



Tomato Leaf Health Classification with Management Recommendations using CNN-based DenseNet-121 for Agricultural Sustainability

Rosh Hashana S. Torres¹

New Era University – College of Informatics and
Computing Studies

9 Central Ave, New Era, Quezon City, 1107 Philippines
rosh.torres@neu.edu.ph

Jannah R. Soriano²

New Era University – College of Informatics and
Computing Studies

9 Central Ave, New Era, Quezon City, 1107 Philippines
jannah.soriano@neu.edu.ph

Abstract

Technological innovations had transformed agriculture, providing solutions for enhanced sustainability. This study addressed the problem of plant health management in agriculture, by proposing a novel approach using a Convolutional Neural Network (CNN)-based DenseNet-121 model. Using advances in deep learning, particularly in computer vision, the researchers created a web-based application for tomato leaf health classification with management recommendations. The PlantVillage and self-acquired dataset, containing images of tomato leaves depicting both healthy and various diseased states, was used for training and validation. The custom DenseNet-121 model showed remarkable performance metrics after thorough experimentation, obtaining high accuracy, sensitivity, specificity, and F1 scores in training and validation processes. The evaluation of the trained model offered positive results, with an accuracy of 97.37% on the training set and 98.59% on the validation set. Additionally, the model had reasonable specificity (99.77% training, 99.88% validation), sensitivity (97.15% training, 98.59% validation), and F1 scores (97.39% training, 98.67% validation), demonstrating that it was efficient in health classification. By incorporating the developed model into a web-based application, farmers and agricultural workers had access to a valuable tool for plant health management. Furthermore, the addition of management recommendations increased the application's usability by offering actionable information on how to effectively manage diagnosed leaf health. This study significantly contributed to sustainable farming methods by combining modern technology with traditional methods of agriculture, ultimately aiming for the betterment of society.

Keywords - Agriculture, Deep Learning, Tomato Leaf, Leaf Health Classification, CNN, DenseNet-121

Introduction

This study aims to develop a tomato leaf health classification application using a CNN-based DenseNet-121 model and providing management recommendations for effective strategies. The researchers will cover two (2) classes, namely, healthy leaf and diseased leaf.

The following can be derived from previous studies and related literature:

According to Wiley (2023), despite the fact that tomatoes are valuable economically and nutritionally, the crop is prone to a number of leaf diseases brought on by fungi, bacteria, or viruses. Septoria leaf spot, early blight, late blight, and mosaic

virus is among the common diseases that affect tomato leaves. The yield, quality, and overall health of tomato plants may all be negatively impacted by these diseases.

Early identification and accurate diagnosis of tomato leaf health are essential for effective management and sustainable agriculture. However, manual examination by farmers can be subjective, time-consuming, and prone to errors, delaying responses and increasing financial losses (Dawod & Dobre, 2022).

Deep learning, specifically convolutional neural networks (CNNs), has revolutionized the field of image classification and object detection (Sujatha et al., 2021). CNNs can automatically learn and extract

complex features from images, making them well-suited for leaf disease classification.

Kumar et al. (2023) proposed a novel plant disease detection technique based on deep learning using the VGG16 architecture. The study specifically targeted disease detection in tomato and potato plants. The approach achieved significant disease detection accuracy of 88.6%.

DenseNet-121 has shown impressive performance in various image classification tasks (Barbedo, 2018; Mohanty et al., 2016). Its dense connectivity allows the model to efficiently extract and propagate features, leading to improved accuracy.

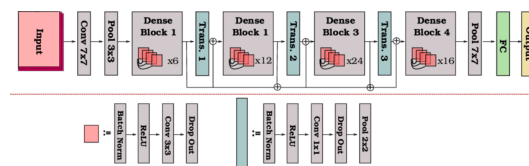


Figure 1. DenseNet-121 Architecture

Moreover, DenseNet-121 can be trained with transfer learning, utilizing pre-trained models on large-scale datasets such as ImageNet (Hussain et al., 2018). This transfer learning approach enables the model to use knowledge learned from a vast amount of data and generalize well to new tasks with limited training data.

