

A Machine Learning Approach to Crop Disease Detection using Image Processing and Feature Extraction

Krishna Iyer¹, Romit Sabnis², Arnav Tripathi³, Akhil Nair⁴ and Kiran V. Ajetao⁵
¹⁻⁵K. J. Somaiya College of Engineering, Somaiya Vidyavihar University, Mumbai, India
Email: iyer.v@somaiya.edu, romit.sabnis, arnav.tripathi, akhil.mn, kiranjetrao}@somaiya.edu

Abstract—One of the major issues faced in agricultural fields worldwide is detection of plant illness. Identification of such diseases in the early stages is crucial to improving overall productivity and reducing production loss. Farms that contain a variety of crops cannot be tracked manually for diseases as it requires skilled labor and it is resource intensive. The thesis introduces an accurate and effective method for detection of plant illness which uses Convolutional Neural Network (CNN) architecture (VGG style). The proposed system can detect 26 different diseases of 11 plants with 98% accuracy.

Index Terms—Crop Disease Detection, Image Processing, Machine Learning, Agriculture, Deep Learning.

I. INTRODUCTION

Farming continues to be the key contributor to the Indian economy and accounts for around 19 per cent of the GDP. The identification of divergent crop illness holds significant importance throughout the agricultural sector. Plant disease prevalence affects agricultural production and economic stability, posing a danger to global food security. In ancient times, farmers used to manually monitor the fields for detecting diseases. The efficacy and precision of traditional plant disease detection techniques are frequently compromised, hence calling for a transition to more sophisticated, technologically driven approaches.

Recent Advancements in image processing technologies have revolutionized the way agricultural practices are conducted. With a VGG-style Convolutional Neural Network (CNN) and an extensive image dataset, this project seeks to improve plant disease detection speed, accuracy, and scalability in several agricultural domains. Machine learning algorithms significantly enhance the capability of image processing systems by enabling them to learn from a vast amount of data and improve their diagnostic precision over time. These techniques aid in identifying specific symptoms associated with various diseases and are critical in designing robust detection systems that can operate efficiently in real-world agricultural settings.

We have observed that a wide variety of algorithms and methodologies are used in all the papers mentioned above. We have used VGG style CNN and have trained our model to detect 26 unique diseases in 11 crops with an accuracy of 98.20% by taking insights from the first paper which has used CNN architectures like AlexNet and GoogleNet and has achieved an accuracy of 99.35% which is trained to classify 26 diseases across 14 plant species.

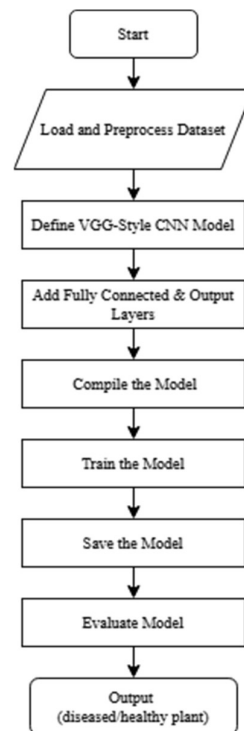
Using our method we have achieved 2% more accuracy than the fourth paper which also uses CNN architecture, and we have achieved 25% higher accuracy than the ones which have used hyperspectral imagery. The thesis is organized as follows: The Literature Review section explores widely available research on crop disease detection techniques and related methodologies. Methodology provides a general review of the dataset, detailing data acquisition, processing methods, feature extraction approaches, and a comparative analysis of various research papers in the field. The Model Evaluation section discusses the training and testing process, including epoch or iteration details. Finally, the Results section presents the outcomes of the model's performance, followed by References to all consulted sources.

II. METHODOLOGY

This segment of the thesis provides a thorough comprehension of the progress made and the overall distinct operations. In our system we acquire images which are captured using a camera. The k-means clustering partitioning method is implemented for extracting features from different plant species and they are classified using CNN.

