Electro-agroguard: Intelligent Plant Disease Identification API Using Neural Networks with Drone-Captured Images

Bidishasaimantini Nayak[1*], Dr. Chandrabhanu Mishra[2], Mrs. Satyabhama Dash[3], Mr. Debi Prasad Dash[4]

[1*]*Student, Electronics and Instrumentation Engineering, OUTR, Bhubaneswar, Odisha, India Email- nayakbidishasaimantini@gmail.com*

[2]*Professor, Electronics and Instrumentation Engineering, OUTR, Bhubaneswar, Odisha, India, Email- cmishra@outr.ac.in*

[3]*Asst. Professor, Electronics and Instrumentation Engineering, OUTR, Bhubaneswar, Odisha, India, Email- sdashie@outr.ac.in*

[4]*Asst. Professor, Electronics and Instrumentation Engineering, OUTR, Bhubaneswar, Odisha, India, Email- dpdashie@outr.ac.in*

Abstract

Plant diseases like Powdery Mildew and Rust lead to enormous losses and harm to the plants like Tomatoes, Carrot, Beets, and Sunflower in world widely. The most effective approach to prevent problems and save losses is to employ precise and prompt illness detection techniques. One of the main areas of research in intelligent agriculture is the application of end-to-end machine learning, which employs computer vision techniques for the early identification of plant diseases and a TensorFlow-based neural network model with a quick API feature to detect plant diseases such as Powdery Mildew and Rust. In this study the complete project lifecycle is covered, starting with the analysis of picture captured by drone data and ending with the implementation of an API utilizing the Fast API framework. An extensive examination of picture data pertaining to plant diseases is part of the first phase. After that, a neural network model is built with Tensor Flow and Keas to categorize photos and detect whether plants have powdery mildew or rust. The resilience of the model in real-world circumstances is ensured by training it on a dataset that comprises varied examples of infected plants.The created model is deployed as a Fast API application to enable end users to query the model for disease detection on plant photos on an easier basis. To improve the disease detection system's flexibility and accessibility, an Arduino-based system is employed for image capture in addition to conventional techniques. By making it easier to capture images in a variety of settings, the use of Drone enhances the project and ultimately raises the general efficacy of the created plant disease detection model. Our method seeks to offer an automated and effective solution for plant disease identification by utilizing the capabilities of Fast API approaches and machine learning algorithms.

Keywords: *Machine learning, TensorFlow, Keras, Fast API, Drone, Powdery mildew, Rust, Plant Leaf Disease*

1. Introduction

A state-of-the-art end-to-end machine learning project, the Intelligent Plant Disease Identification API is intended to accurately detect plant diseases including Powdery mildew and rust. This study combines Support Vector Machines (SVM) and Neural Networks using cutting-edge machine learning techniques to produce a reliable and intelligent system for recognizing plant illnesses. In this study the Arduino based system is employed to analyze drone captured diseased plant images like powdery mildew and rust.

By using Drone, the project gains a useful component that makes it possible to take real-time pictures of plants. The combination of software, hardware (Drone), and machine learning produces a comprehensive solution that can handle the complex problems involved in plant health monitoring.

TensorFlow, a potent open-source platform builds the core of machine learning model. Plant disease detection is ensured by the integration of SVM and Neural Networks, which improves the system's accuracy and dependability.

TensorFlow and Keras for machine learning model development, the Application Programing Interface is constructed by the help of Fast API, Drone is employed for capturing diseased plant images, Linux for system-level operations, and Docker for containerization are some of the key technologies used in this project. These technologies work together seamlessly to produce a flexible and scalable simple solution that makes integration and deployment across a range of situations.

Additionally, the project uses WinSCP and other secure file transfer technologies to facilitate effective data transmission between the Drone captured picture data and the central system. The end product is an Intelligent Plant Disease Identification API that uses a whole technical stack in addition to advanced machine learning algorithms to provide precise, timely, and useful insights into plant health.

