SMART PLANT DETECTOR

AUTOMATED PLANT HEALTH DIAGNOSIS AND REMEDY SUGGESTED USING MACHINE LEARNING

Dhriti
AI-DS
Indira Gandhi Delhi Technical University For Women
Delhi, India
dhritigupta2212@gmail.com

Namya Gautam
AI-DS
Indira Gandhi Delhi Technical University For Women
Delhi, India
namyaatwork@gmail.com

Kiran Malik
AI-DS
Indira Gandhi Delhi Technical University For Women
Assistant Professor
Delhi, India

I. ABSTRACT

Plant health monitoring is essential for sustainable agriculture and effective gardening practices. Traditional methods for detecting diseases often require expert knowledge and can cause delays. These issues lead to reduced crop yield and quality. This research introduces a machine learning system that can identify plant species and assess their health condition. The method uses convolutional neural networks (CNNs) trained on datasets like Plant Pathology 2020 and PlantVillage [1][2]. Input images are preprocessed with OpenCV and PIL to improve feature extraction before classification. A secondary module differentiates between healthy and unhealthy plants. The system also provides general plant-related information and care guidelines. Built with Python, TensorFlow, Keras, and Streamlit, the prototype reliably recognizes plants and classifies their health. This study helps advance automated plant monitoring by providing an accessible and scalable solution that aids farmers, gardeners, and plant enthusiasts in effectively managing plant health awareness.

II. INTRODUCTION

Plants perform a basic function of maintaining human life through the supply of oxygen, food, and basic ecosystem services. Keeping them in good health is hence important not just for agriculture but also for environmental resilience and food security. Detection of plant species and health status at an early stage, however, is challenging. Current methods depend on human observation or consultant expertise, which are time-consuming, subjective, and in many cases inaccessible in rural areas [6] or resource-constrained regions. These constraints prevent timely intervention, both for small scale gardeners handling varied crops and large-scale farmers handling vast cropland.

Even in recent studies, digital solutions for monitoring plants have been introduced, yet most of them are limited to narrow capabilities such as either identification of a species or diagnosis of diseases [1] [4]. Little work has been reported in combining both tasks within an integrated system, and most solutions available are not accessible, hence, of limited utility to non-technical users [3] [7]. These shortcomings underscore the necessity for a machine-centered, image-based method that can identify plant species as well as determine their health status at the same time in a consistent, efficient, and convenient manner.

III. RELATED WORK

Over the past decade, researchers have increasingly turned to computer vision and machine learning as promising tools for plant health monitoring. Early work in this area often relied on traditional image processing techniques, where handcrafted features such as color, texture, and shape were extracted from leaf images and then fed into classifiers like Support Vector Machines (SVMs) or Random Forests [9]. Although these approaches yielded valuable information, they were constrained in their performance by their reliance on expertly crafted features and lack of ability to generalize to a wide variety of plant species and conditions.

The emergence of deep learning, and more specifically convolutional neural networks (CNNs), has revolutionized plant disease detection [10]. With models that have been trained using large-scale datasets such as PlantVillage, high levels of accuracy are seen in identifying diseased versus healthy leaves. For example, experiments employing CNN structures such as AlexNet, VGGNet, and ResNet have attained accuracies frequently above 90% under controlled conditions. These findings unveil the promise of machine based monitoring of plant health, particularly when large and nicely labeled data are present [1] [6].

Nonetheless, despite these gains, there are still various gaps present. Most current research is based on disease classification and ignores the equally vital process of species identification. Others have high accuracy within laboratory settings but lose this under real-world conditions of varying light, ambient noise, and image quality. Also, the majority of systems are formulated for research and not for end-user use. Farmers and gardeners, the most likely beneficiaries, do not frequently have access to accurate and simple tools.

This work takes advantage of these discoveries by suggesting a unified system for plant species recognition and health categorization. In contrast to previous works that segregate these processes, our model brings them under one process. Secondly, by taking advantage of popular Python libraries and minimal frameworks such as Streamlit, the system seeks to be

more user-friendly to non-experts. This union of technical efficacy and practical usability is what makes current research distinct from previous works.

IV. METHODOLOGY

1. Data Collection

The data used in this research work was downloaded from the Plant Pathology Kaggle competition data set, and it consists of labeled apple leaf images classified into four classes: healthy, scab, rust, and multiple diseases. The data set included a train set (train.csv), test set (test.csv), and sample submission file (sample_submission.csv). Every image is identified by a unique image_id.

2. Data Preprocessing

Images were read in using OpenCV and normalized into RGB format to ensure consistency. For exploratory analysis, 100 images were randomly sampled as a subset. Channel-wise (Red, Green, Blue) distributions were plotted to examine the mean pixel intensity across images. Histograms and pie charts were also produced to display the frequency distribution of each disease class. To improve interpretability, images were displayed with varying class labels (healthy, scab, rust, multiple diseases). The categorical distribution of target labels was investigated with parallel category plots and histograms.