Starting with the Flowchart which will explain how our model is made and how the overall code works. The next part is the dataset chosen for training the NN; the dataset is represented in tabular format in the order in which it is chose for training the NN. Along with the dataset are the performance metrics which we get while running and testing the code for each particular class in the dataset, it includes the classes and their accuracy in our model. Following this is the data preprocessing, which is the resizing, normalization of the images etc. Feature extraction in which we use the K nearest neighbour algorithm. The model evaluation step is then explained in detail. After this is the training and testing section which goes in detail about our CNN model, how it is trained and the testing done to verify the results etc. Finally, we have the block diagram which explains how the data given to our model is handled.

A. Flowchart



Plant images are loaded from a dataset containing labelled images for each crop disease. These images are resized to a consistent shape (e.g., 128x128 pixels) to match the programmed input size of the model. Images are augmented using operations such as rotation, flips etc. to create varied versions of the images.

The 'VGG-style' CNN model is built with a series of convolutional blocks. Two or three Conv2d layers form a block, each layer including small (3x3) filters followed by a 'ReLU' activation layer to inculcate non-linearity.

Flattened output obtained from the convolutional layer is converted to 1D vector and is then passed to dense layers with ReLU activation to learn complex patterns. Dropout layers help reduce overfitting. The final SoftMax layer outputs probabilities, enabling classification into disease categories. NN Model is trained implementing Adam optimizer inculcating gradient efficiency function, categorical cross-entropy as a failure rate to measure accuracy, and accuracy as a measure of efficiency. The training loop runs for a set number of epochs. During each epoch, mini batches from the training set pass through the model to generate predictions, with loss calculated and backpropagated to update weights. The number of epochs used to train this particular model is 10. Saving the model at this point allows reusing it without retraining, enabling the system to retain a version of the model that meets performance requirements. The saved model is what will be used in the further applications. The model is evaluated on a separate test set containing unseen images. This step calculates the final test accuracy and additional metrics, such as accuracy and true positive rate, which provides familiarity regarding how well the NN recognises the new data. Higher accuracy here indicates a successful plant disease detection model.

Output received from the model.

B. Dataset

A dataset was gathered consisting of images of crop leaves in both good and bad health. Overall dataset consists of 87 thousand 'RGB' leaf images in good health and in bad health which are categorized into thirty-eight distinct diseases from Plant Village.

TABLE I. DATASET SPECIFICATIONS

Plant	Disease	No. of images
Apple	Apple-scab	2019
	Black-rot	1991
	Cedar-apple-rust	1763
Blueberry	Good health	2013
	Good health	1819
Cherry (including sour)	Powdery-mildew	1686
	Good health	1830
Corn (maize)	Cercospora-leaf-spot-Gray-leaf-spot	1647
	Common-rust	1912
	Northern-Leaf-Blight	1913
	Good health	1864
Grape	Black-rot	1893
	Esca-(Black-Measles)	1925
	Leaf-blight (Isariopsis-Leaf- Spot)	1728
Orange	Good health	1697
	Haunglongbing -(Citrus-greening)	2014
Peach	Bacterial-spot	1842
	Good health	1732
Pepper, bell	Bacterial-spot	1917
	Good health	1992
Potato	Early-blight	1943
	Late-blight	1943
	Good health	1829
Raspberry	Good health	1785
Soyabean	Good health	2026
Squash	Powdery-mildew	1740
Strawberry	Leaf-scorch	1778
	Good health	1828
Tomato	Bacterial-spot	1706
	Early-blight	1924
	Late-blight	1855
	Leaf-Mold	1886

The neural network (NN) was implemented using a public dataset for crop leaf disease detection called Plant Village curated by Sharada P. Mohanty. [1]. Table I represents classes of images present in our dataset along with the quantity of images present in every class.

