iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** X **Month of publication:** October 2024

DOI: https://doi.org/10.22214/ijraset.2024.64643

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue X Oct 2024- Available at www.ijraset.com

Plant Disease Detection Using Machine Learning

Vaishali Rajput[1], Aditi Dharmadhikari[2], Aditya Bhagat[3], Aditri Sivakumar[3], Aditya Kulkarni[4], Aditya Nath Deepak[5], Nirwani Adhau[6]
Department of Engineering, Sciences and Humanities (DESH), Vishwakarma Institute of Technology, Pune, 411037, Maharashtra, India

Abstract: ***In this project, we have developed a plant disease detection model using a machine learning approach on a dataset named "Plant Disease Expert" which is available on Kaggle. The dataset contains 199,611 images across 58 subdirectories. The splitting of the dataset in training, validation and testing is done in ratio 90:5:5. The data preprocessing included resizing the images, normalizing pixel values and applying augmentation techniques. We have used different libraries throughout the project such as NumPy, Pandas, Operating system, Time, Matplotlib, OpenCV, Shutil, TensorFlow, Keras, Seaborn, etc. EfficientNetB3 is used as the base model. EfficientNetB3 is a convolutional neural network architecture developed by Google Brain researchers [1]. It is known for its efficiency and effectiveness in image classification tasks. Additional custom layers such as max pooling, batch normalization, dense layers, and activation functions are used to optimize the performance. The adamax optimizer and categorical cross-entropy were used during training. The model was trained for 10 epochs with a batch size of 20 and a learning rate of 0.01. For evaluation, classification report, accuracy, precision, recall, F1-score and support were used. The model achieved an accuracy of 98.76% [1].***

Keywords: Machine Learning, Computer Vision and Image Analysis, Plant disease detection, EfficientNetB3, Python programming, Convolutional Neural Network.

I. INTRODUCTION

Detection of plant diseases is an essential research area for agricultural sustainability and food security. Plants lost millions of people due to reduced crop yield, quality, and overall productivity. Agriculture plays a significant role in the Indian economy and accounts for 15-20% of the GDP while employing about half of the workforce. Early detection of the disease gives the required timely intervention, thereby reducing crop losses and making more sustainable activity. Since traditional methods, such as chemical pesticides, damage ecosystems and health, modern technology-based solutions are more called for [1][3][7]. This research focuses on the aspect of plant disease detection, a very essential factor when it comes to agricultural sustainability and food security, especially for India, where agriculture forms a significant percentage of the economy and employment. Early detection of plant diseases helps in saving crop losses and reducing the dependency on harmful chemical pesticides. Advanced technologies, such as machine learning, image processing, and IoT devices, bring about new means of real-time monitoring and early intervention for prevention of annual losses up to $220 billion globally in agriculture. Such practices can potentially minimize the damaging impact of the disease on crops, build resilience in crops, and help in the adoption of more sustainable forms of farming, ensuring better food security and environmental conservation [4][8].

II. LITERATURE REVIEW

The focus is in an area of critical importance to plant disease detection, which holds major implications for agriculture productivity and food security in the world. Plant diseases often start from their leaves and quickly spread, significantly lowering yield and quality. Improvements in losses or agricultural output are said to be positive results of early detection of such diseases. It compares various machine learning and deep learning methods such as CNN, SVM, Random Forest, and EfficientNet-B3 that can reach up to 98% precision in the recognition of diseases. The technique of image processing is significant for application - techniques like converting to grayscale, feature extraction using GLCM, and morphological transformations to refine the detection models. To handle problems of intra-class variability, limited data, and mainly real-time implementation issues, it emphasizes pre-trained models, transfer learning, and adaptive augmentation. [1][2][3]. The techniques are highly dependent on large sets of RGB images of both healthy and diseased leaves, relying on features such as contrast, dissimilarity, and homogeneity to classify. Improved speed and accuracy in disease identification support proactive management practices, reductions in dependency on chemicals, optimization of resource use, and sustainable farming practice-all factors critical to advancing early disease detection in a more secure food system globally. [6][7][8]

