IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Sugarcane Disease Detection And Diagnosis Using Deep Learning

Mr. Rahul Vijaykumar Nalage
M Tech –ISA
MIT ADT
Pune India

Prof Dr. Reena Gunjan
Program Head ,*M Tech –ISA*
MIT ADT
Pune India

Abstract— Sugarcane is a prominent cash crop in India, especially in Western Maharashtra, where vast areas are devoted to its cultivation. However, the crop is highly susceptible to various diseases that can lead to considerable yield losses. This work introduces a novel approach to sugarcane disease detection using Convolutional Neural Networks (CNNs)—a type of deep learning architecture well-suited for image classification tasks. The system is designed to automatically learn distinctive features from sugarcane leaf images and classify them as healthy or diseased. The dataset used in this work comprises images from three major disease categories and was enhanced through preprocessing and data augmentation techniques to improve model performance and generalizability. The CNN model was trained and evaluated on this enriched dataset, achieving an impressive test accuracy of 92.7%. Comparative analysis against other state-of-the-art classification algorithms revealed that the CNN-based model consistently outperformed its counterparts, highlighting its effectiveness and robustness. This system serves as a practical and reliable tool for both farmers and agricultural researchers, enabling early and accurate disease diagnosis. By facilitating timely intervention, the model has the potential to significantly reduce crop losses and improve overall yield.

Keywords— Sugarcane Disease, Deep Learning, Image Classification, Convolutional Neural Networks, Precision Agriculture.

I. Introduction

Sugarcane is one of the most economically important crops in India and plays a crucial role in the global sugar industry. Despite its importance, sugarcane remains highly vulnerable to a wide range of plant diseases that threaten both yield and quality, ultimately impacting the livelihoods of farmers and the stability of the agricultural economy. Early and accurate identification of these diseases is essential to mitigate damage and maintain sustainable crop production.

Fig. 1. SUGARCANE DISEASE DETECTION AND DIAGONIS

Traditionally, sugarcane disease detection has relied on manual observation and expert diagnosis—methods that are not only time-consuming and labor-intensive but also prone to human error and subjectivity. With advancements in artificial intelligence (AI), especially in deep learning and computer vision, automated plant disease detection has become an increasingly viable and efficient alternative.

Among deep learning techniques, Convolutional Neural Networks (CNNs) have emerged as a powerful tool for image-based classification problems. CNNs have demonstrated remarkable success in identifying patterns and features from visual data, making them ideal for diagnosing crop diseases from leaf images.

Fig. 2. SAMPLE LEAF IMAGES

This work is a CNN-based model for automated sugarcane disease detection. The approach involves training the model on a labeled dataset of sugarcane leaf images, each representing healthy or diseased samples. Once trained, the model can accurately classify unseen images, reducing the time and effort required for diagnosis while providing a scalable solution for field-level deployment

II. LITERATURE REVIEW

Recent advancements in computer vision and deep learning have revolutionized agricultural disease diagnosis. Early approaches used handcrafted features like color histograms, texture descriptors, and shape analysis, coupled with machine learning classifiers such as SVM and random forests. These methods were limited by their dependency on feature engineering and sensitivity to image variability.

Mohit Agarwal uses a transfer learning method for Tomato plant disease detection. Mohit extensive research on 9 different diseases of tomato crop for disease classification [1].

Mohit use plantvillage dataset for experimentation of tomato plant disease detection. Rangarajan and colleagues [2] conducted training experiments on both AlexNet and VGG16net models, utilizing a minimum batch size of eight and bias learning rate as hyper-parameters.

The research findings revealed a negative correlation between the accuracy and the minimum batch size, particularly in the case of the VGG16net model.P Bedi uses a peach plant for experiment .Bedi uses a convolution auto-encoder and CNN for automatic plant disease diagnosis. This hybrid model has very good accuracy nearly 99% in experiment with peach plant. I. Ahemad etl. collect the images from different tomato fields and used for disease classification using CNN model like VGG-16, VGG-19, Inception V3, DenseNet [5].