The second tabular data represents a classification report containing the detailed performance metrics of the CNN model, which was trained to classify diseases in different plant species. The evaluation is based on four

key metrics: precision, recall, F1-score, and support, computed for every disease category as well as for healthy crops. When the performance across all groups is collected, the overall model accuracy, true positive rate and F1-score is estimated to be 97 %, which supports a total of 17572 images. These results confirm this model's performance in distinguishing healthy crops from diseased crops with high specificity and recall for each category.

TABLE II. PERFORMANCE PARAMETERS

Plant-Disease	Precision(%)	Recall(%)	F1-Score (%)
Apple-Scab	97	95	96
A-Black-rot	99	100	99
Apple---Cedar-apple-rust	98	97	97
Apple-- Good health	97	97	97
Blueberry-- Good health	99	97	98
Cherry-(including-sour)--Powdery-mildew	100	99	99
Cherry-(including-sour)--Good health	97	99	98
Corn-(maize)---Cercospora- ---leaf-spot Gray-leaf-spot	98	88	92
Corn-(maize)---Common-rust	99	100	99
Com-(maize)---Northern-Leaf-Blight	93	97	95
Corn-(maize)-- Good health	100	98	99
Grape---Black-rot	96	98	97
Grape---Esca-(Black-Measles)	98	99	98
Grape-Leaf-blight- (Isariopsis-Leaf-Spot)	100	98	99
Grape--- Good health	99	100	100
Orange---Haunglongbing-(Citrus-greening)	98	99	99
Peach---Bacterial-spot	97	97	97
Peach--- Good health	96	100	98
Pepper---bell---Bacterial-spot	94	97	96
Pepper---bell--- Good health	99	91	95
Potato---Early-blight	99	97	98
Potato---Late-blight	88	99	93
Potato--- Good health	99	96	97
Raspberry--- Good health	99	99	99
Soybean--- Good health	99	98	98
Squash---Powdery-mildew	99	99	99
Strawberry---Leaf-scorch	96	99	97
Strawberry--- Good health	100	100	100
Tomato---Bacterial-spot	100	96	98
Tomato---Early-blight	92	90	91
Tomato---Late-blight	97	81	88
Tomato---Leaf-Mold	96	91	93
Tomato---Septoria-leaf-spot	78	97	87
Tomato---Spider-mites Two-spotted-spider-mite	96	98	97
Tomato---Target-Spot	95	95	95
Tomato---Tomato-Yellow-Leaf-Curl-Virus	99	99	99
Tomato---Tomato-mosaic-virus	99	98	99

C. Data preprocessing

It is an important task which includes resizing images to a constant dimension and normalizing of pixel values to aid in the convergence of the model during training.

These preprocessing steps ensure that the data input is in a suitable form for efficient processing by the neural network.

For obtaining results up to the mark, noise due to interference should be discarded before extracting required features. In the initial stage, the RGB image is transformed to grayscale after which image undergoes Gaussian filtering for image refinement.

D. Feature extraction

K-nearest neighbor model is implemented to extract various characteristics from the dataset. Various characteristics like colour, texture, shape, and size of the crop and any visual symptoms or patterns associated with diseases.

E. Model Evaluation

The model is tested and evaluated with the test dataset to measure its accuracy and performance in detecting various crop illness. Data obtained from performance metrics represents the effectiveness of the model developed for classifying plant diseases across various crops. These metrics help evaluate the accuracy, reliability and applicability of the model in real agricultural practice. To provide a full performance assessment, there is a need for metrics assessment as shown in Table II.

