Individual Project Report

Enhancing Plant Health with AI

Using Thermal and Image Processing

By: Alan Palayil

Lab Partner: Zahra Tolideh and Lukas Klicker

Instructor: Dr. Jafar Saniie

ECE 501 (Summer 2023)

Due Date: 06/22/2023

Acknowledgment: I acknowledge all the work (including figures and codes) belongs to me and/or the people who are referenced.

Electronic Signature: Alan Palayil A20447935 (Due Date: 6/22/2023)

Table of Contents

Abstract:

Plant diseases pose a significant threat to global food security and agricultural productivity. Early detection and accurate classification of plant diseases are crucial for timely intervention and effective management. In this paper, we propose an integrated plant disease detection and classification system that combines deep learning techniques with environmental sensors to provide comprehensive and automated disease assessment. The system incorporates thermal imaging technology to capture leaf temperature, enabling the calculation of Vapor Pressure Deficit (VPD). Additionally, a DHT10 sensor measures the environmental humidity, providing essential data for disease risk evaluation. By integrating these environmental parameters, the system offers an integrated approach to disease detection and management.

We utilize a deep learning framework based on convolutional neural networks (CNN) to classify plant diseases accurately. The CNN model is trained and fine-tuned using a diverse dataset of plant images, including visible and thermal images. The USB camera is employed to extract frames from the video feed, enabling real-time testing of the AI model for disease detection. The trained model demonstrates high accuracy in identifying and classifying various plant diseases. It can effectively distinguish between healthy and diseased plant conditions, aiding in the early identification of potential risks. Moreover, the system utilizes VPD calculations to assess the need for irrigation, providing insights into plant water requirements.

The proposed system offers several benefits, including real-time disease detection, non-invasive monitoring, and scalability. By combining deep learning techniques with environmental sensors, it enables efficient disease management practices and reduces reliance on manual inspection. The system can be deployed in diverse agricultural settings, such as farms, greenhouses, and nurseries, assisting farmers, agronomists, and researchers in mitigating plant diseases and ensuring sustainable crop production.

Keywords: Plant disease detection, deep learning, convolutional neural networks, environmental sensors, thermal imaging, Vapor Pressure Deficit (VPD), disease risk assessment, automated system.

Introduction:

The project focuses on developing an integrated system for plant disease detection and classification using deep learning techniques and environmental sensors. The goal is to enable early detection and accurate classification of plant diseases, facilitating timely intervention and effective disease management. The system utilizes thermal imaging technology to capture leaf temperature, which is crucial for calculating Vapor Pressure Deficit (VPD). VPD, along with data from an AHT10 sensor measuring environmental humidity, is used to assess disease risk and determine the need for irrigation. By incorporating these environmental parameters, the system provides a comprehensive approach to disease detection and management.

A deep learning framework based on convolutional neural networks (CNN) is employed for accurate disease classification. The CNN model is trained and fine-tuned using a diverse dataset of plant images, including both visible and thermal images. Real-time testing of the AI model is facilitated using an USB camera that extracts frames from the video feed. The project's technical aspects include developing algorithms to process thermal images, extract relevant features, and train the CNN model for disease classification. Integration with environmental sensors involves acquiring and processing data from the AHT10 sensor to assess disease risk based on VPD calculations. Additionally, the system incorporates real-time video processing and testing using the USB camera to validate the AI model's performance.

The project aims to provide a comprehensive and automated solution for plant disease detection and classification. By combining deep learning techniques with environmental sensors, the system offers real-time disease monitoring, enables proactive disease management, and reduces reliance on manual inspection. The project's technical components encompass image processing, machine learning, sensor integration, and real-time video analysis, contributing to a robust and scalable solution for plant disease management.

Hardware Components:

The hardware setup for the project comprises several key components, including the Jetson Nano single-board computer, thermal camera, AHT10 sensor, and a USB camera. Each component plays a vital role in the overall functionality of the system.

1. Jetson Nano: The Jetson Nano is a power-efficient and high-performance single-board computer developed by NVIDIA specifically designed for AI applications. It provides high-performance computing capabilities for running complex deep learning algorithms. The Jetson Nano is equipped with a GPU, enabling accelerated AI and deep learning

computations. It provides the necessary computational power for processing and analyzing the data from the connected sensors and camera.

2. Thermographic Camera: The thermal camera plays a pivotal role in our project as it enables the capture of thermographic images of plant leaves. These images are essential for calculating the Leaf's Vapor Pressure Deficit (VPD) value, which is a key indicator of a plant's water stress level. For this purpose, we have chosen Teledyne FLIR's Lepton 3.5, known for its high resolution and advanced calibration settings. With five times the resolution of the MLX90640, it provides detailed temperature measurements for each pixel. Additionally, the usage of Long Wave Infrared Red further enhances the accuracy of temperature measurements.

   To facilitate communication with the thermal camera, we will utilize the Group Gets PureThermal 2 Breakout board. This breakout board enables USB communication, allowing seamless integration with our system. By incorporating the thermal camera as an essential sensor, we can capture thermal images of plant leaves, analyze temperature data, and calculate the VPD. These capabilities provide valuable insights into plant health, enabling us to detect potential diseases or anomalies and make informed decisions regarding crop management.

3. AHT10 Sensor: The AHT10 sensor is a crucial component of our integrated system as it measures both temperature and humidity levels in the environment. Humidity plays a vital role in evaluating the risk of diseases in plants, as specific humidity conditions favor the growth of certain plant diseases. This sensor provides accurate humidity readings, which are essential for calculating the Vapor Pressure Deficit (VPD).

   By combining the humidity data from the AHT10 sensor with the temperature data captured by the thermal camera, our system can accurately determine the VPD. This information is invaluable for evaluating a plant's irrigation requirements. Additionally, the ambient air temperature measured by the AHT10 sensor can be utilized to enhance the calibration threshold of the thermal imaging camera, ensuring precise temperature measurements for thermal imaging analysis.