3. Data Augmentation

In order to prevent overfitting and improve generalization, data augmentation techniques were applied. Using the Keras ImageDataGenerator, images were rescaled and transformed through:

- Horizontal and vertical flips
- Random rotations
- Zoom and shift operations

This step increased the variability of the training data, allowing the model to learn robust disease-related features.

4. Model Architecture

A Convolutional Neural Network (CNN) was implemented using TensorFlow/Keras [5]. The model achitecture contains:

- For feature extraction convolutional layers with ReLU
- Maxpooling layers is used for dimensionality reduction.
- Dropout layers that will reduce overfitting
- Fully connected dense layer that nakes up the output layer.

The final output layer used a softmax activation function to classify images into the four categories.

5. Model Training

The model was trained for 20 epochs at a batch size of 32. The Adam optimizer was employed for optimization, and the categorical cross-entropy loss function was utilized given the multi-class classification problem at hand. Seeds for NumPy and TensorFlow were fixed to ensure reproducibility.

6. Evaluation Metrics

Performance was tracked in terms of accuracy on both training and validation sets. Visualization aids like training vs. validation accuracy and loss plots were employed to examine convergence and overfitting behaviors.

7. Prediction and Submission

Following training, the model was applied to make predictions on the test dataset layer by layer. Predicted probabilities for every class were saved in a .csv file.

V. EXPERIMENTAL SETUP

The model was developed in Python using TensorFlow and Keras, supported by add-on libraries such as OpenCV, Pillow, NumPy, Pandas, and Matplotlib. A simple web-based interface was designed for convenience of deployment and user input using Streamlit. All experiments were executed on a Windows 11 machine with an Intel Core i5 processor and 16 GB RAM, which supplied the required computational power for training the model.

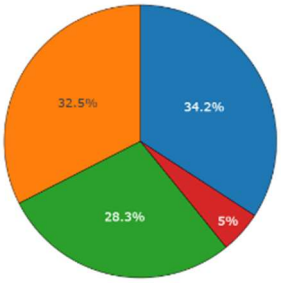
Plant Pathology 2020 dataset [2] by Kaggle was utilized, which contains apple leaf images classified into four categories: healthy, multiple diseases, rust, and scab. Images were resized to 224 × 224 pixels and normalized. Data augmentation techniques [6] such as random flipping, rotation, and zooming were utilized in order to improve the generalization of the model.

A Convolutional Neural Network (CNN) was used with the Adam optimizer and learning rate of 0.001, categorical crossentropy loss, and ReLU activation of the hidden layers and a Softmax output layer. It was trained for 20 epochs with the batch size of 32 to avoid overfitting. Model performance was monitored on the training set and validation set through accuracy and loss.

Finally, the model was integrated into the Streamlit application so that users could upload pictures of leaves and receive predictions regarding plant species and health status in real-time. This made the system available not only to researchers but also to farmers, gardeners, and hobbyists.

VI. RESULTS

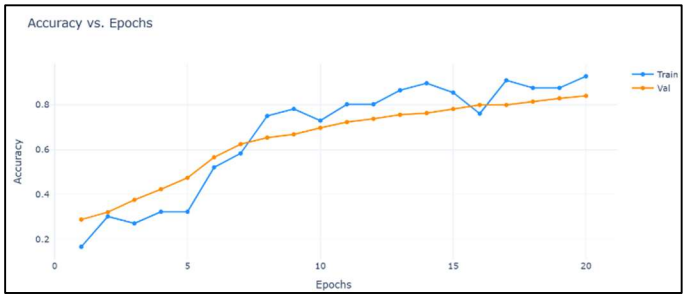
The system was tested with the Plant Pathology dataset, which comprises four classes: rust, scab, healthy, and multiple diseases. The dataset distribution indicates that most samples from the rust (34.2%), scab (32.5%), and healthy (28.3%) classes; the minority class is multiple diseases (5%). This is inefficient for the model in identifying the underrepresented categories.



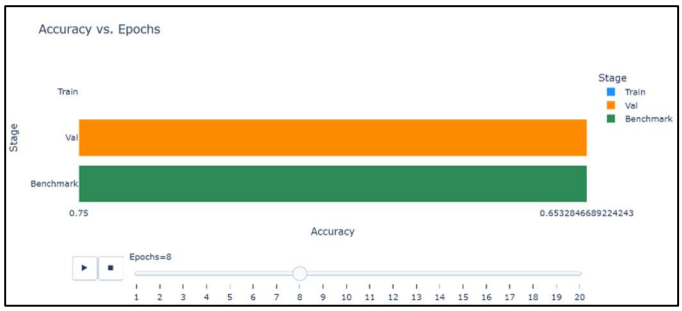
For solving the classification problem, transfer learning was employed with DenseNet121 as the backbone model. The architecture integrates DenseNet121 with a Global Average Pooling layer and a dense output layer and has around 7 million trainable parameters. Due to efficient reuse of features, small size, and established success in image-based classification problems, it was the best selection for the model.