2. Literature Review

A study on different classification methods that can be applied to the categorization of plant leaf diseases is conducted in [1]. This publication provides a graphic of the many classification methods used to classify plant leaf diseases. Among all methods, the k-NN method may be the easiest to use and the major drawback is how time-consuming forecast which can also handle noisy inputs. Which can be overcome by SVM, highly effective at organizing high-dimensional data sets when combined with the finest machine learning algorithms. Computational complexity in SVM is reduced to a quadratic Enhancement problem.

The first stage in the specified processing strategy, according to the authors of paper [2], is to develop a color transformation structure for the RGB input image. The next stage involves hiding or removing green pixels by applying a threshold value to them. The last step is to remove green pixels from the image and mask them with a threshold level that has already been set in order to recover useful segments from the image. The completion of segmentation is the fourth and last major stage.

Plant disease identification and classification have been the subject of numerous Machine Learning (ML) models, as demonstrated by Saleem, M. H. et al.; however, with the development of Deep Learning (DL), this area of study seems to hold great potential. An extensive summary of the DL models that are used to depict various plant diseases. It is now possible to diagnose plant illnesses more transparently and accurately. [3]

In this paper[4] describes four of the several stages in the disease identification process where the input RGB image is first subjected to a color transformation structure; the green pixels are then masked and eliminated based on a predetermined threshold value; the next step covers image segmentation, where texture statistics are computed to obtain segments that can be used for illness classification; the final step employes a classifier applied to the retrieved features. The robustness of the algorithm is demonstrated using experimental results from about 500 plant leaves stored in a database.

In [5] Kulkarni et al. have developed a method for the accurate and timely identification of plant diseases using artificial neural network (ANN) technology and several image processing techniques. Recognition rates rise to 91% when using the proposed method, which uses an ANN classifier for classification and a Gabor filter for feature extraction. Using a combination of texture, color, and features, an ANN-based classifier identifies several plant illnesses.

The majority of plant illnesses in this study [6] are caused by fungi and manifest as patches on the leaves. causes serious loss if treatment is delayed. By utilizing image processing techniques to estimate the severity of the disease, the proper amount and concentration of pesticide may be applied to the disease's locations. The leaf area and lesion region area are separated, respectively, using the simple threshold and triangle thresholding techniques. In the end, the quotient of lesion area and leaf area is used to identify illnesses. The experiment's accuracy is determined to be 96.60%. According to research, this approach is quick and accurate for determining the severity of leaf disease.

This study [11] introduces a general method for object recognition and image segmentation that may adjust the parameters of the image segmentation algorithm in response to changing environmental variables. A group of generalized stochastic learning automata serve as the segmentation parameters, which are trained using connectionist reinforcement learning methods. To lower the computational costs related to model matching in the early stages of adaptation, the edge-border coincidence measure is first employed as reinforcement for segmentation evaluation. However, the result of object recognition cannot be accurately predicted by this measure alone.

As such, it is combined with model matching, wherein the matching confidence serves as a signal for reinforcement to deliver the best segmentation evaluation possible in a closed-loop object recognition system.

3. Model Analysis

With the aid of drones its built-in digital camera-based drone takes pictures of several leaf species, which are then store at our device then processed to identify the damage spots.

This algorithm illustrates how the suggested techniques for picture recognition in real-world scenarios:

