is created successfully and logged in	1. The user will attempt to authenticate with the Google Auth system 2. The system receives the user token from Google Auth 3. The system successfully creates a new user 4. The system successfully logs the user in and redirects him
#2 UserAuthenticationLogin	User has an account	Account is logged in successfully	1. The user will attempt to authenticate with the Google Auth system 2. The system will find the existing user account and log in, redirecting him
#3 CreateNewPlant	User has an account and is logged into his account	New plant number is successfully created within the system	1. The user will create a new plant number according to how they have arranged their orchard 2. The system will successfully create the plant entity and inform the user of success
#4 DeletePlant	User has an account and is logged into his account, the account has a plant number created	The specified plant is deleted from the system	1. The user will choose which plant to remove 2. The system will successfully remove the plant

#5 Upload Image	User has an account and is logged into his account, the account has a plant number created	The image is successfully uploaded to the system	1. The user will choose the plant number they are taking an image for 2. The user will upload the image from their device 3. The system will successfully save the image and pass on to the analysis pipeline as specified in the architecture
#6 LeafDetection	The leaf detection model is able to run predictions. There is an image of a lemon tree to run prediction on	The leaf detection model has run prediction on an image and returned an accurate list of bounding boxes containing leaves in the image	1. The image will be ran through the leaf detection model 2. The model will return accurate bounding boxes for leaves in the image
#7 DiseaseDetection	The disease detection model is able to run predictions. The leaf detection model has returned bounding boxes for an image	The disease detection model will run on all bounding boxes of leaves and will return accurate results for their diagnosis	1. The image and bounding boxes will be sent to the disease detection run function 2. The model will return an accurate list of results for each bounding box
#8 RecommendationGeneration	The recommendations LLM is able to generate. The disease detection model has run an analysis and returned accurate results.	The recommendations LLM will return in readable text the following: 1. What is the diagnosis of each leaf? 2. How can we verify the diagnosis and what further information is needed to make an accurate diagnosis 3. How to further treat the lemon plant	1. The recommendation LLM will receive the diagnosis of each bounding box from the disease detection 2. The LLM will return accurate and readable text as specified in the test result

#9 AnalysisSequences FullRun	The user has an account and is logged into his account, the account has a plant number created. The user has an image ready to upload for analysis	The system will run the steps tested in #5, #6, #7, #8 sequentially. The recommendations LLM will return in readable text similar to test #8	1. The user will choose the correct plant number and upload an image for analysis 2. The system will run the image through the leaf detection model to output bounding boxes for leaves 3. The system will run the bounding boxes through the disease detection model to output diagnosis results 4. The system will run the diagnosis results through the generative LLM to provide the user information and recommendations according to specifications in test #8
#10 ViewHistory	The user has ran previous analysis on a plant	The system will show the user the full history of the analysis he has performed on a plant	1. The user will choose a plant 2. The system will fetch and display all previous analysis

Academic References

1. Charania, I., & Li, X. (2020). Smart farming: Agriculture's shift from a labor intensive to technology native industry. Internet of Things, 9, 100142. https://doi.org/10.1016/j.iot.2019.100142
2. Hassoun, A., Mhlanga, D., Abderahman Rejeb, Bhat, Z., Buheji, M., & Bigliardi, B. (2025). The Role of Industry 4.0 in Global Food Security: A Promising Pathway to Ending Hunger. Smart Agricultural Technology, 100974–100974. https://doi.org/10.1016/j.atech.2025.100974
3. Savary, S., Ficke, A., Jean-Noël Aubertot, & Hollier, C. (2012). Crop losses due to diseases and their implications for global food production losses and food security. Food Security, 4(4), 519–537. https://doi.org/10.1007/s12571-012-0200-5
4. Ali, S., Hameed, A., Ghulam Muhae-Ud-Din, Muhammad Ikhlaq, Ashfaq, M., Muhammad Atiq, … Wang, Y. (2023). Citrus Canker: A Persistent Threat to the Worldwide Citrus Industry — An Analysis. Agronomy, 13(4), 1112–1112. https://doi.org/10.3390/agronomy13041112
5. Anand Kalatippi, & Hota, D. (2021, January 25). Physiological Disorders of Citrus. Retrieved October 15, 2025, from ResearchGate website: https://www.researchgate.net/publication/348750552_Physiological_Disorders_of_Citrus
6. Ben-Hamo, M., Ezra, D., Krasnov, H., & Blank, L. (2020). Spatial and Temporal Dynamics of Mal Secco Disease Spread in Lemon Orchards in Israel. Phytopathology®, 110(4), 863–872. https://doi.org/10.1094/phyto-06-19-0195-r
7. Iván García-Tejero, Durán-Zuazo, V. H., Arriaga-Sevilla, J., & José Luis Muriel-Fernández. (2011). Impact of water stress on citrus yield. Agronomy for Sustainable Development, 32(3), 651–659. https://doi.org/10.1007/s13593-011-0060-y
8. Khanchouch, K., Pane, A., Chriki, A., & Cacciola, S. O. (2017). Major and Emerging Fungal Diseases of Citrus in the Mediterranean Region. Citrus Pathology. https://doi.org/10.5772/66943
9. Apacionado, B. V., & Ahamed, T. (2023). Sooty Mold Detection on Citrus Tree Canopy Using Deep Learning Algorithms. Sensors, 23(20), 8519. https://doi.org/10.3390/s23208519
10. Manal JAOUAD, Alieu MOININA, Said EZRARI, & Rachid LAHLALI. (2020). Key pests and diseases of citrus trees with emphasis on root rot diseases: An overview. Moroccan Journal of Agricultural Sciences, 1(3). Retrieved from https://www.agrimaroc.com/index.php/MJAS/article/view/851
11. Li, X., Wei, Z., Peng, F., Liu, J., & Han, G. (2022). Estimating the distribution of chlorophyll content in CYVCV infected lemon leaf using hyperspectral imaging. Computers and Electronics in Agriculture, 198, 107036–107036. https://doi.org/10.1016/j.compag.2022.107036
12. Harakannanavar, S. S., Rudagi, J. M., Puranikmath, V. I., Siddiqua, A., & R Pramodhini. (2022). Plant leaf disease detection using computer vision and machine learning algorithms. Global Transitions Proceedings, 3(1), 305–310. https://doi.org/10.1016/j.gltp.2022.03.016
13. Islam, M. M., Ahad, A., Talukder, M. A., Uddin, K., Uddin, M. A., Hasan, M. K., … Debnath, S. K. (2023). DeepCrop: Deep learning-based crop disease prediction with web application. Journal of Agriculture and Food Research, 14, 100764–100764. https://doi.org/10.1016/j.jafr.2023.100764
14. Siam, K., Prayma Bishshash, Md Asraful Sharker Nirob, Mamun, S. B., Md Assaduzzaman, & Sheak. (2024). A Comprehensive Image Dataset for the Identification of Lemon Leaf Diseases and Computer Vision Applications. Data in Brief, 58, 111244–111244. https://doi.org/10.1016/j.dib.2024.111244
15. Aktaruzzaman Pramanik, Khan, A. Z., Biswas, A. A., & Rahman, M. (2021). Lemon Leaf Disease Classification Using CNN-based Architectures with Transfer Learning. 2022 13th International Conference on Computing Communication and Networking Technologies (ICCCNT), 1, 1–6. https://doi.org/10.1109/icccnt51525.2021.9579586

Technological References

16. YOLOv11. (2025). Retrieved October 15, 2025, from Yolov11.com website: https://yolov11.com/
17. Industry Leading, Open-Source AI | Llama. (2022). Retrieved October 15, 2025, from Industry Leading, Open-Source AI | Llama website: https://www.llama.com/
18. Welcome to Python.org. (2025, October 15). Retrieved October 15, 2025, from Python.org website: https://www.python.org/
19. Tan, M., & Le, Q. (2021). EfficientNetV2: Smaller Models and Faster Training. Retrieved from https://arxiv.org/pdf/2104.00298
20. FastAPI. (2025). Retrieved October 15, 2025, from Tiangolo.com website: https://fastapi.tiangolo.com/
21. Using OAuth 2.0 to Access Google APIs. (2025). Retrieved October 15, 2025, from Google for Developers website: https://developers.google.com/identity/protocols/oauth2
22. React. (2015). Retrieved October 15, 2025, from React.dev website: https://react.dev/
23. Tailwind CSS - Rapidly build modern websites without ever leaving your HTML. (2025). Retrieved October 15, 2025, from Tailwindcss.com website: https://tailwindcss.com/
24. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., … Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. Retrieved October 15, 2025, from arXiv.org website: https://arxiv.org/abs/1704.04861