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue X Oct 2024- Available at www.ijraset.com

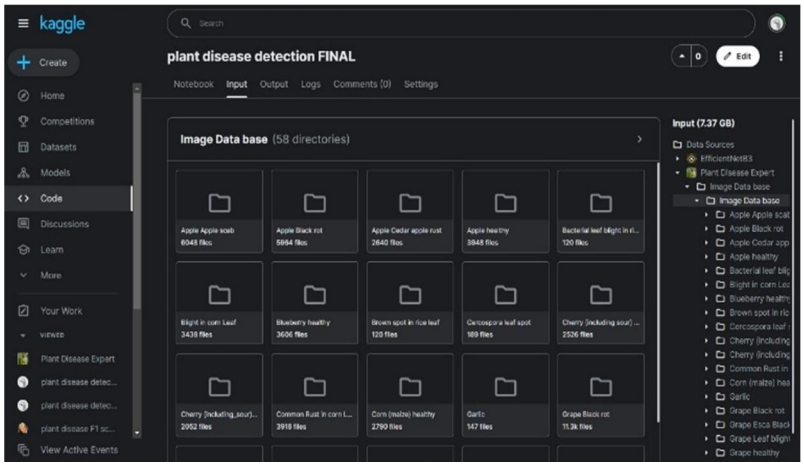
III. METHODOLOGY

A. *Features*

1) *Platform Used:* Kaggle: Kaggle is an online platform for data scientists and machine learning enthusiast. It contains competitions, datasets, and cloud-based tools meant to help someone progress in one's skillset. It also gives easy access to tutorials, free GPUs, and interactive courses, among other facilities. This encourages a lot of interaction among people in the community and learning from one another. [1].
2) *Dataset:* Dataset Introduction: We made use of the dataset "Plant Disease Expert" from Kaggle for developing our machine learning model for plant disease detection. The dataset serves as an enormous collection of images intended to help in identification and classification of various plant diseases, thus making it very useful in the development of robust models using machine learning. The total size is 7.37 GB, but we used the image database. There were 3-4 subdirectories. However, we have only used one main, which is Niche database. [1].

B. *Dataset Structure and Contents*

It has thousands of images categorized into classes, such as various plants which are healthy and diseased leaves. Images are stored in a common format like JPEG, PNG with varied resolutions. There are also separate subsets for the training subset, validation subset, and testing subset that can be used to evaluate the model effectively. We used most of the species; however, some of them were apples, rice, blueberry, corn, cherry, potato, etc. [3].
Total images in training dataset: 179649 images.
Total images for testing: 9981 images.
Total images for validation: 9981 images.
Number of Images: 199611images.
Classes: Multiple classes including healthy and various disease conditions.
Image Format: JPEG, PNG.
Resolution: Varying, typically high-resolution images.



This image displays the 58 subdirectories of which some are shown.

1) *Libraries*

NumPy is the basic library package in the Python language and its main advantage is that it provides powerful tools to work with multidimensional arrays and matrices while Pandas, built upon it, implements high level data structures such as the DataFrame and the Series in order to manipulate and analyze data. Inside the module for operating systems of Python, you will be able to manage files and processes. The module Time deals with tasks related to time. Matplotlib is one of the most versatile libraries for creating visualizations, and OpenCV contains advanced functionalities of image and video processing. Shutil manages file operations, and metrics in Scikit-learn evaluates model performance using accuracy, F1 score, and so on. TensorFlow and Keras are those required deep learning efficiencies wherein a deep model can be very efficiently built, and Seaborn provides nicer-looking statistical graphics for better data visualization [3][12][13].