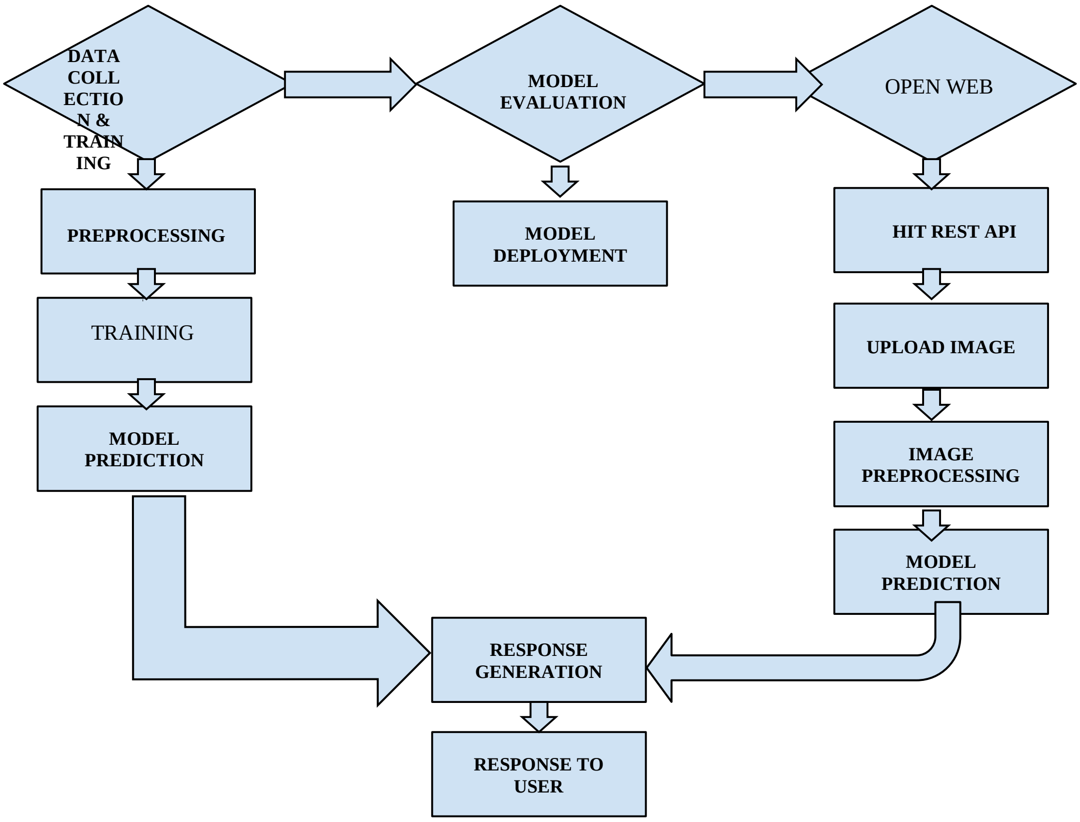

Figure 1: Model Analysis

- **DATA COLLECTING AND TRAINING:** In the 1st stage we collect a dataset of plant images with labels indicating whether the plant is healthy or diseased and use it for a training purpose. Then preprocessing is performed on incoming images to improve their quality, resize pictures to a standard dimension, adjust pixel values to fall inside a certain range. Divide the dataset into sets for testing and training after trained dataset model predicted the result and responded the output as which disease caused on the leaf to the user.
-

- **MODEL EVALUATION AND DEPLOYMENT:** Use a pre-trained convolutional neural network (CNN) as a base. Fine-tune the model on the plant disease dataset. Train the model on the training dataset. Under the Model evaluation it evaluated the model on the testing dataset to assess its performance then deployed as adjusted hyperparameters or the model architecture as needed.

- **API INTEGRATION:**

(i) We have Created a Fast API application and implemented an endpoint for receiving images for prediction.
(ii) Opened any fav Web browser and hit our Rest API. It redirected to our web page it showed post image option we have clicked on it and chose try it out.
(iii) Select the desired image (leaf) and uploaded it, then we executed our page.
(iv) It preprocesses our uploaded image and extract the features and provide us the predicted value.
(v) Then generated the result as Rust, Powdery Mildew or Healthy with a code of 200 for successfully running and it showed to user.

- **MATHMATICAL MODEL**

CNNs are frequently employed in image classification applications. Convolutional layers allow the network to automatically acquire hierarchical elements from the input images, such as patterns, textures, and spatial relationships, convolutional neural networks, or CNNs, are used for image categorization.

Following is the definition of the convolution operation:

$$(f * g)(\mathrm{t}) = f(x) = a_0 + \sum_{n=-\infty}^{\infty} f(a,b)\ g(t-a, t-b) \quad \text{(i)}$$

The Activation function commonly used in CNN:

$$f(x) = \max(0, \mathrm{x}) \quad \text{(ii)}$$

After training the model has an accuracy of 93.75%

4. EXPERIMENTAL RESULT:

The model was constructed with Keras (a TensorFlow backend), trained on the dataset's train images which are drawn from Kaggle, then assessed on the dataset's test images.
Three labels "Healthy," "Powdery," and "Rust" referring to different plant states are present in the dataset that is used to train the neural network. The train, test, and validation set each contain 630 images in total.

The following parameters are used to train the model:

num_of_class = 3
nb_train_samples = 400
nb_validation_samples = 100
nb_eval_samples = 60
epochs = 50
batch_size = 16

While selecting any picture, every image can be thought of as a 3D tensor with shape (height, breadth, channels). Each pixel in an RGB image has three values: red, green, and blue.

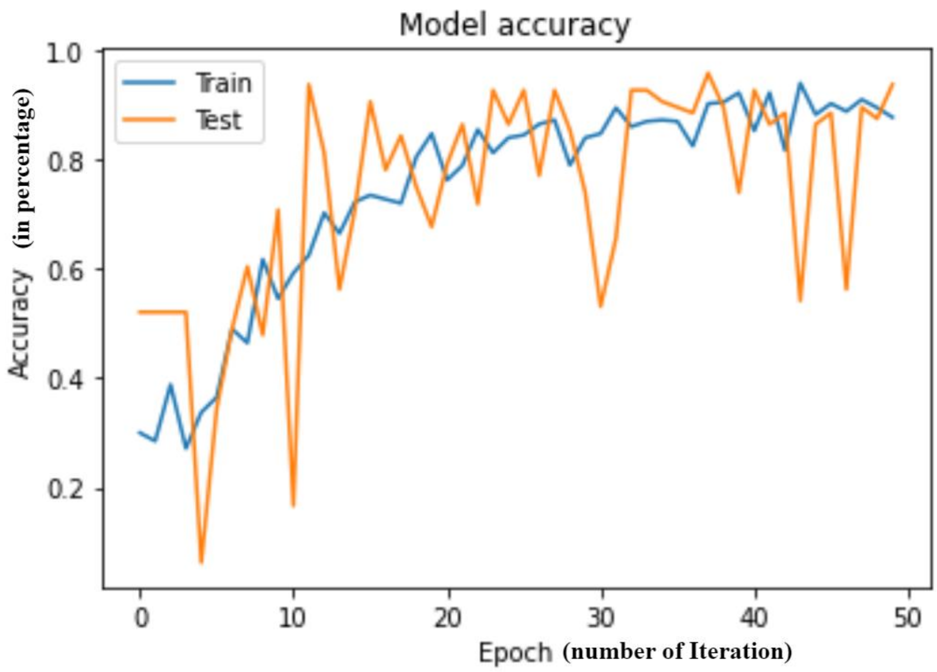

Figure 2: CNN Model accuracy graph

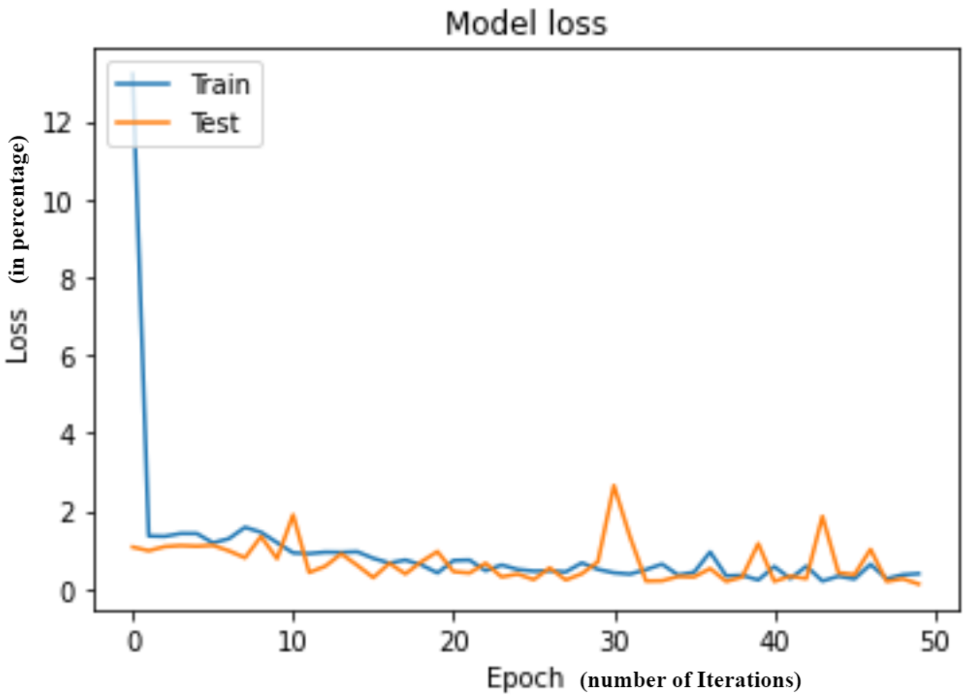

Figure 3: CNN Model loss graph

HOW TO USE THE APPLICATION:

The following steps are involved while browsing the API which are shown below.
Step 1- Open any favorite browser then hit on the API http://20.69.237.18:8000/docs it will redirect to the Web page.

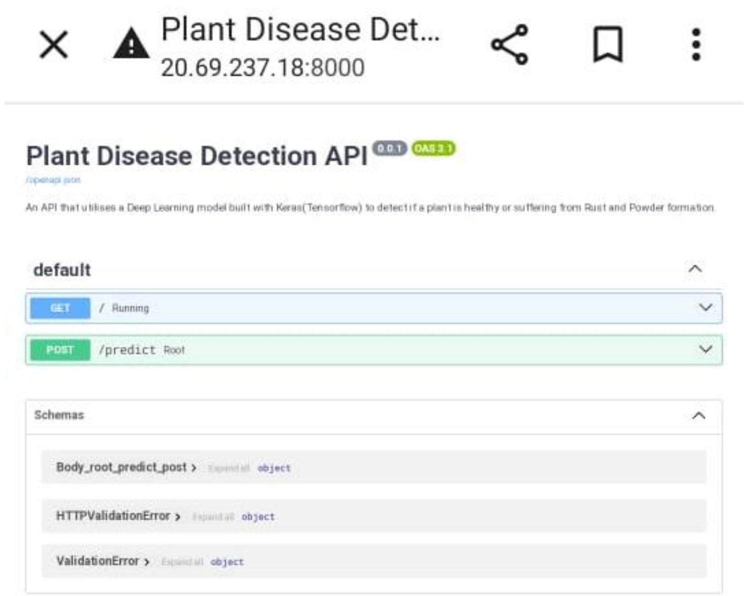

Figure 4: Front-End side of web page

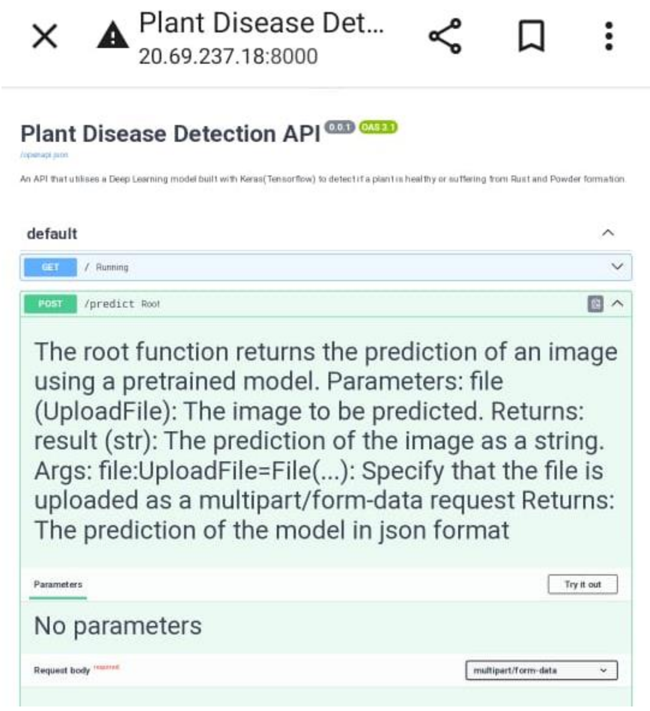

Figure 5: The image posting Web page

Step 2- Click on Post and then try it out.

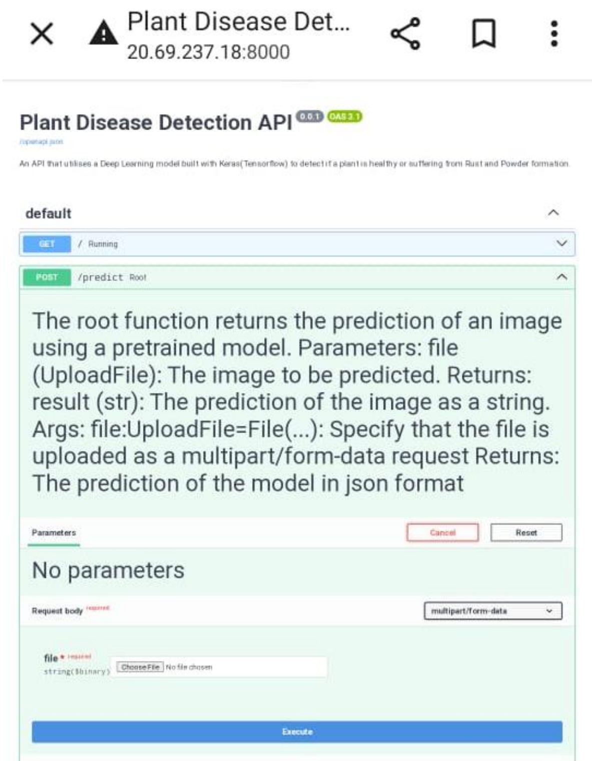

Figure 6: Image selection operation Web page

Step 3-Select desired image captured by Drone.

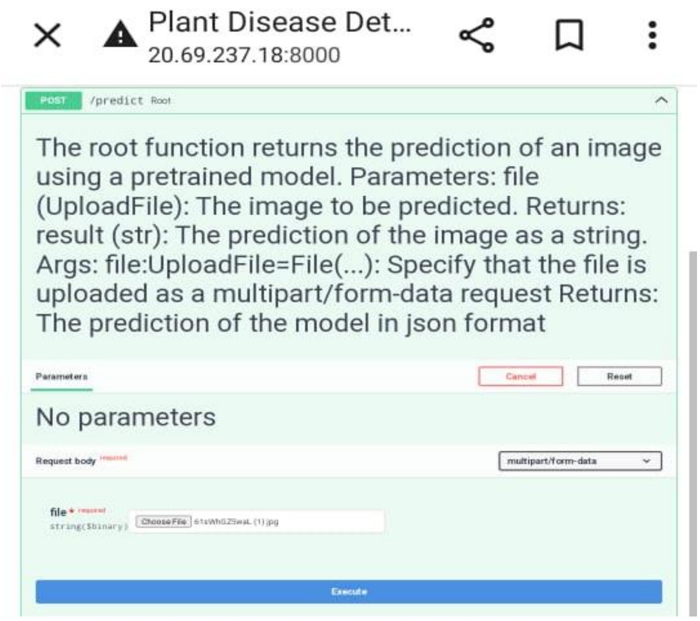

Figure 6: Picture execution Web page

Step-4 Click on Execute.