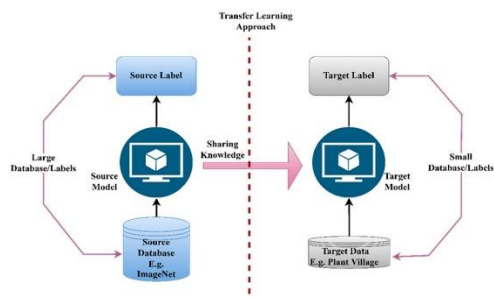


Figure 2. Basic Idea behind Transfer Learning

Andrew et al. (2022) focused on using deep learning techniques for efficient plant disease identification. They fine-tuned pre-trained CNN models, including DenseNet-121, ResNet-50, VGG-16, and Inception V4. DenseNet-121 achieved a classification accuracy of 99.81%, surpassing other models. The study emphasized the potential of DenseNet-121 in accurately identifying plant diseases and its superiority over other models in terms of classification accuracy.

From the previous survey of related literature, the following can be established:

- Leaf diseases present a significant threat to agricultural sustainability, as they can cause substantial losses in crop production and reduce the quality and quantity of food. Identifying and

diagnosing leaf health accurately and early on is crucial for effective management.

- Deep learning models offer promising solutions by automating the feature extraction and classification tasks. These advancements in technology enable early detection and accurate diagnosis, leading to more effective management, thus enhancing agricultural sustainability.
- DenseNet-121 is a powerful deep learning architecture for image classification tasks. Its dense connectivity promotes feature reuse and gradient flow, enabling the model to capture intricate patterns and achieve high accuracy. Using transfer learning further enhances its performance, especially when training data is limited.

This paper then asked: How can a CNN-based DenseNet-121 model be effectively utilized to classify tomato leaf health and provide management recommendations for agricultural sustainability?

This research fulfilled the following:

1. Apply several preprocessing techniques to improve the quality and diversity of the tomato leaf images dataset for training and validating the model.
2. Develop a customized DenseNet-121 model for tomato leaf health image classification, evaluating its performance through classification accuracy, sensitivity, specificity, and F1 score.
3. Create a web-based application that utilizes the trained model to classify tomato leaf health based on images uploaded by users.
4. Integrate management recommendations into the application, imposing the diagnosed health information to suggest appropriate management strategies.

This study tested the following hypotheses:

Ha: The customized DenseNet-121 model, trained on a preprocessed dataset of tomato leaf images and integrated into a web-based application, effectively classify tomato leaf health with high accuracy, sensitivity, specificity, and F1 score.

H0: The customized DenseNet-121 model, trained on a preprocessed dataset of tomato leaf images and integrated into a web-based application, does not effectively classify tomato leaf health with high accuracy, sensitivity, specificity, and F1 score.

Methodology

This project design outlines the specific methods and procedures employed in this study.

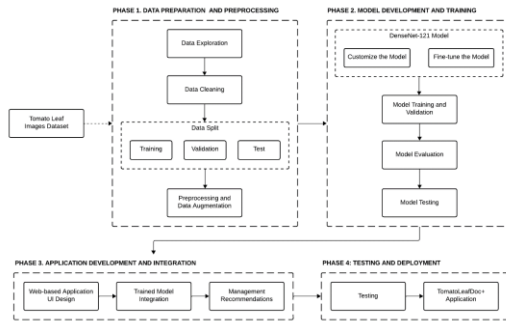


Figure 3. Project Design of the Study

In order to facilitate acquiring the project design, several tools and libraries need to be installed:

- **Jupyter Notebook.** Online application for generating and sharing documents with live code and visualizations. <https://jupyter.org/>
- **OpenCV.** Exceptional tool for image processing and computer vision tasks. <https://opencv.org/>
- **Pandas:** Python library for efficient data manipulation and analysis. <https://pandas.pydata.org/>
- **Matplotlib.** Python library for 2D charting. <https://matplotlib.org/>
- **Seaborn.** Simplified statistical visualization built on Matplotlib. <https://seaborn.pydata.org/>
- **Numpy.** Python library for quick numerical computation with arrays and matrices. <https://numpy.org/>
- **TensorFlow and Keras.** Connected for fast model development; Keras for high-level neural network API. <https://www.tensorflow.org/>
- **scikit-learn.** Python module for statistical modeling and machine learning. <https://scikit-learn.org/stable/>
- **Pillow (PIL).** Foundational Python libraries for image manipulation. <https://python-pillow.org/>
- **Flask.** Compact web framework for Python web development. <https://flask.palletsprojects.com/en/3.0.x/>

Tomato Leaf Images Dataset. In this study, researchers used the "PlantVillage" dataset from Penn State University, led by Dr. David Hughes. This dataset features single-leaf images in ten categories: one healthy and nine diseased, including bacterial spot, early blight, late blight, leaf mold, septoria leaf spot, two-spotted spider mite, target spot, mosaic virus, and yellow leaf curl virus. The leaves were photographed outdoors against grey or black backgrounds on sunny or cloudy days. This study focused exclusively on tomato plant leaves.

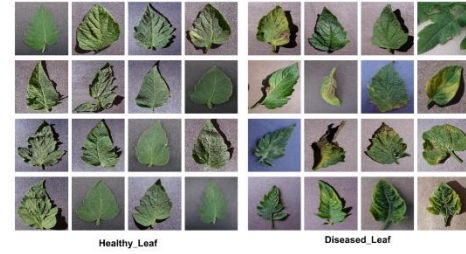


Figure 4. Sample Images from the PlantVillage Dataset for Tomato Leaf

To improve the dataset's credibility, diversity, and quality, researchers added tomato leaf images from Quezon City, Philippines. These were taken with a 13-megapixel Samsung Galaxy J7 (2016) and a 12-megapixel Samsung Galaxy Note 8 (2017).



Figure 5. Location of Tomato Plants in Quezon City (May 2024)

This expansion broadens the geographical range and includes images captured under different conditions and with varied equipment.

The first phase involves preparing data for a tomato leaf health classification application, which includes acquiring images from various sources like Kaggle and Mendeley Data, exploring the data to understand its characteristics, cleaning it to enhance quality, and performing tasks such as splitting, resizing, and augmenting the dataset.

Data Split. The dataset held two classes; it consisted of one (1) healthy leaf class and one (1) diseased leaf class. The researchers split this dataset into training samples, validation samples, and testing samples. The DenseNet-121 model was trained with 80% of the dataset; 10% was used for validation, and the other 10% was used for testing

Table 1. Image Count Distribution by Dataset Type

Dataset Type	Image Count
Training	640
Validation	80
Testing	80
Total	800

Preprocessing and Data Augmentation. For DenseNet-121, the input size is $224 \times 224 \times 3$ (height, width, and channel width), so in order to meet

the input requirement of the DenseNet-121 model, all images are rescaled to 224×224 pixels. Though the dataset is huge, the images match the real-life images captured by farmers. To overcome overfitting, regularization techniques, such as data augmentation after preprocessing, were introduced.

Table 2. Parameters of the Data Augmentation Techniques

Parameter	Value
Scaling	1./255
Rotation	40
Shearing	0.2
Zoom	0.2
Flipping	Vertical/Horizontal
Shifting	Width/Height with range 0.2
Filling	nearest

The images were not duplicated but augmented during the training process, so the physical copies of the augmented images were not stored but were temporarily used in the process.



Figure 6. Sample Preprocessed Images

In the second phase, the focus shifts to developing and training the tomato leaf health classification model using DenseNet-121, a deep learning architecture for image classification. The model undergoes iterative adjustments during training, validated alongside to assess performance, and then evaluated using appropriate metrics to ensure accurate health classification.

DenseNet-121 Model. The researchers custom-built this deep-learning model to use its capabilities in feature extraction and classification tasks. The researchers fine-tune the DenseNet-121 model on their dataset of healthy and diseased leaf images.