2) EfficientNetB3

This architecture was known to balance well between efficiency and accuracy in the classification of images. As part of the EfficientNet family, it aims for the optimization of both computational resources and model performance. In order for EfficientNetB3 to find this balance, it utilized a technique called compound scaling, whereby its network width, depth, and resolution are uniformly scaled. This ensures that growth throughout the dimensions of the network occurs simultaneously, leading to more meaningful representations of complex features at a cost of manageable computational costs. EfficientNetB3 also uses depth-wise separable convolutions. It reduces parameter size and computational complexity without incurring performance degradation. The model has been widely used in more extensive applications throughout computer vision than just classification, including object detection and semantic segmentation. With efficient design coupled with performance, EfficientNetB3 remains a cornerstone for deep learning in vision tasks. [6][9][10].

3) Layers

a) Max Pooling: It is termed 'max cooling' and represents the utmost achievable cooling performance in systems such as air conditioners and refrigeration. Such systems are central to comfort, food preservation, and industrial processes. Engineers strive to maximize cooling capacity while minimizing the use of energy and impact on the environment. This is what propels innovation toward efficiency and sustainability [13].

b) Batch Normalization: Batch normalization normalizes the inputs for each layer in a neural network, keeping its mean at zero and variance at one. It stabilizes, speeds up, and reduces overfitting while improving gradients and model performance overall by reducing internal covariate shift. Thus, it is very popularly used in deep learning applications [13].

c) Dense Layer: A dense layer, which is fully connected, connects every neuron to all neurons of the previous layer. By training, it will be possible to adjust weights. This helps a network learn complex patterns, an essential function for tasks like classification, regression, and sequence modeling in many neural network architectures [13].

d) ReLU Activation: ReLU or Rectified Linear Unit is a fundamental activation function in deep learning, providing both computation efficiency and non-linearity by giving the input if it's positive and zero otherwise. While this function has "dying ReLU" problems, it remains the default choice for many neural networks because of its excellent performance especially when used with He initialization to improve the outcome from training [11].

e) Dropout: Dropout is a kind of regularization technique that prevents overfitting through the training process by randomly dropping a percentage of neurons during training thus improving data representation. It applies to a hidden layer that has rates ranging from 0.2 to 0.5, thus modeling the training of several neural networks on different subsets of the data, thus generalizing more: applications for example in image recognition and natural language processing. [2][5][13].

4) Activation Function - SoftMax

SoftMax is a function used for multi-class classification. It converts raw scores or logits into probabilities such that their summation equals 1. The use of SoftMax implies an increase in larger input values and increases the confidence of the model in its predictions. It is typically used as the activation function for the output layer of a neural network in which the class with the highest probability corresponds to the predicted class [2][5][6][7].

5) Adamax Optimizer

Adamax can be regarded as a variant of the Adam optimizer to address the high-dimensional parameters and sparse gradients. It updates gradients using the first moment mean and scales weights by the infinity norm; thus it can provide better performance under complex models, especially in the applications of deep learning, like natural language processing and recurrent neural networks [13].

6) Categorial Cross Entropy

Categorical cross-entropy: It is the loss function that is used in multi-class classification and calculates the deviation between the true class distribution and probabilities. It penalizes larger discrepancies; thus, training, to minimize loss, is guided using such optimization algorithms as SGD. Such a diversity of applications helps produce accurate class probabilities from deep learning models [13].

7) Classification Report

A classification report summarizes the performance of a model in machine learning, including precision, recall, F1-score and support for each class. This allows practitioners to get an idea of what their models do better and worse at, ultimately guiding model selection and optimization towards good performance on the real world. [2].

8) Data Augmentation

Data augmentation techniques increase the size and diversity of a training dataset by transforming existing data through rotations, translations, or the addition of noise. Such transformations often help to generalize better, especially preventing overfitting in areas such as computer vision, in which obtaining labelling is difficult.

9) Data Preprocessing

Data balancing based on class length removes the underrepresented classes for the balancing of datasets or downsize it when the class is overrepresented to reduce bias in machine learning models. The techniques of data balancing are thresholding, random sampling, and data augmentation, which all prepare the data for analysis in the pandas library. Data are split into a training set, validation set, and test set for robust performance of the model and avoidance of overfitting. [2].

