e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:07/July-2024 Impact Factor- 7.868 www.irjmets.com

DETECTION AND ANALYSIS OF PLANT LEAF DISEASE USING ARTIFICIAL INTELLIGENCE

Kiran Nikhade*1, Dr. D.D. Nawgaje*2

*1Student, Department Of Electronics & Telecommunication Engineering, SSGMCE, Shegaon, Maharashtra, India.

*2Associate Professor, Department Of Electronics & Telecommunication Engineering, SSGMCE, Shegaon, Maharashtra, India.

ABSTRACT

In India about 70% of the crowd relies on agriculture. Identification of the plant diseases is important in order to prevent the losses within the yield. It's terribly troublesome to observe the plant diseases manually. It needs tremendous quantity of labor, expertise within the plant diseases, and conjointly need the excessive time interval. Hence, image processing and machine learning models can be employed for the detection of plant diseases. In this project, we have described the technique for the detection of plant diseases with the help of their leaves pictures. Image processing is a branch of signal processing which can extract the image properties or useful information from the image. The main aim of machine learning is to understand the training data and fit that training data into models that should be useful to the people. So it can assist in good decisions making and predicting the correct output using the large amount of training data. The colour of leaves, amount of damage to leaves, area of the leaf, texture parameters are used for classification. In this project we have analyzed different image parameters or features to identifying different plant leaves diseases to achieve the best accuracy. Previously plant disease detection is done by visual inspection of the leaves or some chemical processes by experts. In such conditions, the recommended system proves to be helpful in monitoring large fields of crops. Automatic detection of the diseases by simply seeing the symptoms on the plant leaves makes it easier as well as cheaper. The proposed solution for plant disease detection is computationally less expensive and requires less time for prediction than other deep learning based approaches since it uses statistical machine learning and image processing algorithm.

Keywords: Analysis, Detection, Technique, Machine Learning, Diseases.

I. INTRODUCTION

The use of ML and DL in plant disease detection has gained popularity and shown promising results in accurately identifying plant diseases from digital images. Traditional ML techniques, such as feature extraction and classification, have been widely used in the field of plant disease detection. These methods extract features from images, such as color, texture, and shape, to train a classifier that can differentiate between healthy and diseased plants. These methods have been widely used for the detection of diseases such as leaf blotch, powdery mildew, and rust, as well as disease symptoms from abiotic stresses such as drought and nutrient deficiency (Mohanty et al., 2016; Anjna et al., 2020; Genaev et al., 2021) but have limitations in accurately identifying subtle symptoms of diseases and early-stage disease detection. In addition, they also struggle to process complex and high-resolution images. Recently, DL techniques such as convolutional neural networks (CNNs) and deep belief networks (DBNs) have been proposed for plant disease detection (Liu et al., 2017; Karthik et al.,2020). These methods involve training a network to learn the underlying features of the images, enabling the identification of subtle symptoms of diseases that traditional image processing methods may not be able to detect (Singh and Misra, 2017; Khan et al., 2021; Liu and Wang, 2021b). DL models can handle complex and large images, making them suitable for high-resolution images (Ullah et al., 2019). However, these methods require a large amount of labelled training data and may not be suitable for unseen diseases. Furthermore, DL models are computationally expensive, which may be a limitation for some applications. In recent years, several research studies have proposed different ML and DL approaches for plant disease detection. However, most studies have focused on a specific type of disease or a specific plant species. Therefore, more research is needed to develop a generalizable and robust model that can work for different plant species and diseases. Additionally, there is a need for more publicly available datasets for training and evaluating models. One of the recent trends in the field is transfer learning, a technique that allows for reusing

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:07/July-2024 Impact Factor- 7.868 www.irjmets.com

pre-trained models on new datasets. Recently, transfer learning and ensemble methods have emerged as popular trends in plant disease detection using ML and DL. Transfer learning involves fine-tuning pre-trained models on a specific dataset to enhance the performance of DL models.

II. METHODOLOGY

Our Vidarbha region has two main crop which are going to sow by each farmer i.e. Cotton and Soyabean. In this project, these two crop has considered for disease detection. Each crop has its own requirement to grow productively and also has its limitations where it may cause production loss. The proposed methodology is as represented below with its all-basic operations performed in this project. The basic technology used to detect crop disease is Decision tree machine learning algorithm with sets of fuzzy logic to introduce concept of Artificial Intelligence.

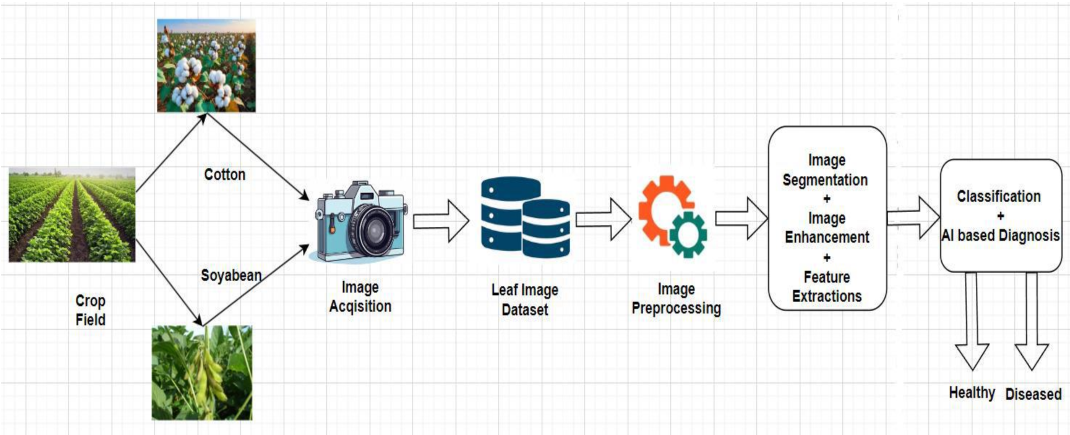


Fig.1- Block diagram of methodology for crop leaf disease detection

1. Cotton crop:

Cotton (Gossypium sp.) belongs to Malvaceae family is oldest of the all fibers used by human beings. It is known as a "King of Fibre" crop due to its global importance in agriculture as well as industrial economy. It contributes significantly to both agriculture and industry in terms of farm income, employment and export earnings. It is grown in more than 100 countries and it is estimated that, the crop is cultivated on about 2.5 per cent of the world cultivable land. Cotton cultivation has traditionally concentrated in a few countries viz: China, United States, India, Pakistan, Brazil, Uzbekistan, Turkey and Australia. Together these countries account for more than three quarter of global production. At global level, cotton area is projected to grow by 9 per cent and yield are only projected to increase 6 per cent.