Table 3. Hyperparameter Specifications and Configurations

Attribute	Value
Image Size	224
Channel	3
Batch Size	32
Epoch	30
Regularization	Batch Normalization
Dropout	0.5
Number of hidden layers	2
Number of neurons per layer	1024, 512
Activation	ReLU
Number of Classes	6
Output Layer Activation	softmax
Optimizer	Adam
Learning Rate	0.001
Loss	categorical cross-entropy
Model Checkpoint, CSV Logger, Learning Rate Scheduler, Early Stopping	Yes

These parameters are crucial as they directly impact the model's learning process and ultimate performance. The image size, set to 224 pixels, ensures consistency in input dimensions, while the three channels correspond to the standard RGB format. A batch size of 32 dictates the number of samples processed in each iteration.

The regularization technique adopted is Batch Normalization, helping in stabilizing and accelerating the training process. A dropout rate of 0.5 is implemented to mitigate overfitting. The model architecture consists of two hidden layers, with 1024 and 512 neurons, respectively, each utilizing the Rectified Linear Unit (ReLU) activation function to introduce non-linearity.

With six classes to predict, the output layer employs softmax activation, generating probabilities for each class.

The Adam optimizer, known for its adaptability and efficiency, is utilized with a learning rate of 0.001 to adjust model parameters during training. Categorical cross-entropy serves as the loss function, evaluating the disparity between predicted and actual class distributions.

Additionally, several training enhancements are added, including model checkpointing, CSV logging, a learning rate scheduler, and early stopping, facilitating efficient model monitoring and management throughout the training process.

Model Evaluation. The performance of the model was evaluated through various metrics,

including classification accuracy, sensitivity, specificity, and F1 score. These metrics give insight on multiple aspects of the model's performance, resulting in a thorough assessment of its predictive capabilities.

Accuracy is a measure of the overall correctness of the model's predictions.

$$\text{accuracy} = \frac{TP + TN}{TP + FN + TN + FP}$$

Specificity indicates the model's ability to correctly identify negative instances, such as unhealthy plants in this context.

$$\text{specificity} = \frac{\text{True Negative}}{(\text{True Negative} + \text{False Positive})}$$

Sensitivity, also known as recall, indicates the model's capacity to correctly detect positive instances, like healthy plants in this scenario.

$$\text{sensitivity} = \frac{\text{True Positive}}{(\text{True Positive} + \text{False Negative})}$$

The F1 score is the harmonic mean of precision and recall, giving an overall measure of a model's accuracy, particularly when there's an uneven class distribution.

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Model Testing. Involves loading the best weights of a trained model, preparing a data generator for the test dataset, and predicting class labels for the test data. These predictions are then compared to true class labels to evaluate the model's ability to generalize and classify new images accurately. A confusion matrix is generated to assess prediction standards, and a comprehensive classification report is produced using scikit-learn functions to evaluate the model's performance across various metrics.

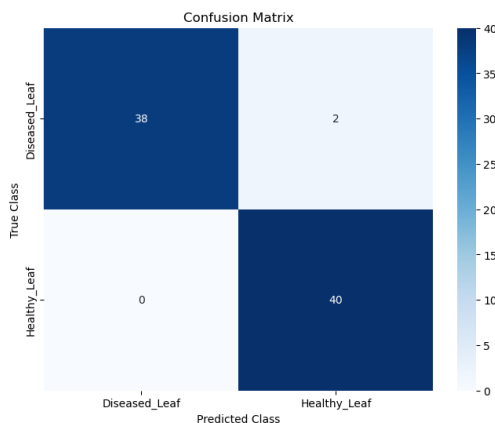


Figure 7. Confusion Matrix

This almost perfect diagonal line from left to right indicates that the binary classification model performed exceptionally well in its predictions.

In the third phase, the focus shifts to creating a web-based application for tomato leaf health classification, featuring an intuitive interface. The application integrates the trained DenseNet-121 model for predicting leaf health from input images and includes a management recommendation system based on predicted health to offer effective management strategies.

Web-based Application UI Design. A web application was developed using Visual Studio Code. Using HTML, CSS, and JavaScript, a cohesive design with a layout that supports usability and engagement was developed, incorporating interactive elements and navigation menus. The application features responsive design using Bootstrap and utilizes Flask for backend functionality.

Management Recommendations. In this study, the researchers utilized the Plantix's library of management recommendations for crop health encompassing both organic and chemical methods. This library serves as a reliable resource for farmers and agriculturists as it was developed in collaboration with leading agronomists and backed by extensive research. These management recommendations are integrated within the Flask web application. This integration allows users to upload an image of a tomato plant with potential disease symptoms and receive immediate treatment recommendations based on the predicted disease. This integration allowed users to upload an image of a tomato plant and receive management recommendations based on the predicted health.

In the fourth and final phase, the developed application undergoes rigorous testing to verify its usability and accuracy, including assessments of health prediction and management recommendations using diverse input images. Upon successful completion of testing, the application is deployed for users, offering a valuable tool for identifying tomato leaf health and receiving management recommendations, thereby supporting agricultural sustainability efforts.

Testing. The developed application undergoes testing to ensure functionality and reliability. This includes testing with various input images representing tomato leaf healthy and diseased to evaluate the integrated model's prediction and management recommendations capabilities. Through these tests, the application's efficacy in real-world scenarios is assessed, enabling the identification and rectification of any issues before deployment. Actual tomato leaves sourced from tomato plants are utilized during testing to validate and refine the application's performance.



Figure 8. Actual Tomato Leaves Utilized for System Testing

Results and Discussions

Subsequent to training, the researchers visualize the training and validation performance of custom DenseNet-121 model over multiple epochs.

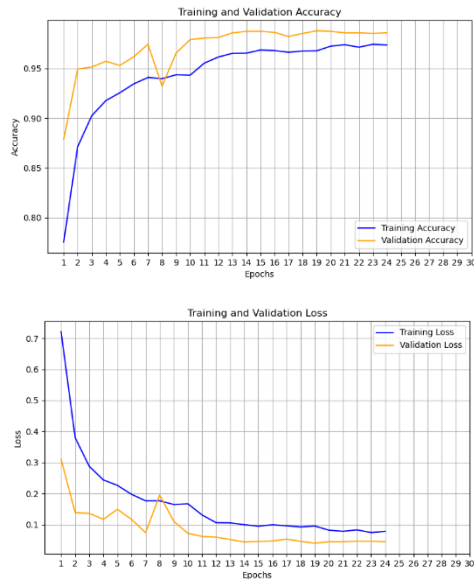


Figure 9. Learning Curve of the Custom DenseNet-121

The model's performance rapidly increased during the initial epochs, as indicated by increasing values of metric scores on both the training and validation sets. Notably, the validation accuracy peaked at around 98.8%, demonstrating that the model was successfully generalizing to unseen data. Also, the validation loss steadily decreased throughout epochs. However, at epoch 19, the validation accuracy plateaued while the validation loss began to fluctuate, indicating that the model's performance was stabilizing. As a result, the early stopping was activated, preventing overfitting by terminating the training process at epoch 24 when the model's performance on the validation set failed to improve.

Table 4. Training and Validation Evaluation

	Train	Validation
Accuracy	97.37%	98.59%
Loss	7.82%	4.48%

During the researchers' data exploration, they found out that their data is imbalance so they just can't rely on accuracy so it is not the best metric for the model's performance as it may give vague results that's why they used other metric scores.

Table 5. Specificity, Sensitivity, and F1 Score

	Train	Validation
Specificity	99.77%	99.88%
Sensitivity	97.15%	98.59%
F1 Score	97.39%	98.67%

The researchers implemented a system within the Flask web application to integrate tomato leaf health classification with management recommendations. Upon receiving an image of a tomato leaf, the application analyzes the image and predicts its health. Once the health is diagnosed, the application retrieves corresponding management recommendations from its predefined dictionary. These recommendations, which were divided into three components—Symptoms, Organic Control, and Chemical Control—were presented to the user in organized way. The web-based application that was developed not only displays the predicted health label but also provides a detailed description of the symptoms observed on the tomato plant.

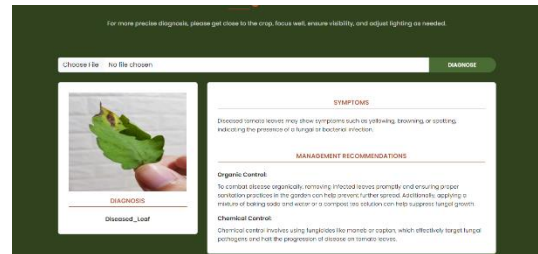


Figure 10. Application User Interface with Diagnosis Result

The figure above shows the result obtained when the user uploaded an image of a diseased tomato leaf. As you can see in the interface, the application presents the disease's symptoms and offers remedies to mitigate its spread and potentially restore leaf health.

During testing, fifty (50) images were used for each class to thoroughly evaluate the application's performance.

Table 6. Application Performance Results

Classes Names	Correctly Identified	Total Tested	Accuracy
Healthy_Leaf	45	50	90%
Diseased_Leaf	47	50	94%

These results demonstrate the integrated model's reliable disease prediction. Thorough evaluation and refinement ensured the application meets desired standards of usability and effectiveness.

Aside from developing a web-based application that utilizes the trained model to classify tomato leaf health based on images uploaded by users, the researchers also implemented a real-time Tomato Leaf Health Detection system using a webcam. The process begins by loading necessary libraries and the model along with class labels. The webcam captures images in real-time, which are then resized, converted to an array, reshaped, and normalized.

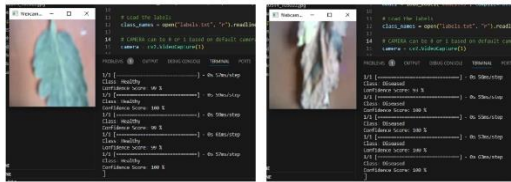


Figure 11. Tomato Leaf Health Detection

Conclusion

This study aimed to address the crucial issue of plant disease management in agriculture, particularly focusing on tomato leaf health, through the development of CNN-based DenseNet-121 model. Integrating deep learning techniques into agricultural practices has great potential for improving agricultural sustainability. Using the capabilities of convolutional neural networks, specifically the DenseNet-121 architecture, the researchers developed a tomato leaf health classification application with remarkable accuracy and efficiency.

To summarize the results, the study successfully developed a tomato leaf health classification application using a customized DenseNet-121 CNN model. This application accurately classifies tomato leaf health and provides tailored management recommendations, helping farmers, agricultural workers, consumers, and plant health authorities in protecting crops and enhancing productivity. The integration of deep learning techniques into agriculture, specifically the DenseNet-121 model, demonstrated exceptional accuracy, specificity, sensitivity, and F1 scores, highlighting its reliability. This advancement represents a significant step toward early disease detection, effective management strategies, agricultural sustainability, and economic prosperity.

The following research objectives were accomplished:

1. Applied various preprocessing techniques to enhance the quality and diversity of the dataset.
2. Developed a customized DenseNet-121 model for tomato leaf health image classification, and thoroughly evaluate its performance.

3. Created a web-based application that utilizes the trained model to classify tomato leaf health based on images uploaded by users.
4. Integrated management recommendations into the application to provide complete assistance for agricultural sustainability.

This research concludes that the customized DenseNet-121 model, trained on a preprocessed dataset of tomato leaf images and integrated into a web-based application, effectively classifies tomato leaf health with high accuracy, sensitivity, specificity, and F1 score. Therefore, this research proves the alternative hypothesis: H_a .

Recommendations

The researchers recommend the following for future research:

Mobile Application Development. Mobile app development could revolutionize farming with real-time disease classification and treatment advice, empowering farmers to diagnose and address plant health issues in the field promptly.

Multi-leaf Disease Classification. Expanding the classification framework to diagnose diseases on groups of leaves offers a more thorough approach, giving farmers a better understanding of overall plant health and enabling more effective disease management.

Integration with Agricultural IoT Devices. Integrating the application with agricultural IoT devices, like sensors and drones, enhances disease detection and management by providing real-time environmental data and images for more accurate recommendations to farmers.

Environmental Factors in Disease Classification. Integrating environmental factors into disease classification enhances diagnosis and treatment recommendations by considering climate, soil conditions, and geographic location, ensuring tailored solutions for specific environmental conditions.

Acknowledgements

The successful completion of this endeavor is owed to the invaluable assistance of numerous individuals, whose contributions and cooperation we deeply applaud and acknowledge. We extend our deepest gratitude to our supportive family and friends, as well as our thesis advisor, Dr. Marc P. Laureta, College of Informatics and Computing Studies, and New Era University for their guidance and resources. Special appreciation is also due to referenced authors and all participants. Above all, we thank the Almighty God for granting us the ability and strength to accomplish this study.

References

- Agriculture in the Global Economy. *Journal of Economic Perspectives*, 28(1), 121–146. <https://doi.org/10.1257/jep.28.1.121>
- Andrew, J., R., J. E., Popescu, D., Chowdary, M. K., & Hemanth, D. J. (2022). Deep Learning-Based Leaf Disease Detection in Crops Using Images for Agricultural Applications. *Agronomy*, 12(10), 2395. <https://doi.org/10.3390/agronomy12102395>
- Barbedo, J. G. A. (2018). Factors influencing the use of deep learning for plant disease recognition. *Biosystems Engineering*, 172, 84–91. <https://doi.org/10.1016/j.biosystemseng.2018.05.013>
- Dawod, R. G., & Dobre, C. (2022). Upper and Lower Leaf Side Detection with Machine Learning Methods. *Sensors*, 22(7), 2696. <https://doi.org/10.3390/s22072696>
- GmbH, P. (n.d.). *Pests & Diseases / Plantix*. Plantix. <https://plantix.net/en/library/plant-diseases/>
- Hussain, M., Bird, J. T., & Faria, D. R. (2018). A Study on CNN Transfer Learning for Image Classification. In *Advances in intelligent systems and computing* (pp. 191–202). Springer Nature. https://doi.org/10.1007/978-3-319-97982-3_16
- Kumar, A., & Kumar, A. (2023). Plant Disease Detection using VGG16. *International Journal of Creative Research Thought*, 11(1). <https://ijcrt.org/papers/IJCRT2301347.pdf>
- Mohanty, S. P., Hughes, D. L., & Salathé, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. *Frontiers in Plant Science*, 7. <https://doi.org/10.3389/fpls.2016.01419>
- PlantVillage*. (n.d.). <https://plantvillage.psu.edu/>
- PlantVillage Dataset*. (2019, September 1). Kaggle. <https://www.kaggle.com/datasets/abdallahali/dev/plantvillage-dataset>
- Sujatha, R., Chatterjee, J. M., Zaman, N., & Brohi, S. N. (2021). Performance of deep learning vs machine learning in plant leaf disease detection. *Microprocessors and Microsystems*, 80, 103615. <https://doi.org/10.1016/j.micpro.2020.103615>

Wiley, D. (2023). 10 common tomato plant diseases that can wreck your crop. *Better Homes & Gardens*. <https://www.bhg.com/gardening/vegetable/vegetables/tomato-plant-diseases/>

Authors' Bio-note



Rosh Hashana S. Torres (Bachelor of Science in Computer Science, New Era University – College of Informatics and Computing Studies)

Rosh is a final-year Bachelor of Science student majoring in Computer Science at New Era University's College of Informatics and Computing Studies. Passionate about technology and its applications, she actively seeks opportunities to expand her knowledge and skills in the field. She enjoys attending seminars the latest developments in technology to stay updated with the rapidly evolving industry trends. She also serves as a member of the Academic Committee in the Association of Computer Science Students, where she contributes to enhancing the academic experience of her peers.



Jannah R. Soriano (Bachelor of Science in Computer Science, New Era University – College of Informatics and Computing Studies)

Jannah is a final-year Bachelor of Science student majoring in Computer Science at New Era University's College of Informatics and Computing Studies. She has carved out a niche in the cutting-edge domains of artificial intelligence and data analytics. In addition to her academic pursuits, she plays a pivotal role within the Association of Computer Science Students, serving as a valued student relations committee member. In this capacity, she actively contributes to the organization's mission of promoting academic excellence and fostering a vibrant community of aspiring computer scientists.