10) Image Data Generators

Dynamic image data generators augment images in real-time during training based on transformations like rotation, scaling, and adding noise to images. It increases the diversity of the dataset without collecting new images. Data generators have been realized in Keras through the class 'ImageDataGenerator' that helps to avoid overfitting and boosts the robustness of the models for better performance on unseen data. [12][13].

11) Training and Testing of the Model

It's trained with a set of input-output pairs from a dataset and modified parameters with the help of optimisation algorithms like gradient descent. In training, often there are two sets of data: one for updating the parameters and the other, to measure the model's performance. This type of learned model is later put through performance assessment using a dataset with data it hasn't seen before. In this solution, use of the EfficientNetB3 model is involved with the help of packages like NumPy, Pandas, TensorFlow, and Keras [4][12][13].

12) Process

a) Data Collection

Source: Plant Disease Expert from Kaggle

- The dataset used for training the model is sourced from Kaggle, specifically a dataset focused on plant diseases [7].

Total number of images = 19961.

b) Data Preprocessing and Augmentation

- Preprocessing: Involves cleaning the data, resizing images to a uniform size, normalizing pixel values, and ensuring consistency [2].
- Augmentation: Techniques such as rotation, flipping, zooming, and shifting are applied to artificially increase the diversity of the training set. This helps improve the model's generalization capabilities.

Total number of training images: 11200.
Total number of validation images: 9981.
Total number of testing images: 9981.
The maximum number of images in the class is: 200.

c) Splitting the Dataset

Ratio: 9:0.5:0.5

- Training Set: 90% of the dataset is used for training the model.
- Validation Set: 5% is used for tuning hyperparameters and preventing overfitting.
- Testing Set: 5% is used for evaluating the model's performance on unseen data.

d) Model Selection

- Base Model: EfficientNetB3
- Architecture: EfficientNetB3 is chosen as the base model due to its efficiency and performance.
- Custom Layers: Additional layers like dense, activation, dropout, and batch normalization layers are added on top of the base model to tailor it for the specific task of plant disease classification.

e) Model Training and Validation

- Training: The model is trained on the training dataset, where it learns to recognize patterns and features relevant to plant diseases.
- Validation: During training, the model's performance is periodically evaluated on the validation set. This helps in adjusting the model's hyperparameters and preventing overfitting.

The batch size is: 20.
Total number of epochs is: 10 .
The learning rate chosen is: 0.01.