Layer (type)	Output Shape	Param #
densenet121 (Functional)	(None, 7, 7, 1024)	7,037,504
global_average_pooling2d (GlobalAveragePooling2D)	(None, 1024)	0
dense (Dense)	(None, 4)	4,100

Training outcomes show high performance. As is evident in the figure, both training and validation accuracies improved consistently across epochs, with training accuracy breaking 90% and validation accuracy leveling off at approximately 85% by the 20th epoch. As observed, the validation curve closely tracked the training curve, indicating little overfitting and high normalization.



The accuracy of validation was compared to a baseline of 75% to further examine the performance and accuracy of the model. The model surpassed this baseline easily with final validation accuracy being more than 83%. This shows the efficiency of the model and how well transfer learning copes with plant disease detection, even for minor imbalance.



Although the results are amazing, but still there are certain limitations. The minority class (multiple diseases) showed poor performance, reflecting the lack of sufficient training examples. Various improvements can be made in future that could involve the use of data augmentation, synthetic oversampling, or weighted loss functions to better balance the classification process. Another limitation lies in external conditions such as varying lighting, image quality, or background clutter, which may reduce performance in real-world settings compared to the relatively controlled dataset.

Despite these issues, the model proposed is a distinct improvement compared to conventional hand-tuned feature approaches as well as baseline CNNs. Earlier papers like AlexNet, VGG, and ResNet [10] achieve accuracies over 90% on large and well-classified datasets like PlantVillage, but their performance usually declines under real-world scenarios. Conversely, our model combines species identification and disease diagnosis within one easy-to-use pipeline that is more appropriate for hobbyists, gardeners, and farmers who need uncomplicated yet reliable tools.

Overall, the DenseNet121-based system shows its promise as affordable, real-time solution for plant disease monitoring. The results verify the applicability of leveraging deep learning models in real-world agricultural scenarios, showing the way for more intelligent tools that can power both sustainability and food security.

VI. CONCLUSION AND FUTURE WORK

This work displayed a solution based on machine learning for recognizing plant species and their health status. Using convolutional neural networks learned on Plant Pathology dataset, the system revealed its potential to make expert-knowledge-intensive processes and manual inspection redundant. Normalization and image augmentation preprocessing techniques enhanced robustness, while Python library implementation guaranteed flexibility and ease of deployment. The system plays a vital role in helping make plant monitoring more scalable and accessible to plant enthusiasts, gardeners, and farmers who don't have technical expertise.

Despite with the encouraging results, the work has a number of limitations [2] [3]. Datasets were mainly collected in

controlled environments, which are more likely to restrict performance in open-world settings where lighting, background noise, and image quality differ. Classification of the plants was limited to distinguishing species and healthy from unhealthy plants, without differentiating diseases or severity levels due to constrain availability of data. The prototype was also tested on relatively constrained hardware and has not yet been proven in field conditions.

Various constraints can be overcome in future works by increasing the dataset with further diverse real-world images and by using transfer learning methods in order to enhance generalization and thus, increasing the efficacy of the model. Enabling multi-class classification of diseases would enable the system to identify certain plant ailments with greater specificity [6] [8]. Deployment on mobile devices or as an edge-computing solution might further increase accessibility to farmers in rural locations, many of whom are not even able to afford mobile phones. Lastly, incorporating user feedback loops may assist in fine-tuning the model on an ongoing basis and also tailoring it to local farming conditions, therefore, training the model on user-feedback for future improvements.

VII. REFERENCES

[1] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Frontiers in Plant Science*, vol. 7, p. 1419, Apr. 2016.

[2] R. Thapa, N. Snavely, S. Belongie, and A. Khan, "The Plant Pathology 2020 Challenge Dataset to Classify Foliar Disease of Apples," *arXiv preprint*, Apr. 2020.

[3] D. Singh et al., "PlantDoc: A Dataset for Visual Plant Disease Detection," *arXiv preprint*, Nov. 2019.

[4] M. Brahimi, S. Mahmoudi, K. Boukhalfa, and A. Moussaoui, "Deep Interpretable Architecture for Plant Diseases Classification," *arXiv preprint*, May 2019.

[5] I. Bouchnafa and M. Amnai, "Grape and Apple Plant Diseases Detection Using Enhanced DenseNet121 Based Convolutional Neural Network," in *Advances in Intelligent System and Smart Technologies (I2ST 2023)*, Lecture Notes in Networks and Systems, vol. 826, Cham, 2024, pp. 213–226.

[6] M. R. Mahum et al., "Plant Disease Detection and Classification by Deep Learning—A Review," *IEEE Access*, vol. 9, pp. 56683–56698, 2021.

[7] S. Yadav, U. Thakur, R. Saxena, V. Pal, V. Bhateja, and J. C.-W. Lin, "AFD-Net: Apple Foliar Disease Multi Classification Using Deep Learning on Plant Pathology Dataset," *Plant and Soil*, vol. 477, pp. 1–17, 2022.

[8] *Transfer learning*, Wikipedia, accessed Aug. 2025.

[9] E. C. Too, L. Yujian, S. Njuki, and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," Computers and Electronics in Agriculture, vol. 161, pp. 272–279, 2019.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems 25 (NeurIPS), 2012, pp. 1097–1105.