2. Soyabean Crop:

Soybean [Glycine max (L.) Merrill] is a significant seed legume and assumes imperative part in human occupation. India positions fifth in territory and production of soybean after USA, Brazil, Argentina, and China1 . It contains top calibre of proteins (40%) and edible oil (20%) containing significant fundamental amino acids. Soybean acts as a top-notch protein hotspot for domesticated animals feed apportions. It is the world's most significant seed vegetable, which adds to 25% of the worldwide edible oil, about two-thirds of the world's protein concentrate for domestic animals feeding. Soybean improve the soil health and fertility by fixing nitrogen through biological nitrogen fixation in soil which is carried out by symbiotic nitrogen fixing bacteria residing in the root nodule of soyabeans. Soybean likewise has the ability to upgrade efficiency of different crops and furthermore protects the climate.

3. Image Acquisition:

In image processing, it is defined as the action of retrieving an image from some source, usually a hardware-based source for processing. It is the first step in the workflow sequence because, without an image, no processing is possible. The image that is acquired is completely unprocessed. Now the incoming energy is transformed into a voltage by the combination of input electrical power and sensor material that is responsive to a particular type of energy being detected. The output voltage waveform is the response of the sensor(s) and a digital quantity is obtained from each sensor by digitizing its response.

IRJMETS
e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:07/July-2024 Impact Factor- 7.868 www.irjmets.com

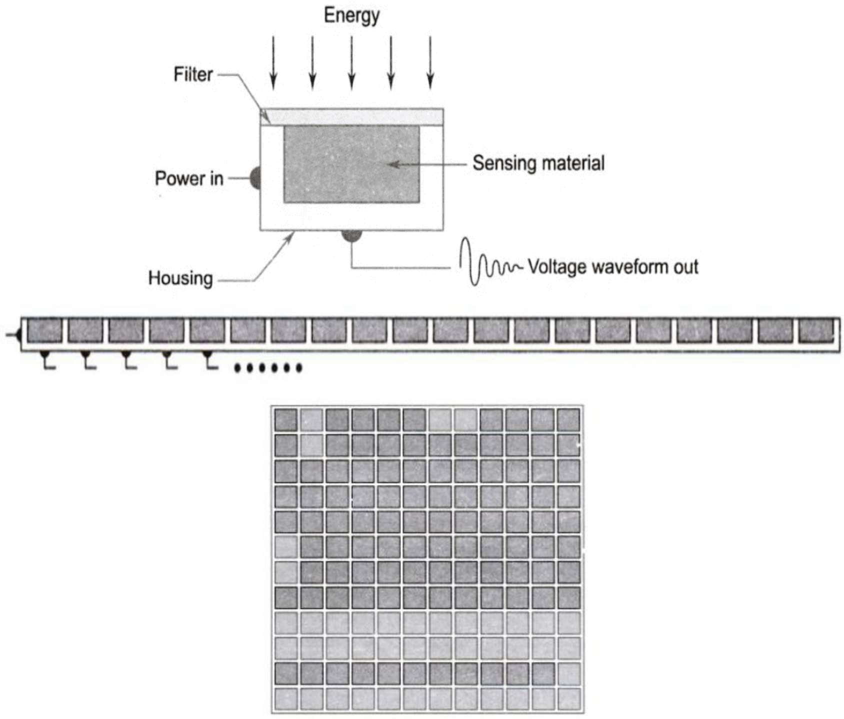


Fig.2- Single Image Sensor, Line Sensor, Array Sensor

4. Leaf Image Dataset:

The relationship between the plants and the environment is multitudinous and complex. They help in nourishing the atmosphere with diverse elements. Plants are also a substantial element in regulating carbon emission and climate change. But in the past, we have destroyed them without hesitation. For the reason that not only we have lost a number of species located in them, but also a severe result has also been encountered in the form of climate change. However, if we choose to give them time and space, plants have an astonishing ability to recover and re-cloth the earth with varied plant and species that we have, so recently, stormed. Therefore, a contribution has been made in this work towards the study of plant leaf for their identification, detection, disease diagnosis, etc.

5. Image Pre-processing:

Image preprocessing is the essential first step in teaching computers to understand pictures. In AI, it's like cleaning and organizing images before the computer learns from them. This process is crucial because it helps the computer recognize patterns more effectively, like making sure all pictures are in the same format before learning.

The clearer the data, the better the computer performs. Think about medical scans or self-driving cars – without proper image preprocessing, mistakes could happen. So, it's like setting the stage for the computer to see and understand pictures accurately, making AI and machine learning more reliable in various real-world applications.

Common Image Preprocessing Techniques:

The following are the most common image preprocessing techniques in machine learning:

a. Rescaling and Normalization:

Pixel values represent image intensity. Scaling ensures these values fall within a specific range (usually 0 to 1 or -1 to 1), making computations more manageable and preventing dominance by large values.

Normalization adjusts pixel values to have a mean of 0 and a standard deviation of 1. It enhances model stability during training by keeping values within a standardized range, aiding convergence.

b. Image Resizing and Cropping:

Resizing ensures all images in a dataset have consistent dimensions, facilitating model training. It prevents computational challenges and allows the model to learn patterns uniformly across different samples.

Cropping focuses on specific regions of an image, excluding irrelevant details. This is valuable for tasks where the location of an object is essential, optimizing the model's ability to recognize specific features.

c. Image Augmentation:

Data augmentation involves creating variations of existing images by applying transformations like rotation, flipping, and zooming. This diversifies the dataset, improving the model's ability to generalize and perform well on unseen data.

- **Rotation:** Introduces variations by rotating images.
- **Flipping:** Mirrors images horizontally or vertically.
- **Zooming:** Adjusts the scale of the image. These techniques simulate different real-world scenarios, enhancing the model's adaptability.

d. Color Space Conversion:

Color spaces represent how colours are encoded. RGB is common for images, while HSV separates color information for better manipulation. Understanding and choosing the right color space is crucial for specific tasks.

6. Image Segmentation:

Image segmentation is a computer vision technique that partitions a digital image into discrete groups of pixels—image segments—to inform object detection and related tasks. By parsing an image's complex visual data into specifically shaped segments, image segmentation enables faster, more advanced image processing.

Image classification applies a class label to an entire image. For example, a simple image classification model might be trained to categorize vehicle images as "car" or "truck". Conventional image classification systems are limited in sophistication, as they do not process individual image features separately.

Object detection combines image classification with object localization, generating rectangular regions, called "bounding boxes", in which objects are located: rather than merely labelling a vehicle image as "car" or "truck", an object detection model could indicate where in the image the cars or trucks can be found. While object detection can classify multiple elements within an image and approximate each element's width and height, it cannot discern precise boundaries or shapes. This limits the ability of conventional object detection models to delineate closely bunched objects with overlapping bounding boxes.

Image segmentation processes visual data at the pixel level, using various techniques to annotate individual pixels as belonging to a specific class or instance. "Classic" image segmentation techniques determine annotations by analysing inherent qualities of each pixel (called "heuristics") like color and intensity, while deep learning models employ complex neural networks for sophisticated pattern recognition. The outputs of this annotation are segmentation masks, representing the specific pixel-by-pixel boundary and shape of each class—typically corresponding to different objects, features or regions in the image.

7. Image Enhancement:

The principal objective of image enhancement is to process a given image so that the result is more suitable than the original image for a specific application.

It accentuates or sharpens image features such as edges, boundaries, or contrast to make a graphic display more helpful for display and analysis. The enhancement doesn't increase the inherent information content of the data, but it increases the dynamic range of the chosen features so that they can be detected easily.

These processing methods are based only on the intensity of single pixels. Simple intensity transformation: Image negatives: Negatives of digital images are useful in numerous applications, such as displaying medical images and photographing a screen with monochrome positive film with the idea of using the resulting negatives as normal slides. Transform function T : g(x,y)=L-f(x,y), where L is the max. intensity.

Before Enhancement

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:07/July-2024 Impact Factor- 7.868 www.irjmets.com

After Enhancement

8. Feature Extraction:

Feature extraction is a process used in machine learning to reduce the number of resources needed for processing without losing important or relevant information. Feature extraction helps in the reduction of the dimensionality of data which is needed to process the data effectively. In other words, feature extraction involves creating new features that still capture the essential information from the original data but in a more efficient way.

When dealing with large datasets, especially in domains like image processing, natural language processing, or signal processing, it's common to have data with numerous features, many of which may be irrelevant or redundant. Feature extraction allows for the simplification of the data which helps algorithms to run faster and more effectively.

- Reduction of Computational Cost: By reducing the dimensionality of the data, machine learning algorithms can run more quickly. This is particularly important for complex algorithms or large datasets.
- Improved Performance: Algorithms often perform better with a reduced number of features. This is because noise and irrelevant details are removed, allowing the algorithm to focus on the most important aspects of the data.
- Prevention of Overfitting: With too many features, models can become overfitted to the training data, meaning they may not generalize well to new, unseen data. Feature extraction helps to prevent this by simplifying the model.
- Better Understanding of Data: Extracting and selecting important features can provide insights into the underlying processes that generated the data.

9. Decision Tree algorithm:

- A decision tree in machine learning is a versatile, interpretable algorithm used for predictive modelling. It structures decisions based on input data, making it suitable for both classification and regression tasks. This article delves into the components, terminologies, construction, and advantages of decision trees, exploring their applications and learning algorithms.
- Decision Tree in Machine Learning: A decision tree is a type of supervised learning algorithm that is commonly used in machine learning to model and predict outcomes based on input data. It is a tree-like structure where each internal node tests on attribute, each branch corresponds to attribute value and each leaf node represents the final decision or prediction. The decision tree algorithm falls under the category of supervised learning. They can be used to solve both regression and classification problems.

Decision Tree Terminologies: There are specialized terms associated with decision trees that denote various components and facets of the tree structure and decision-making procedure :

Root Node: A decision tree's root node, which represents the original choice or feature from which the tree branches, is the highest node.

Internal Nodes (Decision Nodes): Nodes in the tree whose choices are determined by the values of particular attributes. There are branches on these nodes that go to other nodes.

Leaf Nodes (Terminal Nodes): The branches' termini, when choices or forecasts are decided upon. There are no more branches on leaf nodes.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:07/July-2024 Impact Factor- 7.868 www.irjmets.com

Branches (Edges): Links between nodes that show how decisions are made in response to particular circumstances.

Splitting: The process of dividing a node into two or more sub-nodes based on a decision criterion. It involves selecting a feature and a threshold to create subsets of data.

Parent Node: A node that is split into child nodes. The original node from which a split originates.

Child Node: Nodes created as a result of a split from a parent node.

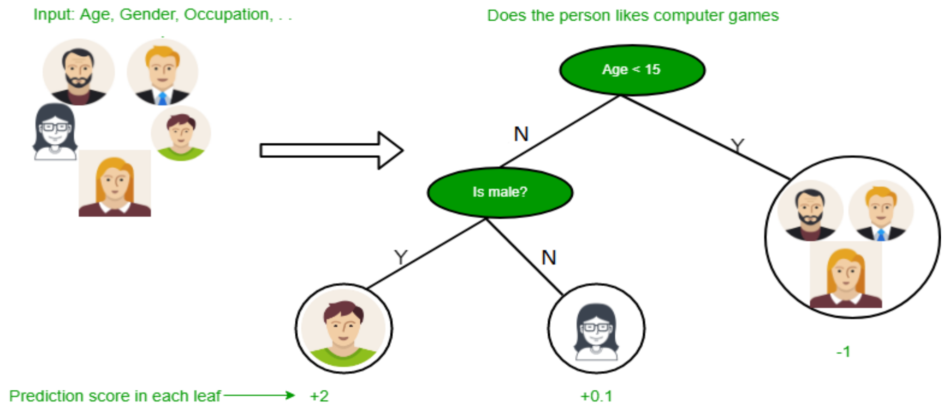
Decision Criterion: The rule or condition used to determine how the data should be split at a decision node. It involves comparing feature values against a threshold.

Pruning: The process of removing branches or nodes from a decision tree to improve its generalisation and prevent overfitting.

Understanding these terminologies is crucial for interpreting and working with decision trees in machine learning applications.

Decision Tree Approach:

Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree. We can represent any boolean function on discrete attributes using the decision tree.



This process is known as attribute selection. We have two popular attribute selection measures:

1. Information Gain
2. Gini Index

1. Information Gain:

When we use a node in a decision tree to partition the training instances into smaller subsets the entropy changes. Information gain is a measure of this change in entropy.

- Suppose S is a set of instances,
- A is an attribute
- Sv is the subset of S
- v represents an individual value that the attribute A can take and Values (A) is the set of all possible values of A, then $Gain(S,A)=Entropy(S)-\Sigma vA|Sv||S|.Entropy(Sv)$Gain(S,A)=Entropy(S)−ΣvA|S||Sv|.Entropy(Sv)

Entropy: is the measure of uncertainty of a random variable, it characterizes the impurity of an arbitrary collection of examples. The higher the entropy more the information content.

Suppose S is a set of instances, A is an attribute, Sv is the subset of S with A = v, and Values (A) is the set of all possible values of A, then

$Gain(S,A)=Entropy(S)-\Sigma v\epsilon Values(A)|Sv||S|.Entropy(Sv)$Gain(S,A)=Entropy(S)−ΣvϵValues(A)|S||Sv|.Entropy(Sv)

Example:

For the set X = {a,a,a,b,b,b,b,b} Total instances: 8 Instances of b: 5 Instances of a: 3

Entropy$H(X)$=[(38)log238+(58)log258]=−[0.375(−1.415)+0.625(−0.678)]=−(−0.53−0.424)=0.954Entropy H(X)=[(83)log283+(85)log285]=−[0.375(−1.415)+0.625(−0.678)]=−(−0.53−0.424)=0.954

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:07/July-2024 Impact Factor- 7.868 www.irjmets.com

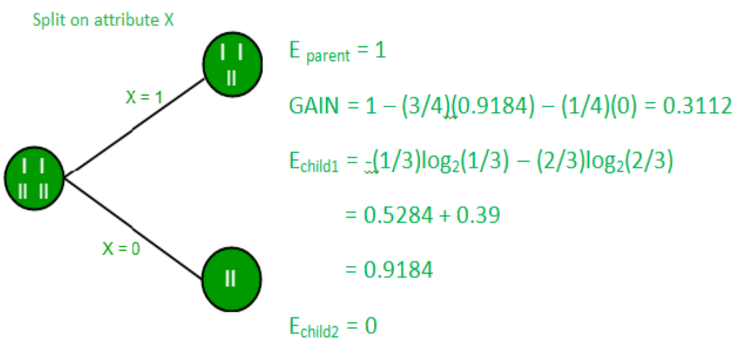
Building Decision Tree using Information Gain the essentials:

- Start with all training instances associated with the root node
- Use info gain to choose which attribute to label each node with
- Note: No root-to-leaf path should contain the same discrete attribute twice
- Recursively construct each subtree on the subset of training instances that would be classified down that path in the tree.
- If all positive or all negative training instances remain, the label that node “yes” or “no” accordingly
- If no attributes remain, label with a majority vote of training instances left at that node
- If no instances remain, label with a majority vote of the parent’s training instances.

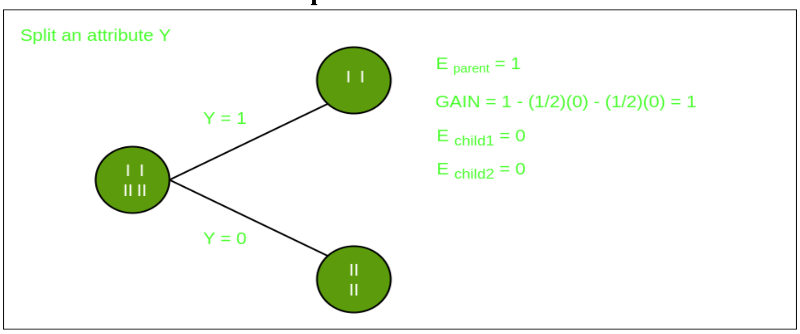
Example:

Now, let us draw a Decision Tree for the following data using Information gain.

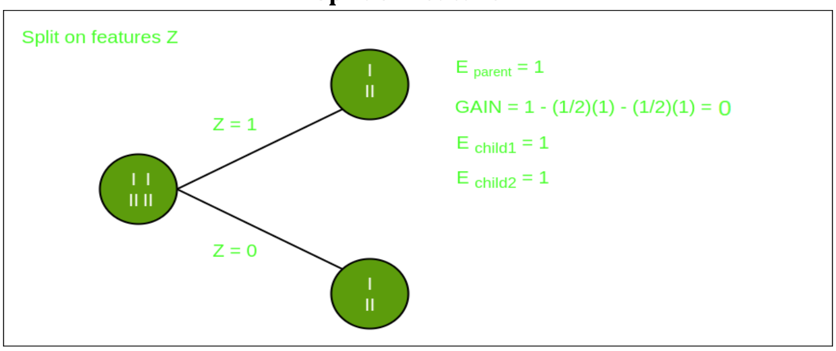
Training set: 3 features and 2 classes X	**Y**	**Z**	**C**
1	1	1	I
1	1	0	I
0	0	1	II
1	0	0	II



Split on feature X



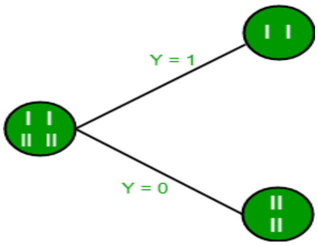
Split on feature Y



Split on feature Z

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:07/July-2024 Impact Factor- 7.868 www.irjmets.com

From the above images, we can see that the information gain is maximum when we make a split on feature Y. So, for the root node best-suited feature is feature Y. Now we can see that while splitting the dataset by feature Y, the child contains a pure subset of the target variable. So we don't need to further split the dataset. The final tree for the above dataset would look like this:



2. Ginni Index:

- Gini Index is a metric to measure how often a randomly chosen element would be incorrectly identified.
- It means an attribute with a lower Gini index should be preferred.
- Sklearn supports "Gini" criteria for Gini Index and by default, it takes "gini" value.
- The Formula for the calculation of the Gini Index is given below.

The Formula for Gini Index is given by :

$$\text{Gini}(S) = 1 - \sum_{i=1}^{c} p_i^2$$

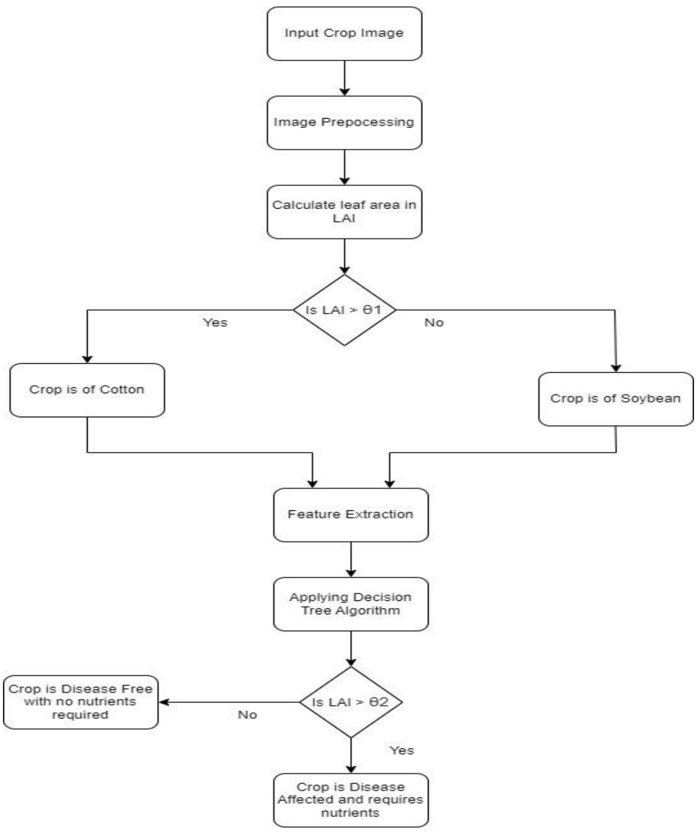
Gini Impurity :

The Gini Index is a measure of the inequality or impurity of a distribution, commonly used in decision trees and other machine learning algorithms. It ranges from 0 to 0.5, where 0 indicates a pure set (all instances belong to the same class), and 0.5 indicates a maximally impure set (instances are evenly distributed across classes).

Some additional features and characteristics of the Gini Index are:

- It is calculated by summing the squared probabilities of each outcome in a distribution and subtracting the result from 1.
- A lower Gini Index indicates a more homogeneous or pure distribution, while a higher Gini Index indicates a more heterogeneous or impure distribution.

III. FLOWCHART



IV. RESULTS AND DISCUSSION

In this project, we have used the Decision tree machine learning algorithm which calculate the threshold value and apply it to various if else conditions to predict plant leaf disease. Complete project is implemented in mat lab software in which graphical user interface GUI is used to read input crop image. It also has an ability to detect crop type i.e. cotton or soyabean using Decision tree. One more push button is implemented which applies Decision tree algorithm on input image and predict the type of disease. It also provides the required nutrients for detected crop leaf. First opening GUI platform can be visualized as shown in following fig…

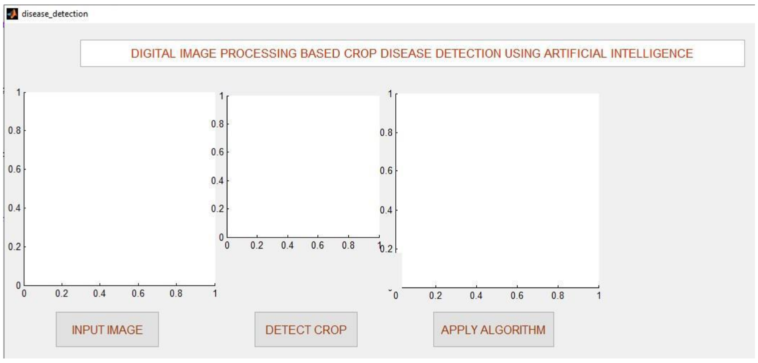


Fig. 4.1 Graphical User Interface (GUI) for crop leaf disease detection

For detection of crop type, leaf area is calculated for cotton and soybean. It is observed that, cotton crop has large leaf area than the soybean crop which help us to predict the type of crop. In this project leaf area 50000 LAI is used as threshold is used to detect crop type. Leaf area greater than threshold is considered as cotton crop and the leaf having area less than threshold is detected as soybean. The result of crop input image and predicted crop type is shown in below figure.

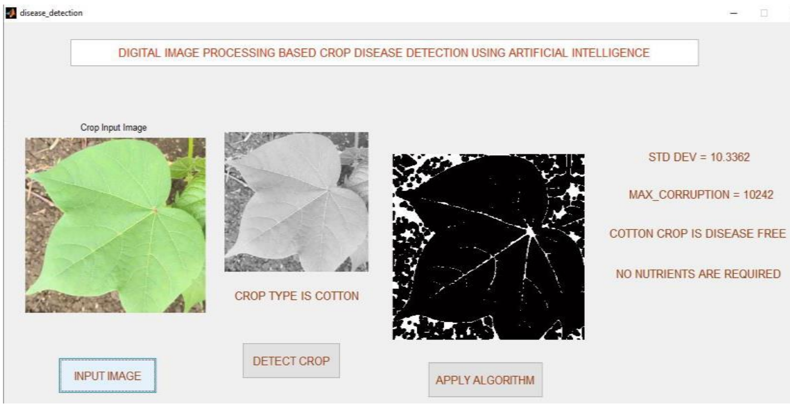


Fig. 4.2 - Cotton crop type detection on input crop image

In above graphical interface, decision tree algorithm also applied for which we have selected the cotton crop leaf which is fresh and disease free. Hence after applying algorithm, result generated showing cotton crop is disease free and not requiring any nutrients.

When user select a leaf which has some kind of disease effect, then decision tree algorithm calculates the maximum amount of leaf corruption caused due to disease. The threshold value selected to detect disease free is below 10000 LAI. When the leaf corruption caused exceeds the threshold, algorithm detect that the crop leaf is affected by disease. It can be visualize using below figure.

e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
(Peer-Reviewed, Open Access, Fully Refereed International Journal)
Volume:06/Issue:07/July-2024 Impact Factor- 7.868 www.irjmets.com

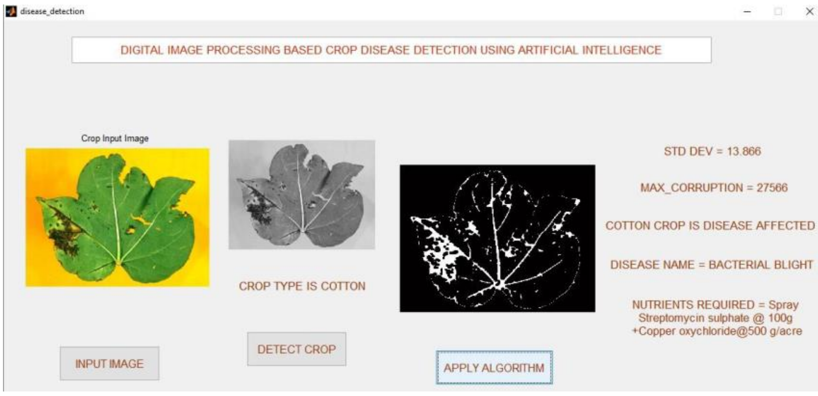


Fig. 4.3- GUI for Cotton crop leaf affected by disease

In this project we have considered two kinds of cotton crop disease i.e. bacterial blight and Alternaria leaf spot. For bacterial blight the value of maximum corruption threshold selected is 25000 LAI. The leaf having maximum corruption between 10000 to 25000 is detected as bacteria blight and hence it shows the nutrients required to spray streptomycin sulphate of 100 gram with copper oxychloride of 500 gram per acre farm area. It can be visualized in below figure.

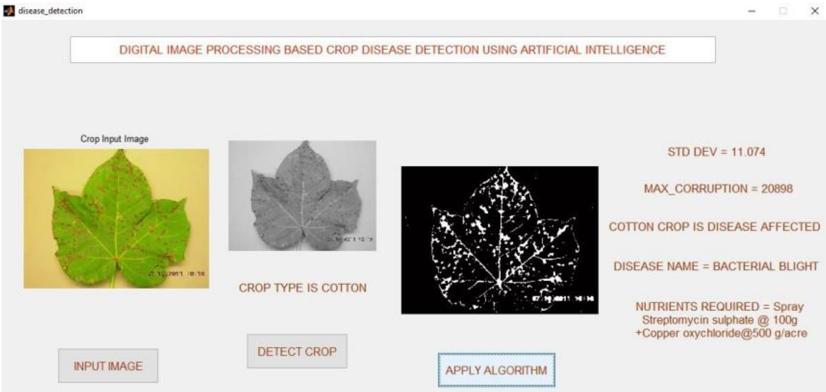


Fig. 4.4- GUI for cotton crop disease bacterial blight with nutrients required

When the maximum corruption threshold limit exceeds 25000 LAI, then an algorithm detects another disease as Alternaria leaf spot for which it suggest the nutrients potash to be used and acid delisting seed during sowing next time. It can also be seen in following figure.

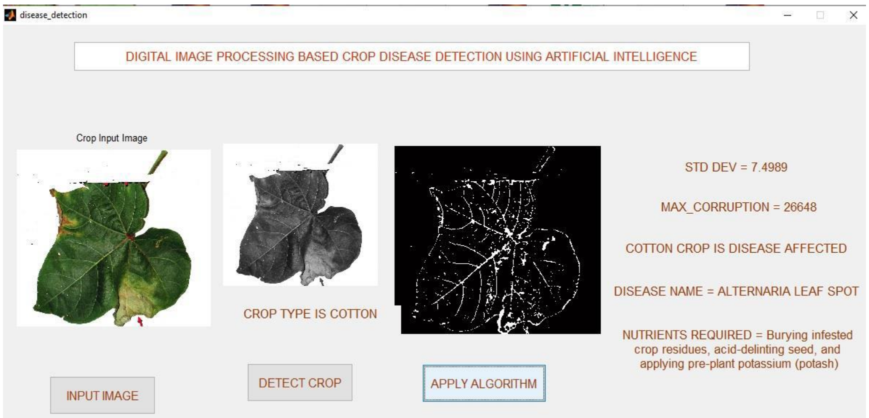


Fig. 4.5 GUI for cotton crop detected with Alternaria leaf spot

When the user selects soybean as an input image, it first detects the crop type as soybean. When fresh leaf is selected, algorithm detects soybean crop is disease free without requiring any kind of nutrients.

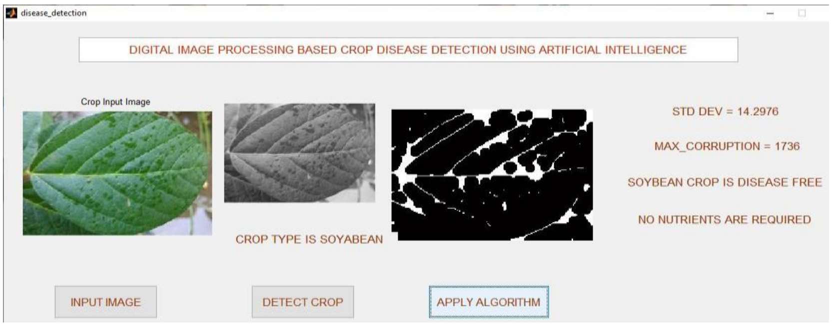


Fig. 4.6 - Soybean crop type detected without any kind of disease and nutrients

Similarly, when user select soybean crop leaf with disease, max corruption increases and algorithm predicts the disease name with its nutrients required. It uses the standard deviation and maximum corruption for thresholding and conditions. When standard deviation decreases below 10 with increase in max corruption, it predicts the disease as Soya rust with nutrients as mancozeb 2.5 gram per Liter during growth period.

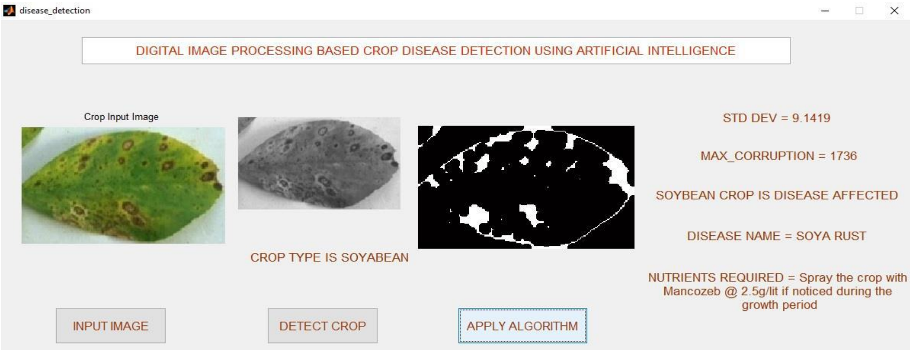


Fig. 4.7- Soybean crop with disease detected as soya rust and nutrients required

Fig. 4.8 -Soybean crop with disease detected as soya rust and nutrients required

On other side, when max corruption increases above 2000 LAI, it detects the soya crop disease as angular leaf spot with nutrients as streptomycin sulphate 100 gram with copper oxychloride 500 gram per acre of farm area.

Fig. 4.9- Soybean crop with disease detected as angular leaf spot and nutrients required

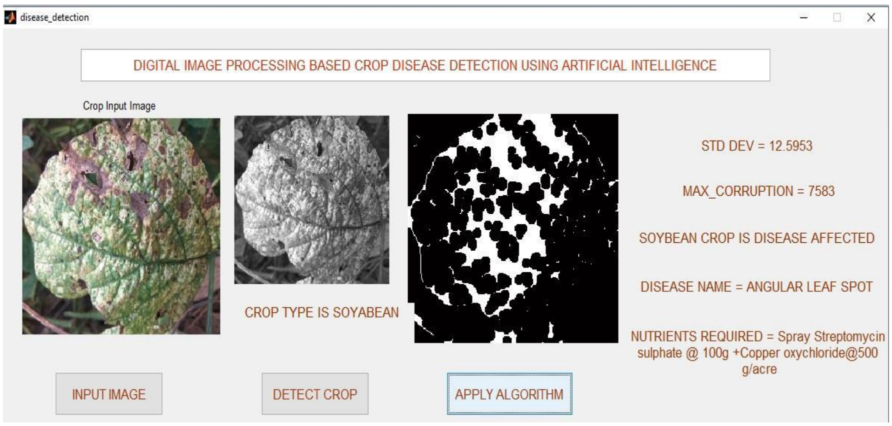


Fig. 4.10- Soybean crop with disease detected as angular leaf spot and nutrients required

As we know, every algorithm has not having 100 percent of efficiency. It fails to detect in some cases where all decided thresholds fails. Following input leaf is an example of cotton crop where our algorithm fails to detect. The input leaf is of cotton but it also looks like the soyabean leaf. Hence, the algorithm fails and detect it as soybean leaf. It also fails to detect disease type with nutrients required as all values are not fitting to any decision conditions.

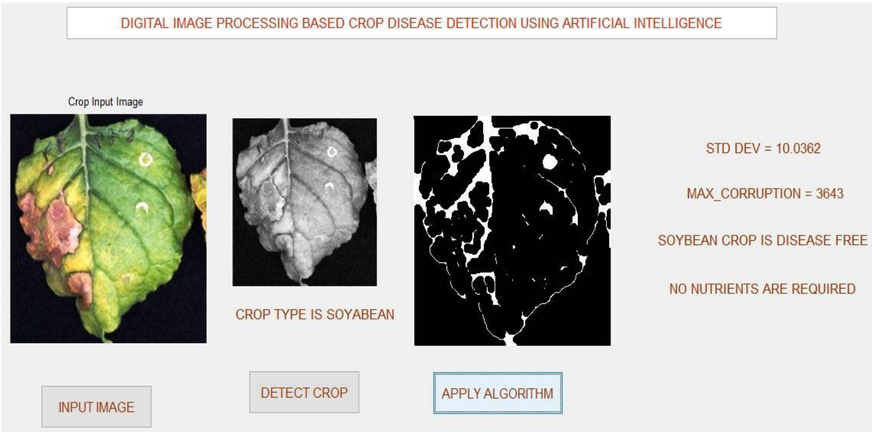


Fig. 4.11-Crop input where algorithm fails to predict the type and disease.

The graphs also plotted for the maximum corruption are for about 20 cotton leaf and 15 soya leaf. It can be observed using following graphs.

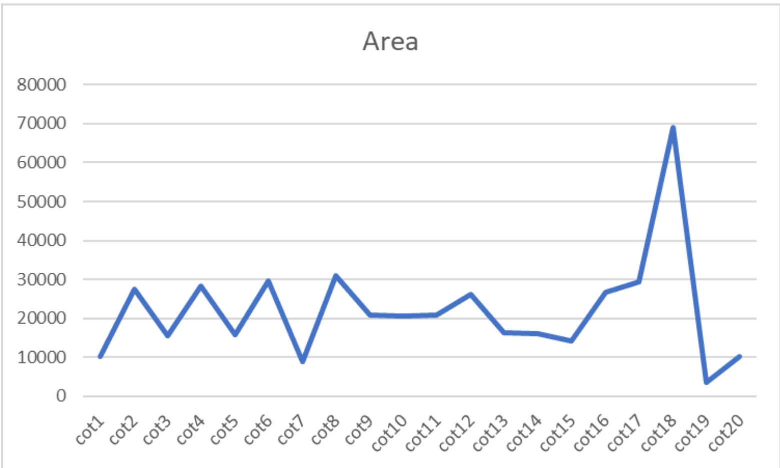


Fig. 4.12- Graph of maximum corruption of cotton crop.

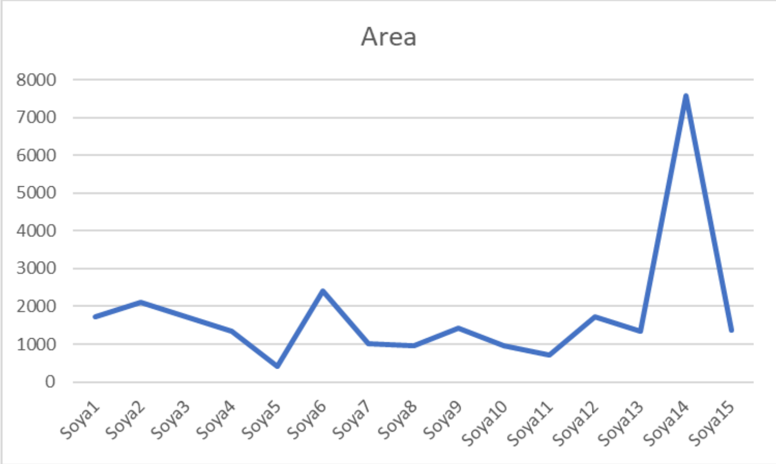


Fig. 4.13-Graph of maximum corruption of soya crop

Similarly, graph for standard deviation also plotted for both types of crop. It is as shown below.

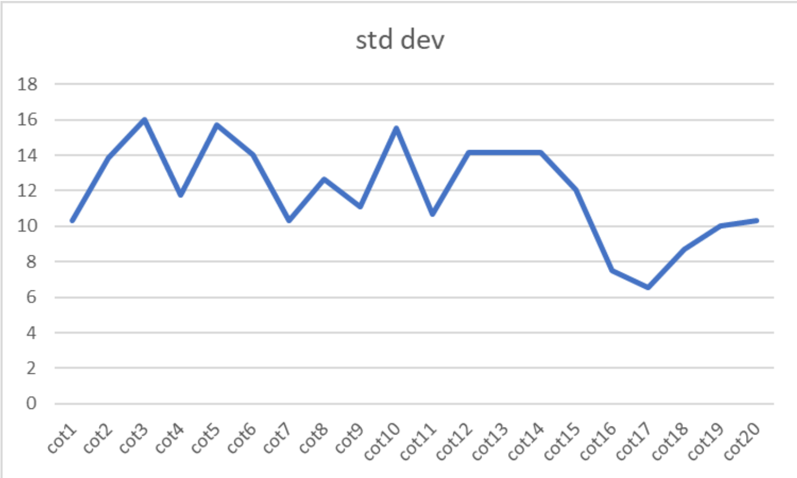


Fig. 4.14- Graph of standard deviation of cotton crop

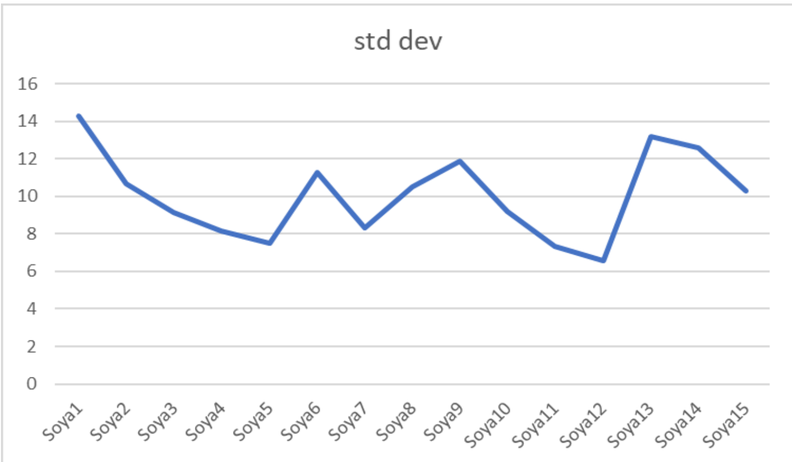


Fig. 4.15-Graph of standard deviation of soya crop

V. CONCLUSION

Plant and Leaf disease detection and classification problems are crucial and challenging problems in agriculture worldwide. The objective of this project is to recognize abnormalities that occur on plant leaf in their greenhouses or natural environment. The image captured is usually taken with a plain background to eliminate occlusion. The decision tree algorithm is used for more than 60 crop leaf of cotton and soybean crop. An algorithm predicts approximate 90 percent accuracy. This accuracy can be increased when more number of features extracted and considered to decide various decision thresholds. Thus, we have successfully developed a decision tree algorithm for detection and analysis of plant leaf disease using artificial intelligence.

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my heartfelt thanks to my guide Dr. D. D. Nawgaje and Dr. M. N. Tibdewal HOD of E & TC, for his esteemed guidance and encouragement, especially through difficult times. His suggestions broaden my vision and guided me to succeed in this work. I am also very grateful for his guidance and comments while designing part of my project and learnt many things under his leadership.

Also I would like to thanks to all teaching and non-teaching staff of E & TC department for their encouragement and suggestions for my project. My sincere thanks to all my friends, who helped me directly or indirectly in completing this project work. I would like to express my appreciation for the wonderful experience while completions of this project work.

VI. REFERENCES

[1] S. D. Khirade and A. B. Patil, "Plant Disease Detection Using Image Processing," 2015 International Conference on Computing Communication Control and Automation, 2015, pp. 768-771, doi: 10.1109/ICCUBEA.2015.153.

[2] S. C. Madiwalar and M. V. Wyawahare, "Plant disease identification: A comparative study," 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI), 2017, pp. 13-18, doi: 10.1109/ICDMAI.2017.8073478.

[3] P. Moghadam, D. Ward, E. Goan, S. Jayawardena, P. Sikka and E. Hernandez, "Plant Disease Detection Using Hyperspectral Imaging," 2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA), 2017, pp. 1-8, doi: 10.1109/DICTA.2017.8227476.

[4] S. D.M., Akhilesh, S. A. Kumar, R. M.G. and P. C., "Image based Plant Disease Detection in Pomegranate Plant for Bacterial Blight," 2019 International Conference on Communication and Signal Processing (ICCSP), 2019, pp. 0645-0649, doi: 10.1109/ICCSP.2019.8698007.

[5] G. Shrestha, Deepsikha, M. Das and N. Dey, "Plant Disease Detection Using CNN," 2020 IEEE Applied Signal Processing Conference (ASPCON), 2020, pp. 109-113, doi: 10.1109/ASPCON49795.2020.9276722.

[6] Mohanty SP, Hughes DP and Salathe M (2016) Using Deep Learning for Image-Based Plant Disease Detection. Front. Plant Sci. 7:1419. doi: 10.3389/fpls.2016.01419.

[7] R. M. Haralick, K. Shanmugam and I. Dinstein, "Textural Features for Image Classification," in IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-3, no. 6, pp. 610-621, Nov. 1973, doi: 10.1109/TSMC.1973.4309314.

[8] Breiman, L. Random Forests. Machine Learning 45, 5–32 (2001). https://doi.org/10.1023/A:1010933404324.

[9] Sanjeev S Sannakki, Vijay S Rajpurohit, V B Nargund, Pallavi Kulkarni, “Diagnosis and Classification of Grape Leaf Diseases using Neural Networks”, IEEE proceedings of 4ICCCNT, 2013.

[10] S. Arivazhagan, R. Newlin Shebiah, S. Ananthi, S. Vishnu Varthini, “Detection of unhealthy region of plant leaves and classification of plant leaf diseases using texture features”, CIGR Journal ,Vol. 15, No.1, 2013.