F. Training & Testing

The Convolutional Neural Network (CNN) implemented in the training model resembles a VGG-like network, specifically like VGG16 or VGG19 architectures. The 'CNN' Model is trained using TensorFlow's Keras Library, version 2.17.0. The model is sequentially organized and consists of several layers:

- **Input Layer & Initial Convolutional Blocks** The 'CNN' Model starts with a Two-Dimensional Convolutional layer consisting of 32 filters, a kernel of size 3x3 with 'ReLU' activation and an input shape of 128x128x3, which indicates that the input image. Another convolutional layer with the same parameters follows, and then a MaxPooling-Two- Dimensional layer with a pool size of 2x2 and strides of 2x2 is implemented to decrease the spatial layout of the feature mappings of the input shape.
- **Subsequent Convolutional Blocks:** The model continues with multiple convolutional blocks, each consisting of two Conv2D layers and a MaxPooling2D layer. The count of the filter doubles with every block, going from 64 to 512 filters. This progression enables the network to learn increasingly complex features while reducing the spatial dimensions through max pooling.
- **Dropout Layers:** Dropout layers are included with dropout rates of 0.25 and 0.4 after certain layers pertaining to the input shape are prevented from overfitting by arbitrarily setting a few of the units to 0 while testing phase.
- **Flattening and Dense Layers:** After the convolutional blocks, the network the three-dimensional feature maps into a one- dimensional vector using a Flatten layer, which is then followed by a denser layer containing 1500 units with 'ReLU' activation. 1500 units and 'ReLU' activation. This layer is designed to capture the higher-level features processed and learned by the convolutional layers.
- **Output Layer:** The model concludes with a dense output layer consisting of 38 units with 'softmax' activation, indicating that the model is performing multi-class classification with 38 possible classes.
- **Compilation:** Finally, the model uses Adam optimizer, a cross-entropy categorical loss function as the evaluation metric with a loss of 0.0542, validation loss of 0.1116 and a validation accuracy of 96.76%. This setup is typical for classification tasks where the output is one of several classes.

G. Schematic

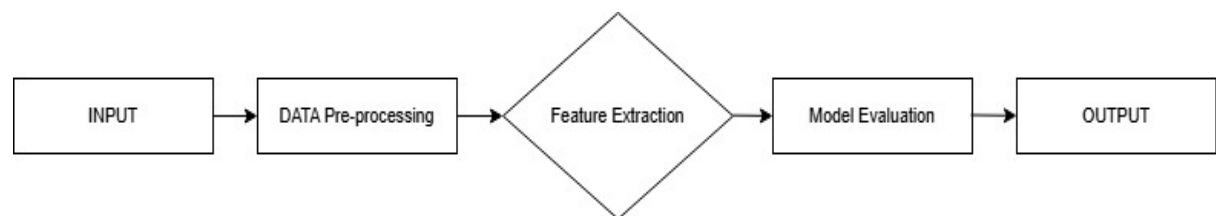


Figure 1. Schematic

Figure 1 shows the flow of our image processing model. Input image is fed by the user to the system which undergoes pre-processing, and the required features are extracted, and the model is evaluated on the parameters and generates the final output.

Firstly, the model is built using a series of convolutional blocks followed by addition of output layers and the model is configured with the Adam optimizer.

The training of the model is done with 10 epochs using batches of images from the dataset and is evaluated on a separate test set consisting of unseen images with an accuracy and validation accuracy percentage of 98.26% and 96.74% respectively.

III. RESULT

A. Confusion Matrix

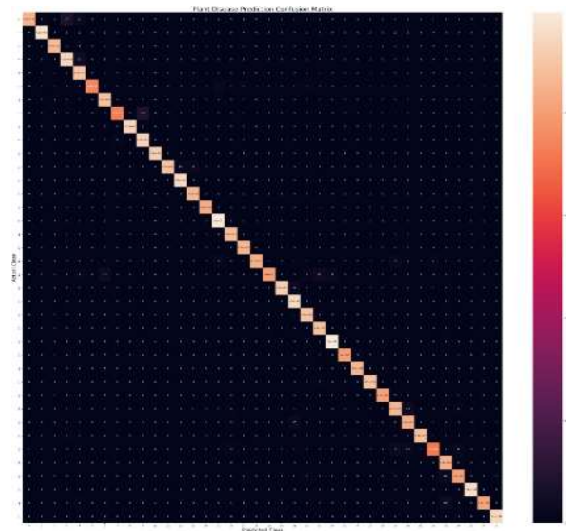


Figure 2. Confusion Matrix

The structured collection of data is categorized into training group ($\approx 80\%$) for training the NN and testing group ($\approx 20\%$) for validation. The confusion matrix reveals the accuracy for each class of the model in predicting plant diseases, as indicated by the dominance of lighter shades along the diagonal, signifying correct predictions, with minimal misclassification evident by the darker off-diagonal elements. With the help of confusion matrices, false positives, false negatives, and true predictions can be analysed.

B. Input Images vs Predictions

After training, the model is tested using sample images of plant diseases. Table III shows the input sample image vs the prediction made by our model. The leaf images taken at random were Potato-early-blight, Tomato-septoria-leaf-spot and Tomato-yellow-leaf-curl virus. The model successfully predicted all the diseases that were tested.

TABLE III. TEST RESULTS

Input Image	Test image		
			
Predictions	Disease Name: Potato Early blight	Disease Name: Tomato Septoria leaf spot	Disease Name: Tomato Tomato Yellow Leaf Curl Virus

With respect to Table III, our model is able to predict the disease and classify it accurately. Table III is a small collection of test images used to check the accuracy. These test results help us verify the working of the model and supporting its potential for future deployment for user applications.

C. Homepage of Website

We have developed a user-friendly website using Streamlit open-source python framework wherein the user can upload image of the plant's leaf and our model will predict whether the plant is diseased or healthy. The website hosts 3 total webpages, one being the homepage, second being an about page with all the information about our project from the dataset to the accuracy. The final page being the tool that users can access our model through.

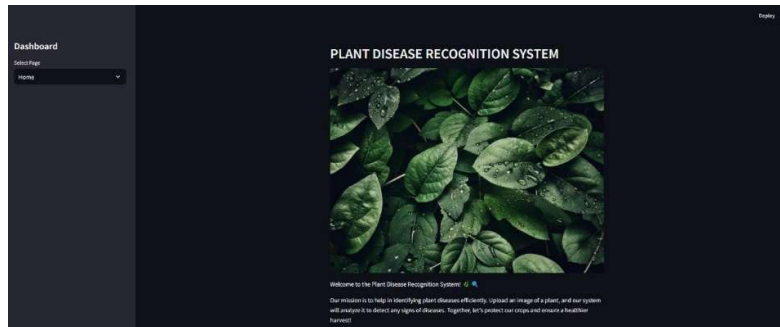


Figure 3. Basic layout of website

Figure 3 shows the homepage of our website made using 'streamlit'. Users can switch to the Disease detection section using the dropdown menu in the 'top left' column and upload the leaf image and get desired results. The users can verify the uploaded images in the site while the disease or plant is being identified.

IV. CONCLUSION

The importance of early plant disease detection, particularly across a wide range of species, cannot be overstated. Our machine learning- based system offers a proactive solution for disease management by accurately classifying plant images into one of 26 of 11 plants. This fast detection strategy plays a crucial role in minimizing crop losses and promoting agricultural sustainability. By identifying diseases at an initial stage, farm workers execute swift action, which includes applying appropriate treatments and adjusting cultivation practices, to avoid the transmission and increase of diseases. This leads to healthier crops, improved yields, and enhanced overall crop quality.