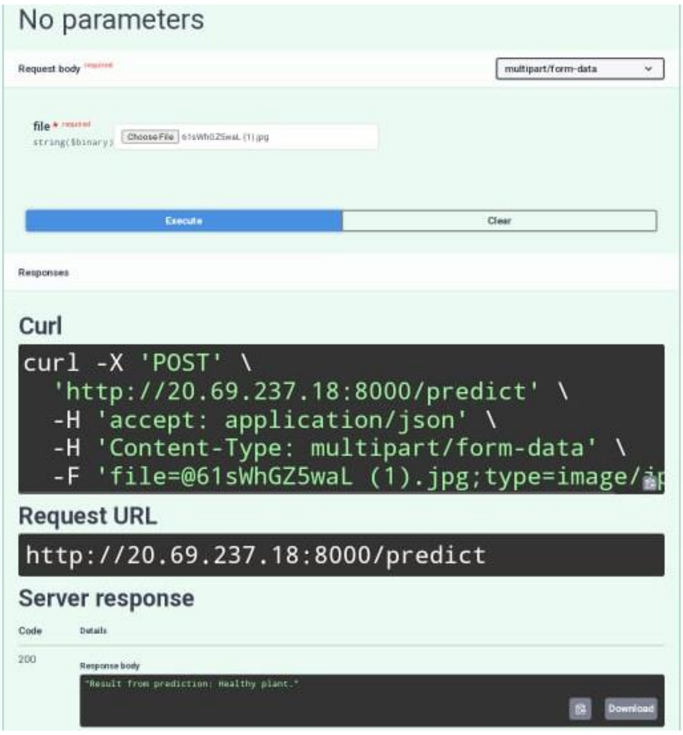

Figure 7: Result of Executed image Web page

It will generate the response as the desired output result as Rust, Powdery Mildew or Healthy.

CONCLUSION:

The study presents a method for creation and application of an API for detecting plant diseases using TensorFlow and Keras, which achieves a remarkable 93.75% accuracy rate. Plant diseases like powdery mildew and rust have resulted in significant issues that have been addressed in part by the application of deep learning algorithms. Our understanding of crop protection and yield optimization could undergo a radical change as a result of the model's high accuracy and robustness, which supports early and accurate disease identification.

Farmers are now able to take proactive steps to lessen the effect of crop diseases because of technology advancements that provide opportunities for prompt intervention. As a leading innovation, Electro-AgroGuard provides a clever, user-friendly, and effective solution for plant disease identification. With continued development, the API has the potential to completely transform crop management techniques, supporting sustainable agriculture and guaranteeing food security in a world that is changing all the time. With its combination of advanced machine learning methods, Arduino technology, and neural networks, Electro-AgroGuard is a potentially useful tool for people trying to protect the health of Plants.

References

[1] *Savita N. Ghaiwat, Parul Arora, "Detection and Classification of Plant Leaf Diseases Using Image processing Techniques: A Review", International Journal of Recent Advances in Engineering and Technology, ISSN(online) :2347-2812, Volume 2 Issues 3 2014.*

[2] *Sanjay B. Dhaygude, Nitin P. Kumbhar Agricultural plant leaf disease detection using image processing Int J Adv Res Electr Electron Instrum Eng, 2 (1) (2013). R.Badnakhe Mrunalini, Prashant.*

[3] *Arivazhagan, S., R. N. Shebiah, S. S. Nidhyanandhan, and L. Ganesan, "Classification ofcitrus and non-citrus fruits using texture features", Computing Communication and Networking Technologies, ICCCNT-2010.*

[4] *S. Arivazhagan, R. NewlinShebiah, S. An anthi, S. Vishnu Varthini Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features Agric Eng Int CIGR, 15 (1) (2013), pp. 211-217.*

[5] *Anand.H.Kulkarni and Ashwin Patil R. K, "Applying image processing technique to detect plant diseases", International Journal of Modern Engineering Research (IJMER), Vol.2, Issue.5, Sep-Oct. 2012 pp-3661-3664 ISSN: 2249-6645 2012.*

[6] *Sanjay B. Patil, et al. Leaf disease severity measurement using image processing Int J Eng Technol, 3 (5) (2011), pp. 297-301.*

[7] *Yan K, Rahul Sukthankar , "pca-sift: a more distinctive representation for local image descriptors" Computer Vision and Pattern Recognition, 2004, CVPR 2004.,Proceedings of the 2004, IEEE Computer Society Conference on ,2004 pp.506-513.*

[8] *ArtiN. Rathod, Bhavesh Tanawal, Vatsal Shah Image processing techniques for detection of leaf disease Int J Adv Res Comput Sci Softw Eng, 3 (11) (2013).*

[9] *S. Beucher, F. Meyer The morphological approach to segmentation: the watershed transforms E.R. Dougherty (Ed.), Mathema tical morphology image processing, vol. 12, Marcel Dekker, New York (1993),pp. 433-481.*

[10] *B. Bhanu, S. Lee, J. Ming Adaptive image segmentation using a genetic algorithm IEEE Trans Syst Man Cybern, 25 (Dec 1995), pp. 1543-1567.*

[11] *B. Bhanu, J. Peng Adaptive integrated image segmentation and object recognition IEEE Trans Syst Man Cybern Part C, 30 (2000), pp. 427-441.*