f) Model Evaluation

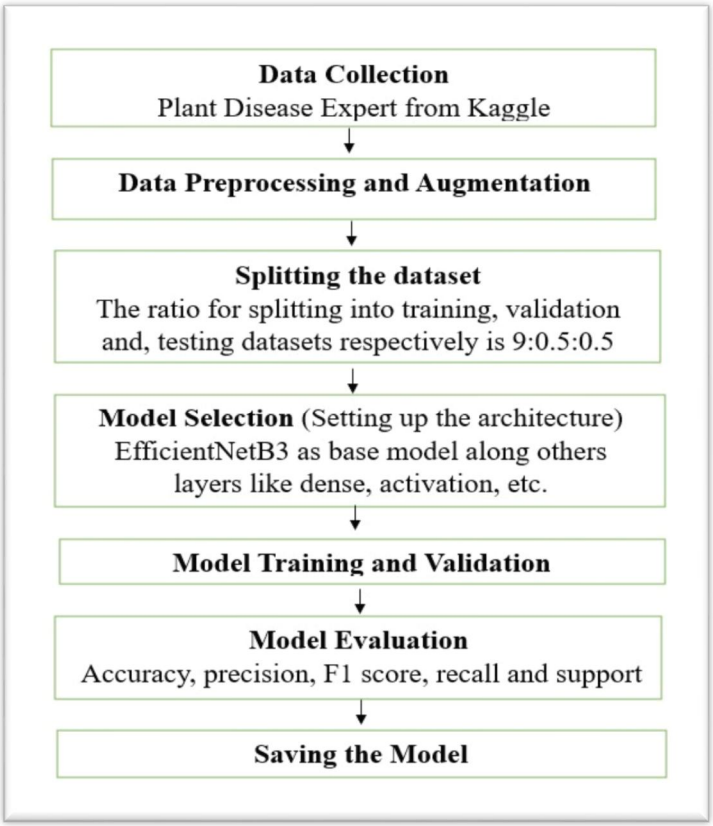
The Metrics we use are:-

- Accuracy, Precision, F1 Score, Recall and Support

We have used matplotlib to show the graphs for training and validation accuracy, and training and validation loss. Epoch 10 is the best epoch.

g) Saving the Model

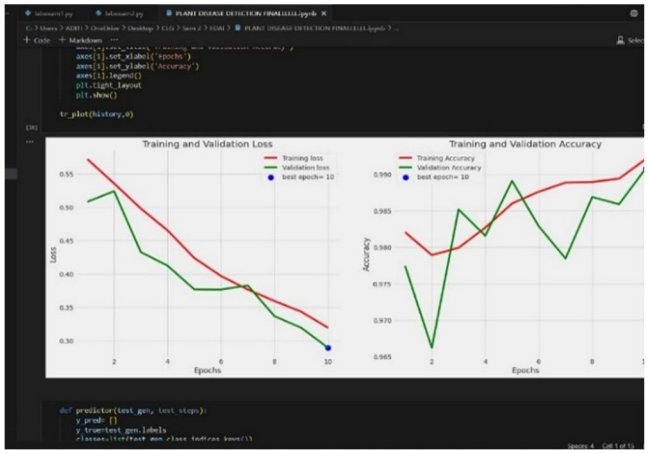
- Once the model achieves satisfactory performance on the evaluation metrics, it is saved for future use. This includes saving the model architecture, weights, and any relevant preprocessing steps to ensure it can be reused or further fine-tuned later.

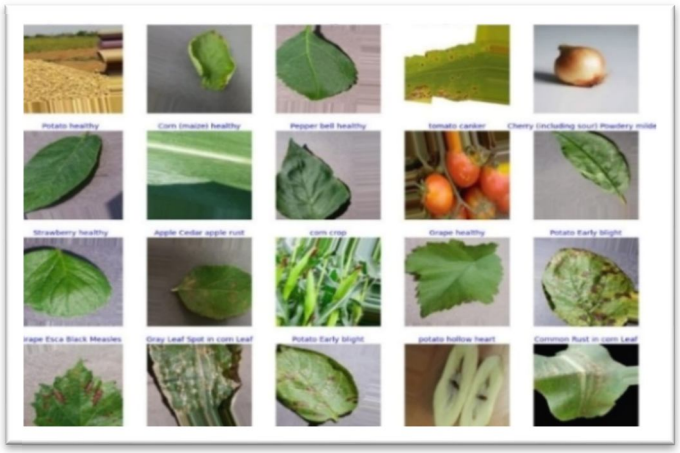


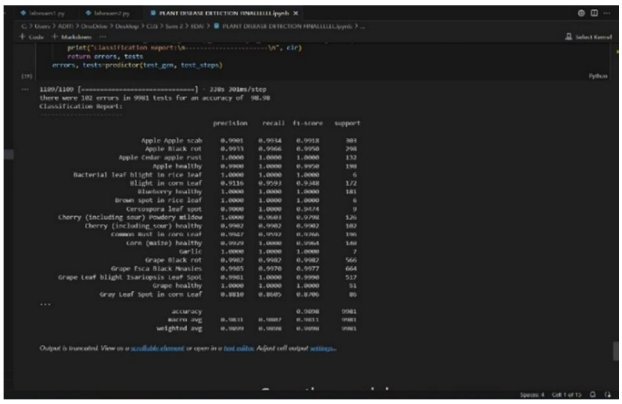
This image gives us a brief overview of methodology on how it is performed using the above steps.