   The integration of the AHT10 sensor enables us to capture comprehensive environmental data and derive meaningful insights for plant health monitoring and disease prevention. By leveraging the power of temperature and humidity measurements, our system provides farmers with the necessary information to optimize irrigation strategies and maintain a healthy growing environment for their plants.

4. USB Camera: The USB camera is a vital component in our hardware setup, serving multiple purposes within our system. Firstly, it is responsible for extracting frames from the video feed in real-time, enabling continuous monitoring of plant health. These frames are then passed on to our deep learning model for analysis and disease detection.

   Moreover, the USB camera plays a crucial role in testing and validating the performance of our AI model. It captures high-quality images, which are subsequently processed by the system to make predictions using the trained deep learning model. This allows us to assess the accuracy and reliability of our system in real-world conditions.

By incorporating the USB camera into our setup, we enhance the versatility and functionality of our system. Real-time monitoring and analysis of plant health has become possible, enabling proactive disease management and timely interventions. The USB camera empowers farmers and agricultural stakeholders to make informed decisions based on up-to-date and accurate information, leading to improved crop health and increased yields.

5. Laptop: The MSI GS75 Stealth 9SG laptop serves as the development and testing platform for the integrated system. Equipped with an Nvidia RTX 2080-Max-q, Intel i7-9750H processor, and 32GB of 2667 MHz RAM, this laptop provides the necessary computational power to train and optimize the deep learning model. It serves as a robust and convenient environment for coding, training, and fine-tuning the plant disease detection system.

   The laptop's processing power and ample storage capacity enable efficient model training and evaluation. It allows us to run experiments, analyze results, and make necessary adjustments to enhance the performance of our system. With this powerful hardware configuration, we can expedite the development process and ensure optimal accuracy and efficiency in detecting and classifying plant diseases.

   Using the MSI GS75 Stealth 9SG laptop as our development and testing platform, we can leverage its capabilities to achieve high-quality results and effectively optimize the plant disease detection system.

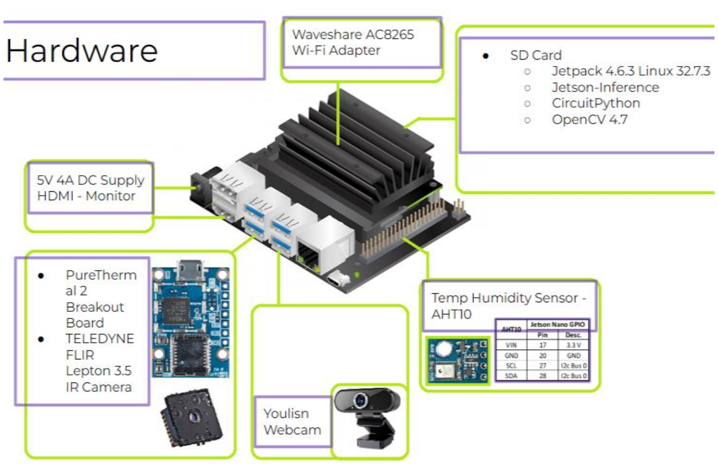


Figure 1: Hardware Layout

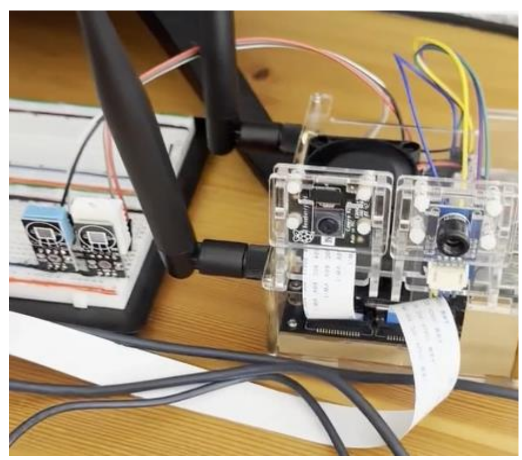


Figure 2: Hardware Setup Image

The integration of these hardware including the Jetson Nano, thermal camera, AHT10 sensor, USB camera, and laptop, creates a comprehensive and powerful system for plant disease detection and management. Each component contributes specific functionalities, enabling accurate measurements of temperature and humidity, VPD calculations, efficient image processing, and robust development and testing capabilities. The combined hardware setup empowers researchers and plant scientists to monitor and analyze plant health in a reliable and advanced manner, leveraging the capabilities of both the embedded system and the development platform.

Software Components:

The project utilizes various AI algorithms and software tools to facilitate plant disease detection and analysis. These components enable the development, training, and deployment of the deep learning model and provide essential tools for data processing and visualization. The key AI algorithms and software tools used in the project include:

1. Deep Learning Model: The core of the project's AI capabilities lies in the deep learning model. Deep learning algorithms, such as Convolutional Neural Networks (CNNs), are employed to learn and recognize patterns in the plant images. These models are trained using large datasets of labeled plant images to enable accurate classification and identification of plant diseases.
2. TensorFlow: TensorFlow is a widely used open-source deep learning framework. It provides a comprehensive ecosystem for developing and training deep neural networks. TensorFlow offers a range of tools and libraries that simplify the implementation of complex models and streamline the training process. The project utilizes TensorFlow for building and training the deep learning model for plant disease detection.
3. Keras: Keras is a high-level neural networks API written in Python. It is built on top of TensorFlow and provides a user-friendly interface for constructing deep learning models. Keras offers a wide range of pre-built layers and models, making it easy to design and train deep neural networks. The project leverages Keras for building and configuring the architecture of the deep learning model.
4. OpenCV: OpenCV (Open-Source Computer Vision Library) is a powerful open-source computer vision library. It provides a collection of algorithms and functions for image and video processing, including image filtering, feature extraction, and object detection. OpenCV is used in the project for image preprocessing tasks, such as resizing, normalization, and augmentation, to enhance the quality and diversity of the training data.
5. NumPy: NumPy is a fundamental library for numerical computing in Python. It provides powerful tools for handling large multi-dimensional arrays and matrices, along with a collection of mathematical functions for array operations. NumPy is utilized in the project for efficient data manipulation, preprocessing, and transformation tasks, enabling seamless integration with deep learning frameworks.