REFERENCES

- [1] Mohanty SP, Hughes DP, Salathé M. Using Deep Learning for Image-Based Plant Disease Detection. *Front Plant Sci.* 2016 Sep 22;7:1419. doi: 10.3389/fpls.2016.01419. PMID: 27713752; PMCID: PMC5032846.
- [2] Oo, Y.M.; Htun, N.C. Plant Leaf Disease Detection and Classification using Image Processing. 2018, 5, 516–523.
- [3] Saleem MH, Khanchi S, Potgieter J, Arif KM. Image-Based Plant Disease Identification by Deep Learning Meta-Architectures. *Plants.* 2020; 9(11):1451.
- [4] Sladojevic, S., Arsenovic, M., Anderla, A., & Culibrk, D. (2016). Deep neural networksbased recognition of plant diseases by leaf image classification. *Computational intelligenceand neuroscience*, 2016.
- [5] Kuswidiyanto, Lukas Wiku, Hyun-Ho Noh, and Xiongze Han. 2022. "Plant Disease Diagnosis Using Deep Learning Based on Aerial Hyperspectral Images: A Review" *Remote Sensing* 14, no. 23: 6031.
- [6] Kumar, Surender & Kaur, Rupinder. (2015). Plant Disease Detection using Image Processing- A Review. *International Journal of Computer Applications.* 124. 6-9. 10.5120/ijca2015905789.
- [7] Kulkarni, Pranesh & Karwande, Atharva & Kolhe, Tejas & Kamble, Soham & Joshi, Akshay & Wyawahare, Medha. (2021). Plant Disease Detection Using Image Processing and Machine Learning. 10.48550/arXiv.2106.10698.
- [8] Garcia-Ruiz, F., Sankaran, S., Maja, J. M., Lee, W. S., Rasmussen, J., and Ehsani R. (2013). Comparison of two aerial imaging platforms for identification of huanglongbing-infected citrus trees. *Comput. Electron. Agric.* 91, 106–115. doi: 10.1016/j.compag.2012.12.002
- [9] Rousseau, D., Lucidarme, P., Bertheloot, J., Morel, P., & tienne Belin, A. (2012). On the use of depth camera for 3D phenotyping of entire plants.
- [10] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In 2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05) (Vol. 1, pp. 886-893). Ieee.
- [11] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.
- [12] Everingham, M., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88, 303-338.

- [13] Hernández-Rabadán, D. L., Ramos-Quintana, F., & Guerrero Juk, J. (2014). Integrating SOMs and a Bayesian classifier for segmenting diseased plants in uncontrolled environments. *The scientific world journal*, 2014(1), 214674.
- [14] Patil, P., Tadval, P., Tanpure, R., Yamgar, P., & Zarkar, S. (2023). Smart Farming Using Deep Learning.
- [15] Hussein, O. A., & Mahdi, M. S. (2023). REVIEW ON DETECTION OF RICE PLANT LEAVES DISEASES USING DATA AUGMENTATION AND TRANSFER LEARNING TECHNIQUES. *Iraqi Journal for Computers and Informatics*, 49(1), 30-40.
- [16] Awasthi, R. K., & Singh, S. (2023). An Overview of Machine Learning Methods for the Detection of Diseases in Rice Plants in Agricultural Research.
- [17] Ahmed, I., & Yadav, P. K. (2023). Plant disease detection using machine learning approaches. *Expert Systems*, 40(5), e13136.
- [18] Abdullah, H. M., Mohana, N. T., Khan, B. M., Ahmed, S. M., Hossain, M., Islam, K. S., ... & Ahamed, T. (2023). Present and future scopes and challenges of plant pest and disease (P&D) monitoring: Remote sensing, image processing, and artificial intelligence perspectives. *Remote Sensing Applications: Society and Environment*, 32, 100996.
- [19] Moupojou, E., Tagne, A., Retraint, F., Tadonkemwa, A., Wilfried, D., Tapamo, H., & Nkenlifack, M. (2023). FieldPlant: A dataset of field plant images for plant disease detection and classification with deep learning. *IEEE Access*, 11, 35398-35410.
- [20] Kotwal, J., Kashyap, R., & Pathan, S. (2023). Agricultural plant diseases identification: From traditional approach to deep learning. *Materials Today: Proceedings*, 80, 344-356.