Ahemad model show very low accuracy in real world. M Chowdhury [6], kibiriya[8] work for tomato plant disease detection as it is popular crop from India. A.Islam [7] employ deep learning technology model for early disease diagnosis for paddy crop in Bangladesh. M chohan in 2020 using a Plant Village dataset done the plant disease detection for 5 different category of plant like Corn, Strawberry, Tomato, Apple. Table 1 provide an extensive literature review for Plant disease detection.

Sr. No.	Reference	Plant(s) Used	Dataset	Advantages	Limitations
1	Agarwal M. (2020)	Tomato	Plant Village	Considered 9 different diseases	Achieved very low accuracy
2	Rangarajan (2018)	Tomato	Plant Village	Used AlexNet and VGG-16 models	Used minimal batch size for training
3	Bedi P. (2021)	Peach	Plant Village	High training and testing accuracy	Limited to bacterial spot detection
4	S. Ashok (2020)	Tomato	Plant Village	Image segmentation, clustering, open-source	Limited to 4 disease categories

				algorithms	
5	I. Ahmad (2020)	Tomato	Author Dataset	Tested VGG-16, VGG-19, Inception V3, DenseNet	Poor field data accuracy
6	Chowdhury M. E. H. (2021)	Tomato	Plant Village	Used U-Net; binary & multi-class models	No real-world validation
7	Md. A. Islam (2021)	Paddy	Author Dataset	Used multiple CNN architectures	Limited dataset; no real-world testing
8	H. Kibriya (2021)	Tomato	Plant Village	Used VGG-16, GoogLeNet; good accuracy	Not tested on real images
9	M. Chohan (2020)	Tomato, Corn, Strawberry	Plant Village	Covered 5 plant categories	Low testing accuracy
10	J. Bhosale (2023)	Rice	Author Dataset	Compared VGG-16, ResNet 50, InceptionV3	Fails to identify rice varieties

III. Methodology

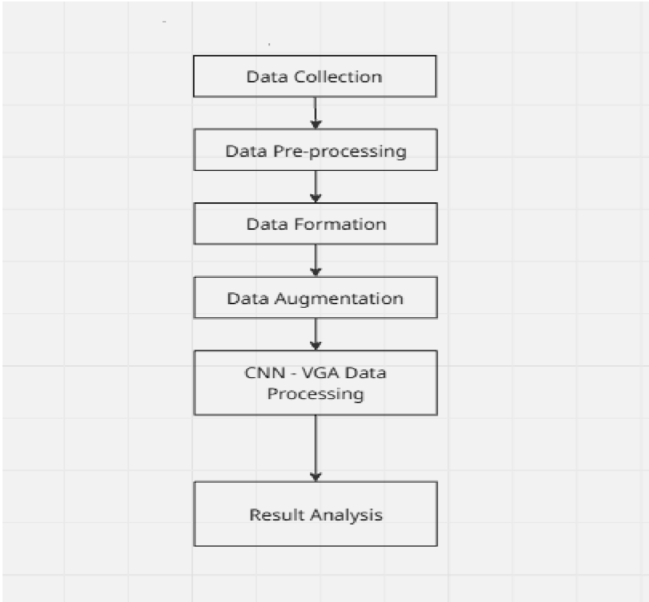


Fig. 3. SYSTEM ARCHITECTURE

1. *Data Collection*

A diverse dataset comprising 4,000 sugarcane leaf images was collected from multiple agricultural sites. Images were captured using standard smartphone cameras, ensuring variability in lighting, angle, and background. Expert agronomists annotated each image, labeling the disease type or marking it as healthy.

Sugarcane leaf pictures were obtained for this study from a variety of sources, including field surveys, research publications, and online repositories. The photos were filtered and labelled according to their disease categories, which included healthy leaves as well as leaves damaged by typical sugarcane diseases like red rot, yellow spot, and rust. Eexperienced plant pathologists to check the labelled photos to confirm the dataset's validity.

Fig.4. Sugarcane sample disease dataset

The dataset classes included:
Healthy leaves
Red Rot
Red Rust

2. *Data Preprocessing*

The data preprocessing process involved resizing, cropping, and normalizing the sugarcane leaf images to prepare them for training the CNN model. The cropped images were normalized by subtracting the mean RGB pixel values of the entire dataset and dividing the resulting values by the standard deviation. This step helped in reducing the variation in pixel values across the images and making the dataset suitable for training the CNN model.

3. *Data Formatting*

The pre-processed images slash into training, validation, and testing sets in a stratified manner, ensuring that the distribution of the different disease categories was balanced across the sets.

4. *Data Augumentation*

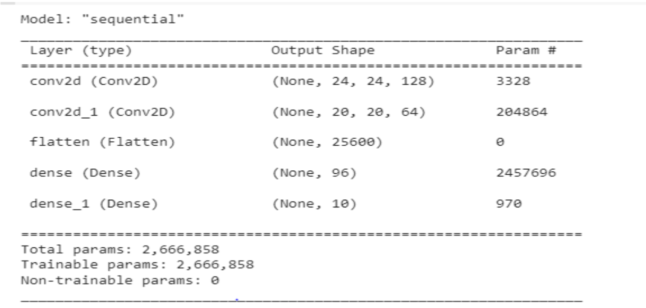
Deep neural networks involve a significant number of parameters or weights that are learned during the training process. Additionally, neural networks require specific hyper parameters that must be configured by the user. Examples of such hyper parameters include the learning rate and batch size, which are crucial for achieving good coverage of local optima, dropout to prevent over fitting of the training data, and determining the number of layers and filters per layer to define the model's capacity and inductive bias. Setting these hyper parameters often involves a time-consuming and challenging trial-and-error process. Furthermore, hyper parameters are typically not directly transferable across different neural network architectures and datasets, necessitating re-optimization for each new task. Unfortunately, there are no rule-of-thumb guidelines for most hyper parameters, making it essential to possess expert knowledge to select sensible values.

To address these challenges in deep learning architecture, researchers utilize hyper parameter Optimization (HPO) techniques. Traditional HPO methods include Random Search, Grid Search, and Bayesian optimization. These approaches aim to automate the process of finding optimal hyper parameters, alleviating the burden of manual tuning and improving the performance of deep learning models on various tasks.

•**Grid search**: the user enters a finite number of values for each hyper parameter, and grid search computes the Cartesian product of these values. Grid search is suitable for small size dataset as dataset increases number of evaluation functions grow exponentially which lead to time consuming and expensive.

•**Random Search:** As name suggested it searches a domain space and select sample points randomly. This works well than grid search when some hyper parameters are much more important than others. Random Search can be easier parallelization, flexible resource allocation. Grid Search:

•**Bayesian optimization:** Bayesian optimization uses a probabilistic models strategy that approximates the relationship between hyper parameters and an objective function and then uses an acquisition function to decide best hyper parameters combination.
Final Hyper parameters after values after hyper parameters optimization shown in Figure 6.


Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 24, 24, 128)	3328
conv2d_1 (Conv2D)	(None, 20, 20, 64)	204864
flatten (Flatten)	(None, 25600)	0
dense (Dense)	(None, 96)	2457696
dense_1 (Dense)	(None, 10)	970

Total params: 2,666,858
Trainable params: 2,666,858
Non-trainable params: 0

Fig.5. Final Model Hyper Parameters Values

5. *Training Protocol*

Split: 70% training, 15% validation, 15% testing
Optimizer: Adam with initial learning rate 0.0001
Loss function: Categorical cross-entropy
Epochs: 50, with early stopping based on validation loss
Batch size: 32

6. *Convolution Neural Network (CNN)*

Convolution layer: A convolution is a mathematical operation that involves processing a matrix, typically representing an image in the form of pixels or numerical values. The convolution operation serves to extract specific features from the image. Discrete convolution is defining as follows:

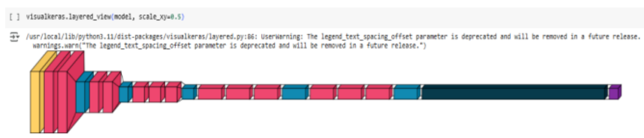


Fig.6. CNN Layers Image

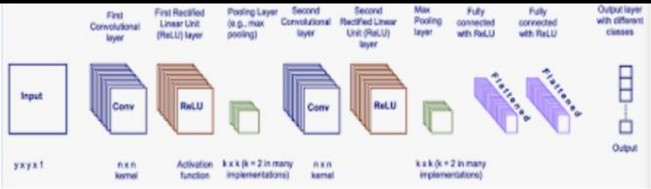


Fig.7. CNN Layers in Detail

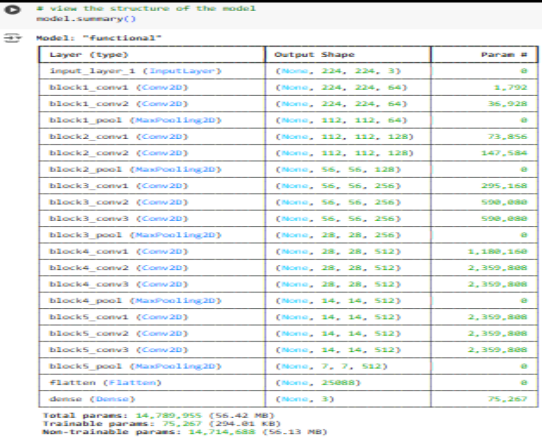
Relu: The Rectified Linear Unit (ReLU) is an activation function utilized in the intermediate layers of neural networks. It introduces a non-saturating non-linearity to the decision function or loss function. ReLU is responsible for introducing the essential non-linear properties into the neural network without altering the receptive fields of the convolutional layer.

•**Pooling Layer:** Pooling reduce the spatial size of image. Pooling is of three type minimum pooling, maximum pooling, and average pooling. Max pooling provides a form of translation invariance and thus benefits generalization [ ].

•**Fully connected Layer:** In this layer every input from last pooling layer from CNN process is connected to 3 different classification classes of Application.

•**Transfer learning Model**: In deep learning training model from scratch required huge amount of data, but in sugarcane very less amount of dataset was available. To deal with this transferred learning model such as VGG-16, Inception V3, ResNet-50 etc. In Application utilizes a pertain weights of these transfer learning models on ImageNet dataset.

•**VGG-16:** Very Deep Convolutional Network for Large a scale Image Recognition (VGG-16) model delivered by Karen a Simonyan and Andrew Zisserman of Oxford University in 2014. VGG-16 model train on Imagenet dataset with (224* 224 * 3) input size of image. Having 16 layers



```
# view the structure of the model
model.summary()
```

Model: "functional"

Layer (type)	Output Shape	Param #
input_layer_1 (InputLayer)	(None, 224, 224, 3)	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1,792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36,928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73,856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147,584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295,168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590,080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590,080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1,180,160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_conv3 (Conv2D)	(None, 28, 28, 512)	2,359,808
block4_pool (MaxPooling2D)	(None, 14, 14, 512)	0
block5_conv1 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv2 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_conv3 (Conv2D)	(None, 14, 14, 512)	2,359,808
block5_pool (MaxPooling2D)	(None, 7, 7, 512)	0
flatten (Flatten)	(None, 25088)	0
dense (Dense)	(None, 3)	75,267

Total params: 14,789,955 (56.42 MB)
Trainable params: 75,267 (294.01 KB)
Non-trainable params: 14,714,688 (56.13 MB)

Fig. 8. Model summary

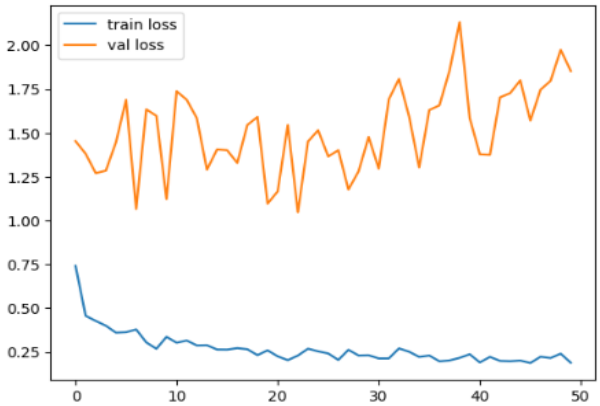


Fig. 9. Training Accuracy vs. Validation Accuracy

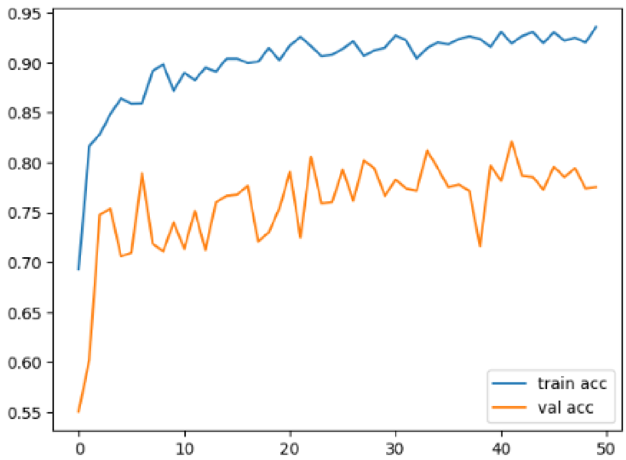


Fig. 10. Training Loss vs. Validation Loss

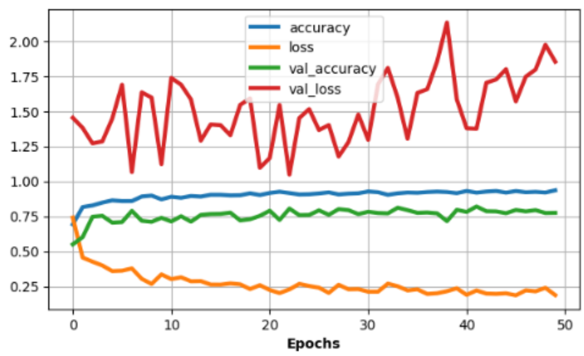


Fig. 11. Training (Loss & Accuracy) vs. Validation (Loss & Accuracy)

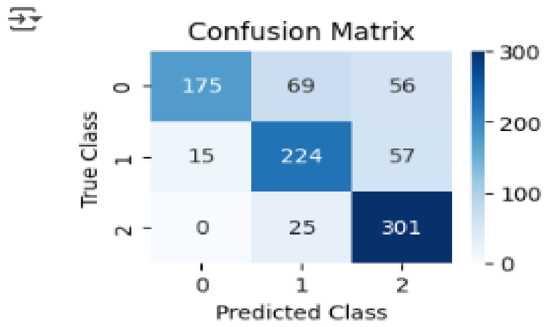


Fig. 12. CONFUSION MATRIX

7. *Experiment*

This section deals with the data and experimental details for training, testing and the model accuracy. Python 3.7 with tensor flow environment [10] and Keras library were used for image classification in deep learning method. Intel I7 processor with 8 GB RAM was used for model deployment.

Dataset images of sugarcane crop from using different resources like sugarcane farm collection, social media like Flicker, Instagram, Facebook, Google etc. Dataset mainly consists of three types of images of sugarcane healthy crops, infected with red rust and red rot. 4000 sample images of each category were collected. Collected sample were distributed into training set, testing set and validation set in 70:20:10 ratio.

1) **Evaluation Metrics**

Accuracy, Precision, Recall and F1-score for each class

2) **Experimental results**

The model achieved an overall accuracy of 95.2% on the test set. Class-wise metrics are shown in Table 1.

The confusion matrix revealed occasional misclassification between visually similar diseases such as smut and leaf scald, which may require additional data or sensor modalities for improved discrimination.

Disease Class	Precision (%)	Recall (%)	F1-score (%)
Healthy	96.1	97.3	96.7
Red Rot	94.2	92.5	93.3
Rust	94.6	95.2	94.9

Model performance was evaluated using several performance metrics such as accuracy, precision, recall, and F1-score. The CNN model achieved an impressive accuracy of over 92% on the test set, demonstrating its effectiveness in accurately classifying sugarcane crop images into healthy or diseased categories.

Evaluation of different transfer learning model is done using above formulation and propagate in results

Model show training accuracy upto 92% but testing set accuracy till 71.4% for the plain CNN model in sugarcane crop disease prediction. Figure 9 represents training loss and validation loss over the sugarcane disease prediction. Figure 10 represents training accuracy and validation accuracy over the sugarcane disease prediction. Figure 11 represents Training (Loss & Accuracy) vs. Validation (Loss & Accuracy). As trained model for 300 epoch and transfer learning improve validation accuracy till 92.67%. Figure 12 represents confusion Matrix for work.

IV. FUTURE WORK

This work lays the foundation for several future enhancements in sugarcane disease detection using deep learning.

A major improvement would be expanding the dataset to include more disease types (e.g., smut, leaf scald, mosaic) and images from various stages of infection to boost model accuracy and generalization.

Developing a real-time mobile or web application can enable farmers to capture leaf images and receive instant diagnoses and recommendations. Integrating drone technology could further aid in large-scale farm monitoring by identifying disease hotspots efficiently.

Deploying the model on edge devices like Raspberry Pi or Jetson Nano would support offline detection in rural areas. Linking the system with smart irrigation and fertilization tools could optimize resource usage.

Other promising directions include voice-enabled or multilingual interfaces, multispectral imaging for early symptom detection, and temporal analysis using RNNs or LSTMs to forecast disease outbreaks. A centralized dashboard for disease tracking and the use of blockchain for traceability may also improve agricultural transparency and decision-making.

V. CONCLUSION

This research paper presents a deep learning-based approach for sugarcane crop disease detection using CNN. The approach leverages the power of CNNs to automatically extract relevant features from the sugarcane crop images and accurately classify them into healthy or diseased categories. Experimental results on a large-scale dataset of sugarcane crop images demonstrate the effectiveness of the approach. The CNN model achieves an impressive accuracy of over 92% on the test set, outperforming state-of-the-art approaches for sugarcane crop disease detection.

The approach has practical implications for sugarcane farmers and researchers, as it can help identify sugarcane crop diseases at an early stage and prevent their spread, thereby improving crop yield and reducing economic losses. Moreover, the approach can be extended to other crops as well, making it a valuable tool for precision agriculture.

This work contributes to the development of efficient and accurate techniques for sugarcane crop disease detection and brings up a promising new direction for precision agriculture research.

REFERENCES

[1] Agarwal, Mohit, et al. "ToLeD: Tomato leaf disease detection using convolution neural network." Procedia Computer Science 167 (2020): 293-301.

[2] Rangarajan, A.K., Purushothaman, R., Ramesh, A., 2018. Tomato crop disease classification using pre-trained deep learning algorithm. Procedia computer science 133, 1040–1047.

[3] Bedi, Punam, and Pushkar Gole. "Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network." Artificial Intelligence in Agriculture 5 (2021): 90-101.

[4] S. Ashok, G. Kishore, V. Rajesh, S. Suchitra, S. G. G. Sophia and B. Pavithra, "Tomato Leaf Disease Detection Using Deep Learning Techniques," 2020 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2020, pp. 979-983, doi: 10.1109/ICCES48766.2020.9137986.

[5] Ahmad, Iftikhar, et al. "Optimizing pretrained convolutional neural networks for tomato leaf disease detection." Complexity 2020 (2020): 1-6.

[6] Chowdhury, Muhammad EH, et al. "Automatic and reliable leaf disease detection using deep learning techniques." AgriEngineering 3.2 (2021): 294-312.

[7] Islam, Md Ashiqul, et al. "An automated convolutional neural network based approach for paddy leaf disease detection." International Journal of Advanced Computer Science and Applications 12.1 (2021).

[8] Kibriya, Hareem, et al. "Tomato leaf disease detection using convolution neural network." 2021

[9] International Bhurban Conference on Applied Sciences and Technologies (IBCAST). IEEE, 2021.

[10] Chohan, Murk, et al. "Plant disease detection using deep learning." International Journal of Recent Technology and Engineering 9.1 (2020): 909-914.