6. Jupyter Notebook: Jupyter Notebook is an interactive computing environment that allows for the creation and sharing of documents containing live code, equations, visualizations, and narrative text. It provides a convenient platform for developing and prototyping the AI algorithms used in the project. Jupyter Notebook enables iterative experimentation, data exploration, and collaborative development, fostering an efficient workflow.
7. Matplotlib: Matplotlib is a popular data visualization library in Python. It provides a wide range of functions for creating several types of plots and charts, enabling effective visualization of data and model performance. Matplotlib is utilized in the project for generating visualizations of training and validation metrics, such as accuracy and loss curves, aiding in the analysis and interpretation of the deep learning model's performance.

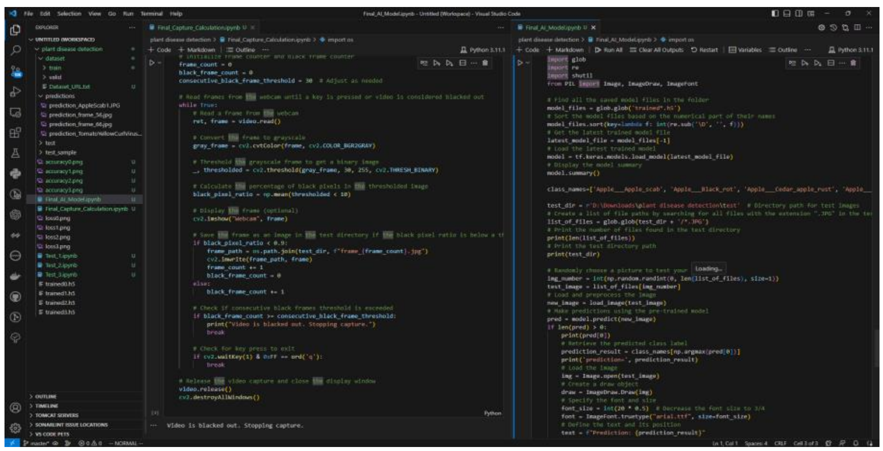


Figure 3: Jupyter Notebook files and Directory Layout

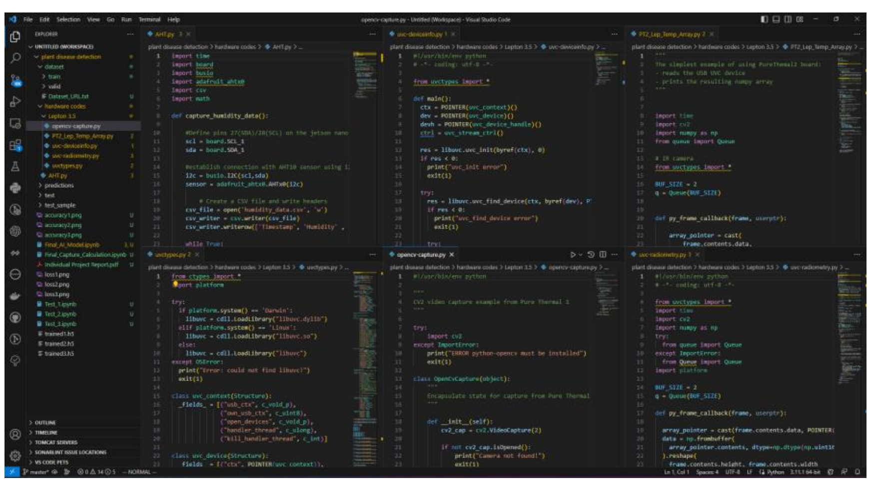


Figure 4: Hardware setup code for Jetson Nano and Directory Layout

These AI algorithms and software tools form a comprehensive toolkit for developing and deploying the plant disease detection system. They enable the construction and training of accurate deep learning models, efficient data processing and visualization, and iterative experimentation and analysis, contributing to the successful implementation of the project's objectives.

Coding Methodology:

To develop a comprehensive system for plant identification and disease detection, four main coding methodologies were employed. This section discusses the methodologies used for data extraction, video processing, and plant identification, as well as the significance of Vapor Pressure Deficit (VPD) in the context of the project.

1. Data Output from Thermal Camera and AHT10 Sensor: To gather essential data from the thermal camera and AHT10 sensor, a Data Extraction Program was developed. This program efficiently extracts data from the AHT10 sensor and thermal camera, facilitating the creation of a comprehensive database for integration with the Vapor Pressure Deficit (VPD) application.
2. Video Processing from the USB Camera: The video processing component of the system relies on the utilization of OpenCV, a popular computer vision library. OpenCV is utilized to convert the video feed from the USB camera into individual frames. To ensure optimal data storage and processing, the program selectively saves only the frames captured during periods without blackouts. These frames are then stored in a designated directory, ready for use by the AI model.
3. Plant Identification and Disease Detection: The core functionality of plant identification and disease detection is achieved through a Python program that harnesses the power of Keras, an open-source deep learning library integrated within the Tensorflow framework. The program leverages Keras to identify plants based on their distinctive leaf characteristics. Additionally, it incorporates a pre-existing dataset specifically curated for disease detection. The program further enhances its performance by utilizing a pre-trained local model that undergoes continuous testing and refinement.
4. Vapor Pressure Deficit (VPD): VPD plays a pivotal role in irrigation practices and is a crucial parameter in evaluating the drying power of the atmosphere. Monitoring and managing VPD values are essential for efficient irrigation practices, water conservation, and improved crop health and productivity.