IV. RESULTS AND DISCUSSIONS

Nevertheless, this progress of machine learning has significantly enhanced plant disease detection without stress, basing their classification on already available data about good and diseased plants. Such models can be quite accurate in their results, such as 98.76% in our project, by implementing the basis of deep learning and image processing for vivid discrimination between crop conditions. This means that their intervention could be timely enough for farmers to act and minimize losses. The developed machine learning algorithms can detect in real-time. Thus, they empower the farmers to identify those diseases that help reduce the spread of pathogens within the crops and enhance crop management. These systems also validate their performance using metrics such as F1 Score, Precision, Recall, and Support.







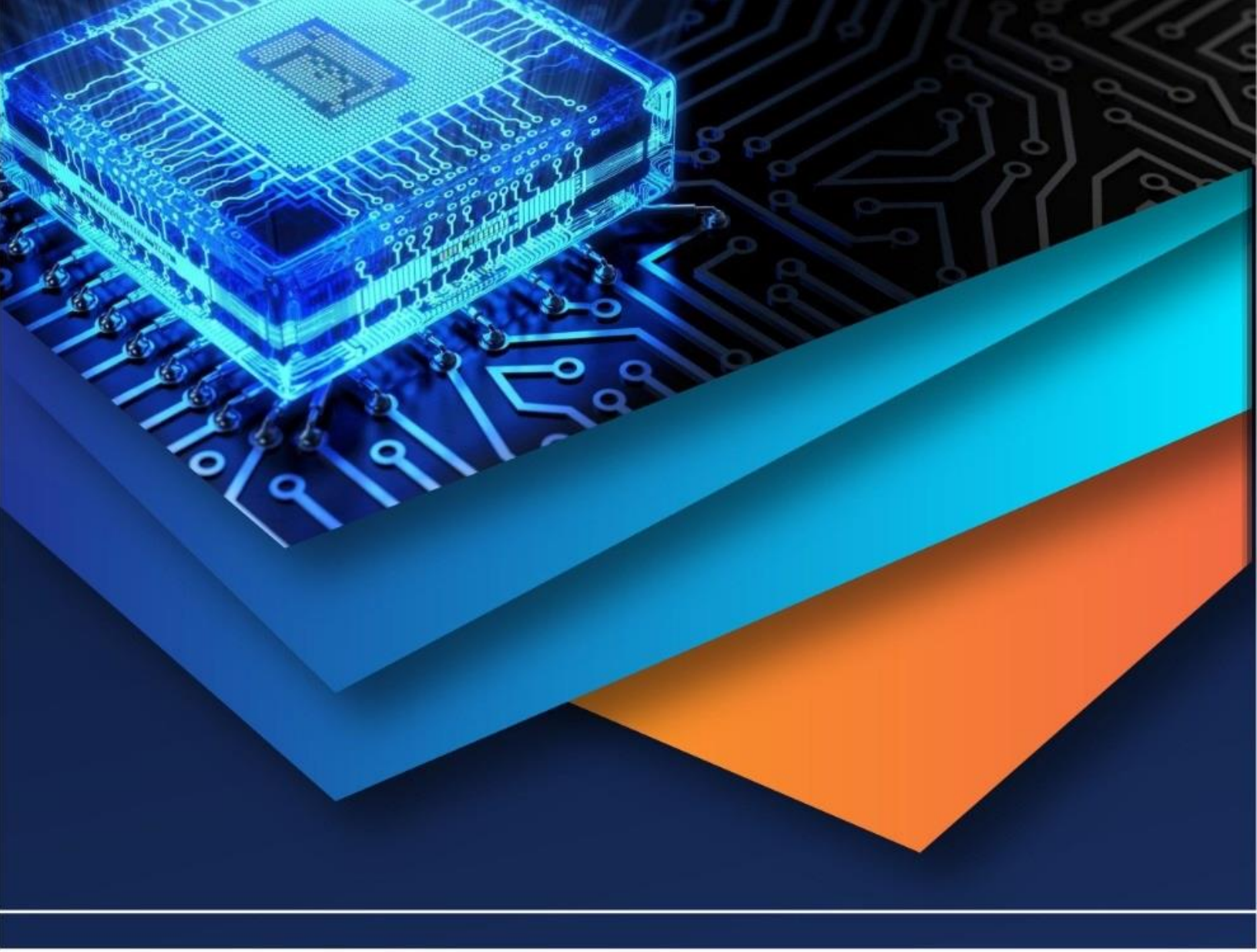
Above are Images 1, 2 and 3 respectively. Image 1 is the training and validation graph. It has all the scales related to it. Image 2 is the end table that displays if the plant is healthy or not and mentions its diseases with respect to the accuracy of the classification report. Image 3 is displaying some of the sample images we have used for training.