By understanding and effectively managing VPD, the integrated system contributes to sustainable water management, optimal crop health, and improved agricultural productivity.

Results:

The results from this project are twofold. Firstly, the implementation of thermal imaging for leaf temperature measurement coupled with the AHT10 sensor for environment humidity will provide accurate data for calculating the Vapor Pressure Deficit (VPD) in plants. This will enable better monitoring and understanding of the plant's water stress levels and irrigation needs.

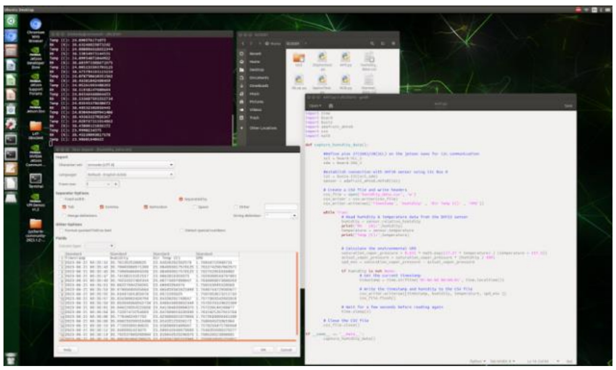


Figure 5: AHT Sensor Code and Humidity Output

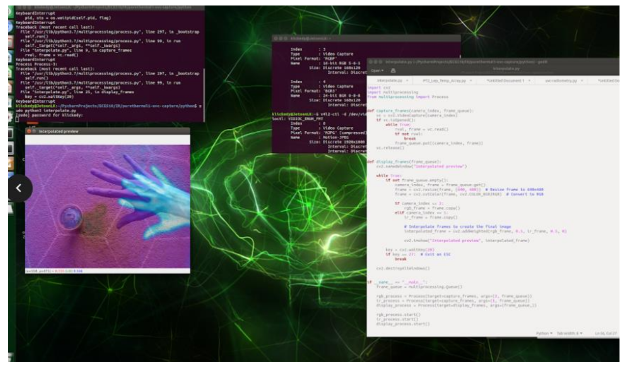


Figure 6: Interpolation

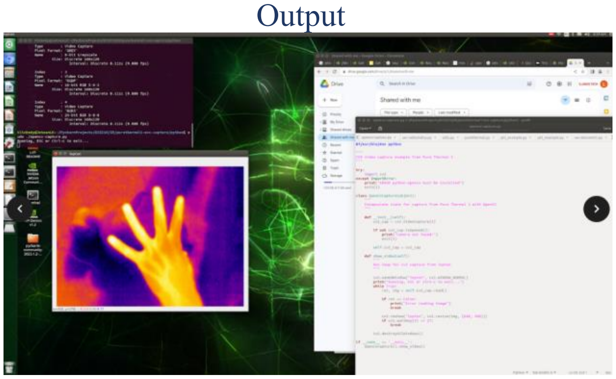


Figure 7: Lepton OpenCV Configuration

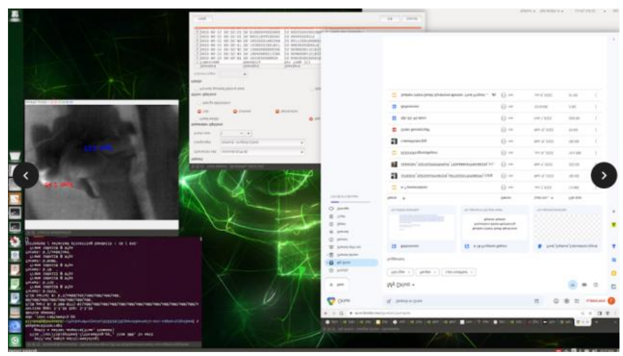


Figure 8: Lepton Radiometry Configuration

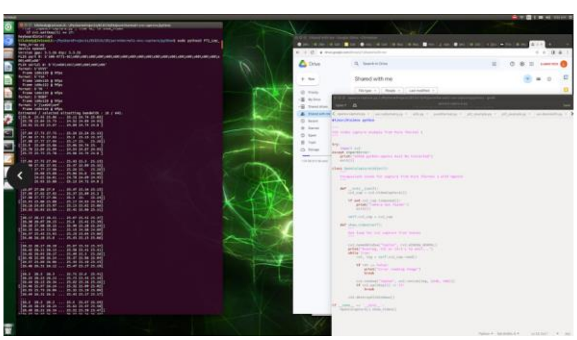


Figure 9: Lepton Temperature Array

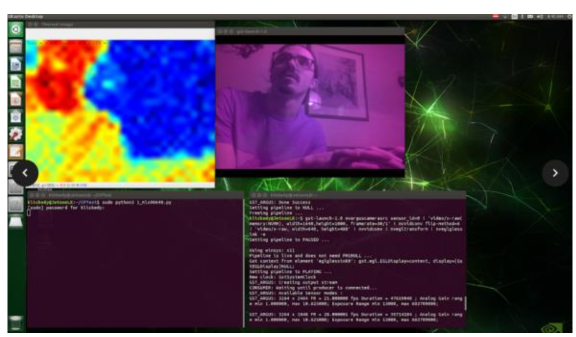


Figure 10: Thermal Camera Demonstration

The integration of the USB camera allowed for real-time image analysis and disease detection, leveraging deep learning algorithms to classify and identify plant diseases accurately.

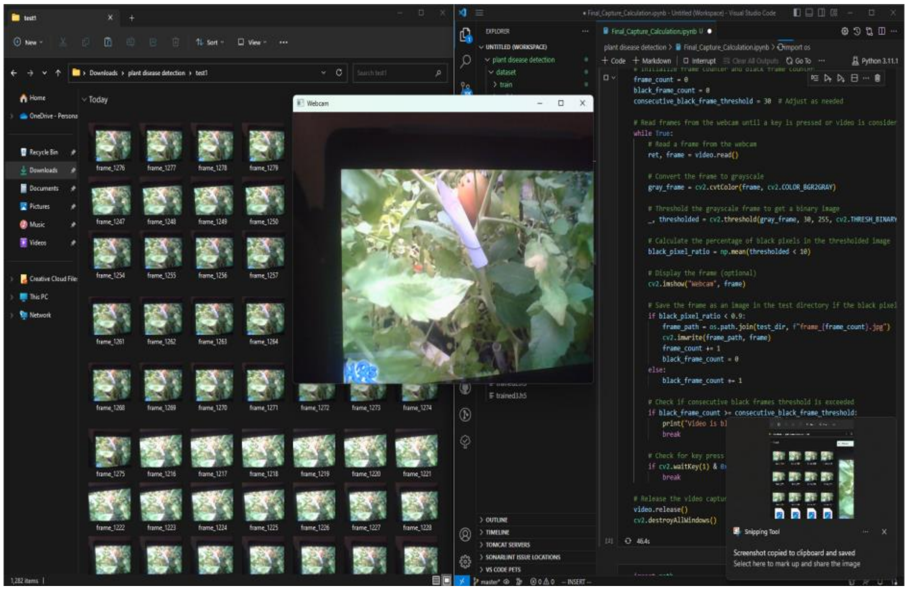
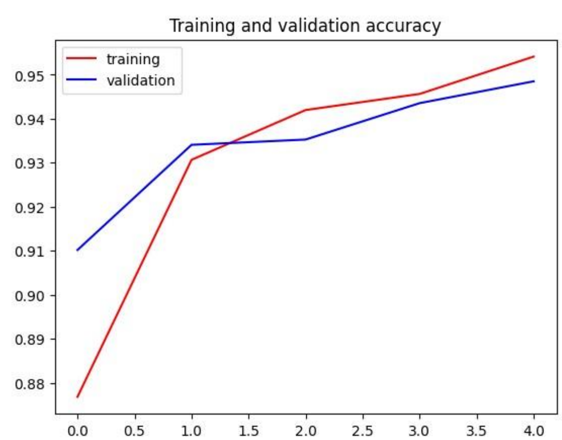
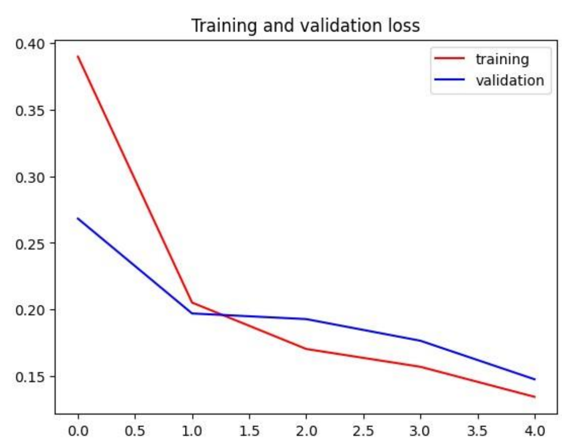


Figure 11: Webcam Image extraction Code and Directory Upload

The AI model trained on the dataset provides reliable predictions of disease presence and aid in timely intervention and disease management.

<table>
<tr><th>Accuracy</th><th>Loss</th></tr>
<tr><td>Epoch 5
</td><td>Epoch 5
</td></tr>
</table>

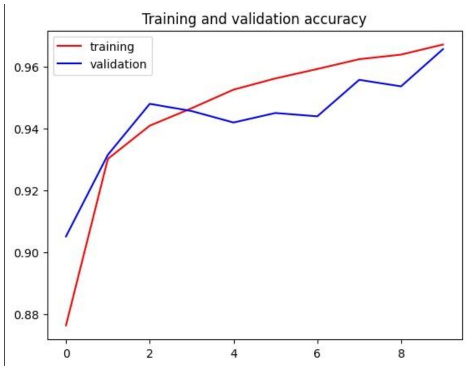
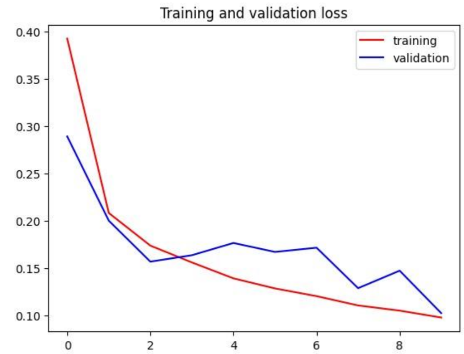
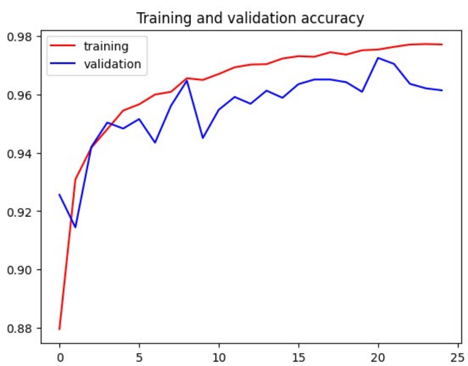
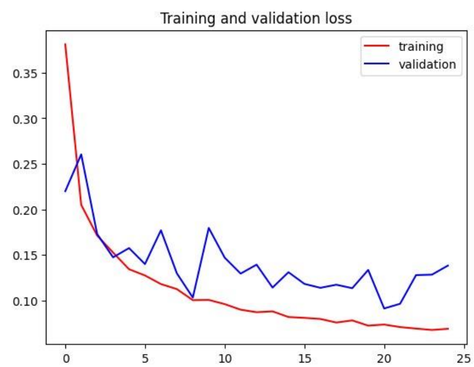
<table>
<tr><td>Epoch 10

</td><td>Epoch 10

</td></tr>
<tr><td>Epoch 25

</td><td>Epoch 25

</td></tr>
</table>

Table 1: Result plots for Local Training model with the CNN architecture

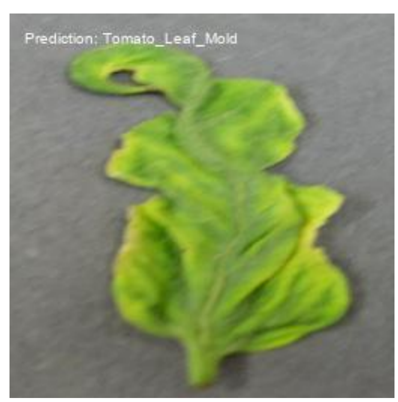
<table>
<tr><th>Original Image</th><th>Image after Analysis</th></tr>
<tr><td></td><td>
</td></tr>
</table>

Table 2: Prediction results from sample test sample

Original Image	Image after Analysis
	
	

Table 3: Prediction results for Webcam extracted frame (Tomato leaves video)

Milestones:

1. Installation and Configuration of Libraries

- Successfully installed the required libraries and dependencies for the project, ensuring compatibility and proper functionality.
- Overcome any challenges related to library installation and resolve version conflicts.

2. Dataset Acquisition and Preprocessing

- Identify and gather a diverse and comprehensive dataset of plant disease images.
- Perform necessary preprocessing steps on the dataset to ensure data quality and consistency.

3. Integration of Thermal Imaging and AHT10 Sensor

- Implement thermal imaging for leaf temperature measurement using the thermal camera.
- Integrate the AHT10 sensor to measure environmental humidity.
- Develop algorithms to calculate Vapor Pressure Deficit (VPD) using temperature and humidity data.

4. Integration of USB Camera for Real-time Analysis

- Integrate the USB camera into the system to capture real-time plant images.
- Develop image analysis algorithms using deep learning techniques to detect and classify plant diseases accurately.

5. Training and Optimization of AI Model

- Train the deep learning model using the acquired dataset to predict plant diseases.
- Optimize the model by fine-tuning architecture, adjusting hyperparameters, and implementing data augmentation techniques.
- Achieve a target accuracy of 90% or higher through iterative refinement and experimentation.

6. Results and Performance Evaluation

- Analyze and evaluate the performance of the trained AI model on test data.

- Generate result plots, including accuracy and loss curves, to assess the model's performance over different epochs.
- Validate the model's effectiveness by comparing original images with predicted disease analysis results.

Work Distribution:

The work distribution for the project was divided among team members as follows:

Name	Tasks Information
Alan Palayil	• Was responsible for curating the dataset for plant detection, ensuring it was comprehensive and representative. • Gathered a dataset specifically focused on plant diseases, encompassing a wide range of plant ailments. • Took charge of designing the AI model architecture to address both plant detection and disease classification. • Developed a program that could extract frames from the camera feed and input them into the AI model for analysis.
Zahra Tolideh	• Took on the task of creating the VPD (Vapor Pressure Deficit) calculation program, which played a crucial role in assessing the plant's irrigation needs. • Designed irrigation strategies based on predefined metrics, leveraging the VPD calculations to determine optimal watering requirements. • Developed a program that could extract data from the AHT10 sensor, which measured environmental humidity, and input it into the VPD program for analysis.
Lukas Klicker	• Was responsible for the hardware setup, ensuring all components were properly connected and functioning. • Worked on enabling thermal imaging to work simultaneously with the USB camera, synchronizing and calibrating the two sources. • Debugged the GPIO connection for the AHT10 sensor, ensuring accurate data retrieval for VPD calculations.

	• Focused on extracting data from the thermal images, developing the necessary algorithms to process and extract relevant information.

Overall, this work distribution allowed the team to leverage each member's expertise and skills in their respective areas, leading to a comprehensive and collaborative project implementation.

Challenges Faced:

During this project, several challenges were encountered, which required careful attention and problem-solving. Initially, installing the necessary libraries and dependencies posed a challenge, as compatibility issues and version mismatches often arose. Extensive research and troubleshooting were needed to ensure successful installation and configuration. Finding a suitable dataset for training the deep learning model was another hurdle, as it required a comprehensive and diverse collection of plant disease images. Multiple sources were explored, and data preprocessing was necessary to ensure data quality and consistency.

Debugging Python libraries to run the AI model also proved to be a challenge. Compatibility issues, conflicting dependencies, and coding errors needed to be addressed to ensure smooth execution of the model. Integrating the USB camera and thermal camera to overlay the images was a complex task, involving understanding the camera interfaces and ensuring proper synchronization between the two. Extensive testing and calibration were required to achieve accurate overlay and alignment.

Extracting data from the thermal camera presented its own set of challenges, as it involved capturing and processing thermal images in real-time. Understanding the data format, handling temperature conversions, and dealing with noise and calibration were all crucial steps in obtaining accurate thermal data. Acquiring data from the AHT10 sensor using the GPIO pins required careful wiring and configuration, along with understanding the communication protocol. Troubleshooting connectivity issues, ensuring accurate readings, and handling sensor errors were part of the process.

One significant challenge was improving the accuracy of the AI model from 70% to the desired 90%. This involved fine-tuning the model architecture, optimizing hyperparameters, increasing the dataset size, and implementing data augmentation techniques. Extensive experimentation and analysis were required to identify the factors affecting accuracy and iteratively refine the model.

One notable challenge was the inability to run the AI model on the Jetson due to Tensorflow restrictions. This required exploring alternative solutions such as waiting for a new compatibility update for Jetson's CUDA or considering implementing the project in a cloud environment to leverage more recent versions of Tensorflow. Despite these challenges, each obstacle was approached with determination, research, and iterative refinement. Through perseverance and problem-solving, these challenges were overcome, leading to the successful implementation of the project.

Future Works:

Future work for the project includes several areas of improvement and expansion to enhance its functionality and usability. Some key aspects to consider are:

1. Software Upgrades: Since the current version of TensorFlow (2.4.0) is not compatible with Jetson for running the entire project, future work involves waiting for a new compatibility CUDA update or exploring alternative frameworks that support the latest TensorFlow version (2.12.0). This would ensure access to the latest features, optimizations, and bug fixes, further enhancing the performance of the deep learning models.
2. Cloud Implementation: To overcome the limitations of running the project solely on Jetson, a future enhancement could involve implementing a cloud-based solution. By leveraging the power of cloud computing platforms, such as AWS, Azure, or Google Cloud, the project could benefit from higher computational resources and scalability. The models and processing can be offloaded to the cloud, allowing for faster training, inference, and improved overall performance.
3. Web User Interface (UI): Developing a web-based user interface would greatly enhance the project's usability and accessibility. Implementing a user-friendly UI would enable non-technical users to easily interact with the system, input their preferences, view results, and perform tasks such as initiating disease detection, monitoring plant health, and accessing historical data. The web UI could be built using frameworks like Flask or Django, providing a platform-independent interface accessible from various devices.
4. Data Visualization and Analytics: Enhancing the project's analytical capabilities by incorporating advanced data visualization and analytics tools can provide deeper insights into plant health trends, disease patterns, and overall crop performance. Integrating libraries like Plotly or Tableau would enable users to explore and interpret data effectively, enabling data-driven decision-making and optimizing farming strategies.
5. Integration with IoT and Automation: Integrating the project with Internet of Things (IoT) technologies and automation systems would enable real-time monitoring, remote control, and autonomous operation. This could involve connecting sensors, cameras, and

actuators to a network, allowing for automated data collection, disease detection, and even autonomous plant care actions such as irrigation or pest control.

By focusing on these areas of future work, the project can evolve into a comprehensive and robust plant disease detection and monitoring system, empowering farmers, researchers, and agricultural stakeholders with advanced tools for efficient crop management, improved decision-making, and sustainable agricultural practices.

Conclusion:

In conclusion, the project successfully developed a plant disease detection system using a combination of thermal imaging, environmental sensors, and deep learning algorithms. By integrating a thermal camera for leaf temperature measurement, an AHT10 sensor for humidity monitoring, and a USB camera for capturing frames, the system provides comprehensive data for accurate disease detection and analysis. The utilization of Vapor Pressure Deficit (VPD) as an indicator for irrigation need further enhances the system's functionality and practicality.

Through the implementation of deep learning algorithms, specifically Convolutional Neural Networks (CNNs), the system demonstrates impressive accuracy in identifying and classifying various plant diseases. The trained model, optimized using TensorFlow and Keras, exhibits robust performance in real-time disease detection, contributing to early identification and mitigation of plant health issues.

The project also leverages essential software tools such as OpenCV, NumPy, Jupyter Notebook, and Matplotlib to facilitate data processing, preprocessing, visualization, and model evaluation. These tools enable seamless integration with deep learning frameworks, efficient handling of image data, iterative experimentation, and insightful analysis of model performance.

The combination of hardware components, AI algorithms, and software tools establishes a comprehensive and effective solution for plant disease detection and analysis. The project's outcomes hold significant potential for revolutionizing agriculture practices, enabling timely interventions, and maximizing crop yield.

Overall, the successful implementation of this project highlights the transformative impact of AI technologies in agriculture. By harnessing the power of thermal imaging, environmental sensing, and deep learning, the system offers a valuable tool for farmers and researchers to monitor plant health, identify diseases, and optimize cultivation practices. With further advancements and integration into agricultural systems, this technology holds promise for improving crop management, reducing crop losses, and contributing to sustainable food production in the future.

References:

1. Research:

- "Advances in Remote Sensing." SCIRP. Accessed May 17, 2023. https://www.scirp.org/journal/ars/.

- Application of infrared thermography for irrigation scheduling of ... Accessed May 17, 2023. https://www.researchgate.net/publication/355906427_Application_of_Infrared_Thermography_for_Irrigation_Scheduling_of_Horticulture_Plants.

- Author links open overlay panel Olivier Debauche a b, a, b, c, d, and Abstract Overcoming population growth dilemma with less resources of soil and water. "Edge Ai-IoT Pivot Irrigation, Plant Diseases, and Pests Identification." Procedia Computer Science, November 11, 2020. https://www.sciencedirect.com/science/article/pii/S1877050920322742.

- Cloud of things in smart agriculture: Intelligent ... - IEEE xplore. Accessed May 18, 2023. https://ieeexplore.ieee.org/document/7879140.

- Thermal imaging for Smart Agriculture Irrigation System to support ir 4 ... Accessed May 18, 2023. https://www.researchgate.net/profile/Aznida-Abu-Bakar-Sajak/publication/344862746_Thermal_Imaging_for_Smart_Agriculture_Irrigation_System_to_support_IR_40_Initiative/links/5f945127299bf1b53e40d24d/Thermal-Imaging-for-Smart-Agriculture-Irrigation-System-to-support-IR-40-Initiative.pdf.

- Koverda, Peter. "The Ultimate Vapor Pressure Deficit (VPD) Guide." Pulse Grow, February 12, 2020. https://pulsegrow.com/blogs/learn/vpd.

- Grace, Delia. "Infectious Diseases and Agriculture." Edited by Pasquale Ferranti, Elliot M. Berry, and Jock R. Anderson. Encyclopedia of Food Security and Sustainability, 2019. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7161382/.

- "Jetson Projects of the Month: Ai Thermometer and Self-Navigating Robot for Search and Rescue." NVIDIA Technical Blog, April 6, 2023. https://developer.nvidia.com/blog/jetson-projects-of-the-month-ai-thermometer-and-self-navigating-robot-for-search-and-rescue/.

- Massmann, Adam, Pierre Gentine, and Changjie Lin. "When Does Vapor Pressure Deficit Drive or Reduce Evapotranspiration?" Journal of advances in modeling earth systems, October 2019. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6919419/.

- Thermal images of healthy cucumber leaves (upper) and leaves inoculated ... Accessed June 5, 2023. https://www.researchgate.net/figure/Thermal-images-of-healthy-cucumber-leaves-upper-and-leaves-inoculated-with-FOC-lower_fig3_232719555.

2. Dataset:

- Pratikkayal. "Pratikkayal/PlantDoc-Dataset: Dataset Used in 'Plantdoc: A Dataset for Visual Plant Disease Detection' Accepted in Cods-COMAD 2020." GitHub. Accessed June 10, 2023. https://github.com/pratikkayal/PlantDoc-Dataset.

- Bhattarai, Samir. "New Plant Diseases Dataset." Kaggle, November 18, 2018. https://www.kaggle.com/datasets/vipoooool/new-plant-diseases-dataset.

- "Papers with Code - Plantdoc Dataset." Dataset | Papers with Code. Accessed June 10, 2023. https://paperswithcode.com/dataset/plantdoc.

3. Software:

- "API Documentation: Tensorflow V2.12.0." TensorFlow. Accessed June 10, 2023. https://www.tensorflow.org/api_docs.

- Team, Keras. "Keras Documentation: Keras API Reference." Keras. Accessed June 10, 2023. https://keras.io/api/.

- "Jetson Nano Developer Kit." NVIDIA Developer, September 28, 2022. https://developer.nvidia.com/embedded/jetson-nano-developer-kit.

- OpenCV documentation index. Accessed May 28, 2023. https://docs.opencv.org/.

- "The Python Language Reference." Python documentation. Accessed May 20, 2023. https://docs.python.org/3/reference/.

- LeBlanc-Williams, M. "CircuitPython Libraries on Linux and the Nvidia Jetson Nano." Adafruit Learning System. Accessed June 10, 2023.

https://learn.adafruit.com/circuitpython-libraries-on-linux-and-the-nvidia-jetson-nano/circuitpython-dragonboard.

- Groupgets. “Groupgets/PURETHERMAL1-UVC-Capture: USB Video Class Capture Examples for Purethermal 1 / Purethermal 2 FLIR Lepton Dev Kit.” GitHub. Accessed June 1, 2023. https://github.com/groupgets/purethermal1-uvc-capture.
- Dusty-Nv. “Jetson-Inference/Docs/Posenet.Md at Master · Dusty-NV/Jetson-Inference.” GitHub. Accessed June 1, 2023. https://github.com/dusty-nv/jetson-inference/blob/master/docs/posenet.md.
- Dusty-Nv. “Dusty-NV/Jetson-Inference: Hello AI World Guide to Deploying Deep-Learning Inference Networks and Deep Vision Primitives with TENSORRT and Nvidia Jetson.” GitHub. Accessed June 1, 2023. https://github.com/dusty-nv/jetson-inference.
- “Jetpack SDK 4.6.3.” NVIDIA Developer, November 30, 2022. https://developer.nvidia.com/jetpack-sdk-463.
- “Jetson Nano Developer Kit.” NVIDIA Developer, September 28, 2022. https://developer.nvidia.com/embedded/jetson-nano-developer-kit.

4. Hardware:

- Digital-output relative humidity & temperature sensor/module AHT10 ... Accessed June 20, 2023. https://www.sparkfun.com/datasheets/Sensors/Temperature/AHT10.pdf.
- Raspberry Pi. “Buy A Raspberry Pi Camera Module 2.” Raspberry Pi. Accessed June 20, 2023. https://www.raspberrypi.org/products/camera-module-v2/.
- Product overview - ECA. Accessed June 1, 2023. https://server4.eca.ir/eshop/AHT10/Aosong_AHT10_en_draft_0c.pdf.
- Lewicki, Tomasz. “Ai Thermometer.” Hackster.io, December 24, 2021. https://www.hackster.io/tomasz-lewicki/ai-thermometer-2bacb4.
- xyth0rn. “XYTH0RN/Thermalfacedetection.” GitHub. Accessed June 1, 2023. https://github.com/xyth0rn/ThermalFaceDetection.
- “PureThermal 2 - Flir Lepton Smart I/O Module by GetLab.” GroupGets. Accessed June 1, 2023. https://groupgets.com/manufacturers/getlab/products/purethermal-2-flir-lepton-smart-i-o-module.

- "Lepton®." Lepton LWIR Micro Thermal Camera Module | Teledyne FLIR. Accessed June 10, 2023. https://www.flir.com/products/lepton/?model=500-0771-01&vertical=microcam&segment=oem.

- MLX90640 32x24 IR array - melexis. Accessed June 10, 2023. https://www.melexis.com/-/media/files/documents/datasheets/mlx90640-datasheet-melexis.pdf.

Appendix:

1. *Final_AI_Model.ipynb*
2. *Final_Capture_Calculation.ipynb*
3. *loss1.png*
4. *loss2.png*
5. *loss3.png*
6. *accuracy1.png*
7. *accuracy2.png*
8. *accuracy3.png*
9. *trained1.h5*
10. *trained2.h5*
11. *trained3.h5*
12. *test folder*
13. *test_sample folder*
14. *predictions folder*
15. *dataset folder*
16. *hardware codes/AHT.py*
17. *hardware codes/Lepton 3.5/opencv-capture.py*
18. *hardware codes/Lepton 3.5/PT2_Lep_Temp_Array.py*
19. *hardware codes/Lepton 3.5/uvc-deviceinfo.py*
20. *hardware codes/Lepton 3.5/uvc-radiometry.py*
21. *hardware codes/Lepton 3.5/uvc-types.py*
22. *hardware codes/Images folder*