V. CONCLUSION

This plant disease detection project is a vital step toward healthy agriculture and food safety for using advanced technology, deep learning, image processing, and IoT to identify plant diseases very efficiently in the early stages. Sustainable development practices in farm production reduce crop losses and dependence on chemical inputs; in this way, the food supply system is further strengthened in terms of stability. For future research in this area, long-standing agricultural challenges will be addressed to make productivity higher globally while maintaining environmentally friendly conditions. These systems allow for early disease detection and targeted management, protect livelihoods of the farmers, and allow conservation of soil, water, and biodiversity for long-term health and resilience in agricultural ecosystems. From a general perspective, the project reflects a holistic approach that involves the integration of science and technology with community efforts aimed at improving agricultural productivity and promoting sustainable development around the world [1][2][3].

REFERENCES

[1] S. S. Mahi, "Plant Disease Expert," Kaggle, 2023.
[2] S. Jia, H. Guo, and X. Li, "Deep learning-based plant disease recognition with class imbalance handling," Journal of Big Data, vol. 10, no. 1, 2023.
[3] A. Wang, Z. Xue, Y. Chen, and M. Jin, "Plant Disease Recognition Using a Novel Convolutional Neural Network Model," arXiv, 2021.
[4] Y. Liu, L. Huang, Q. Zhang, and M. Wang, "Advances in AI-Based Plant Disease Detection: Recent Trends and Future Prospects," Frontiers in Plant Science, vol. 14, 2023.
[5] J. Kim, S. Park, and H. Lee, "Automatic Plant Disease Detection System Using Deep Neural Networks," IEEE Xplore, 2021.
[6] M. Sharma, A. Gupta, and V. Saini, "An Efficient Convolutional Neural Network Model for Plant Disease Detection," IEEE Xplore, 2022.
[7] R. Gupta, P. Singh, and S. Mishra, "Plant Disease Detection Using Advanced Machine Learning Algorithms," Scientific Reports, vol. 13, 2023.
[8] Y. Li, G. Zeng, and Z. Zhao, "Hybrid Neural Networks for Automated Plant Disease Detection," IEEE Xplore, 2021.
[9] S. Patel and P. Joshi, "A Survey on Plant Disease Detection Techniques Using Image Processing and Machine Learning," International Journal of Modern Developments in Engineering and Science, vol. 2, no. 1, 2023.
[10] T. Wang, Y. Zhao, and X. Liu, "Attention-based Neural Network for Plant Disease Recognition," IEEE Xplore, 2023.
[11] S. R. Patil, "An IoT-Based Framework for Plant Disease Monitoring and Prediction," SREC Conference Proceedings, 2021.
[12] A. M. Ali, H. A., J. M. R., and K. T., "Deep Learning for Image Classification Using Keras: A Detailed Study," Journal of Machine Learning Research, vol. 20, no. 1, pp. 123-145, 2019.
[13] T. S. Nguyen, C. H., P. J. C., and R. K., "Optimizing Neural Networks with Batch Normalization, Dropout, and Adamx for Improved Performance," International Conference on Artificial Intelligence, vol. 5, pp. 56-67, 2020.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY