International Journal of Innovative Research in Engineering
Volume 3, Issue 2 (March-April 2022), PP: 229-234.
www.theijire.com
ISSN No: 2582-8746

Plant Disease Detection Using CNN

Prof. Hitendra A.Chavan[1], Sayli Rajendra Bodare[2], Vighnesh Raju Jadhav[3], Gaurav Ganesh Moolya[4]
[1]Assistant Professor, Information Technology, BVCOE/Mumbai University, India
[2,3,4] B.E. Student, Information Technology, BVCOE/Mumbai University, India.

How to cite this paper:
Prof. Hitendra A.Chavan[1], Sayli Rajendra Bodare[2], Vighnesh Raju Jadhav[3], Gaurav Ganesh Moolya[4], "Plant Disease Detection Using CNN", IJIRE-V3I02-229-234.

Abstract: *Plant Disease Detection is very much important in agricultural field. Plant Disease Detection helps in early detection of diseases in crops and helps in developing the efficient crop yield. Without right knowledge and expertise it is difficult for farmers to do identification of plant diseases. Most of the farmers and the people who grow plants in houses, yards, etc. It is now easy to detect and diagnose the diseases in plants with the Deep Learning Technology. Deep Learning makes it to identify the plant disease and find the cure for the same by using camera for capturing images and applying several algorithms on them to get the several types of diseases in plants. This study includes the detection of multiple diseases in several varieties of plants. Convolution Neural Network is used for the detection of several plant diseases. An android application is made which take the input image from the user and displays the detected disease and the cure for that particular disease.*

Key Word: *Plant Disease Detection, CNN, Deep Learning, Mobile application, PyTorch.*

I. INTRODUCTION

Agriculture plays one of the most important role in India. The production in agriculture has significant impact on India's economy. It supports more than 50 percent of population directly. So, the early detection of diseases in plants is must to help reduce the damages that may happen. The need of disease identification is because the lack of expertise in identifying the diseases in plant growers or agriculturists. Disease detection by Humans may be wrong sometimes, But with the help of new technologies it is way faster and easier to get the accurate identification of diseases with their cures. Deep learning is one of the advanced method of machine learning that uses Neural Networks which works like the human brain. LeChun et al. have described Deep learning as a neural network learning process and one feature of Deep learning is that it can automatically obtain features from images through the patterns of images. A convolutional neural network (CNN) is a Deep learning model which is measurely used in image processing.

II. MATERIAL AND METHODS

CNN Structure :

We have trained a CNN model with 2 convolutional layers, one input layer and one output layer. The I = [i1, i2, . . . , ir] and O = [o1, o2, . . . , oh] represents input and output vectors, where, r represents the number of elements in the input feature set and h represents the number of classes. The main goal of the network is to learn a compressed representation of the dataset. In other words, we can say it tries to approximately learns the identity function F, which is defined as:

$$F_{W,B}(l) \cong 1 \qquad (1)$$

where W , B are the whole network weights and biases vectors.

A log sigmoid function is selected as activation function f in the hidden and output neurons. log sigmoid function s is a special case of the logistic function in the t space, which is defined by the formula:

$$s(t) = \frac{1}{1+e^{-t}} \qquad (2)$$

The weights of CNN network creates the decision boundaries in the feature space, and the resulting discriminating surfaces can classify complex boundaries. During the process of training, these weights are adapted for each new training image. In general, feeding the CNN model with more images can recognize plant diseases more accurately. We have used the back-propagation algorithm, which has a linear time computational complexity, for training the CNN model.

The input value Θ going into a node i in the network is calculated by the weighted sum of outputs from all nodes connected to it, as follows:

$$\theta_i = \sum(w_{i,j} * Y_j) + \mu_i \qquad (3)$$

where $\omega_{i,j}$ is weight on the connections between neuron j to i; Y_j is output value of neuron j; and μ_i is threshold value for neuron i, which represents a baseline input to neuron i in the absence of any other inputs. If value of $\omega_{i,j}$ is negative, it is tagged as inhibitory value and excluded because it decreases net input.

The training algorithm involves two phases that are forward and backward phases. During the forward phase, the networks weights are kept fixed, and the input data is propagated through the network layer by layer. The forward phase is concluded when error signal i.e e_i computations converge as follows:

$$e_i = (d_i - O_i) \quad (4)$$

where d_i and o_i are the desired i.e target and actual outputs of i^{th} training image, respectively.

In the backward phase, error signal e_i is propagated through the network in backward direction. During this phase, error adjustments are applied to CNN network's weights for minimizing e_i .

We have used the gradient descent first-order iterative optimization algorithm to calculate the change of each neuron weight $\Delta\omega_{i,j}$, which is defined as follows:

$$\Delta w_{i,j} = \eta \frac{\delta \varepsilon(n)}{\delta e_j(n)} y_i(n) \quad (5)$$

Where, $y_i(n)$ is intermediate output of the previous neuron n, η is the learning rate, and ε(n) is the error signal in the entire output. ε(n) can be calculated as follows:

$$\varepsilon(\mathrm{n}) = \frac{1}{2} \sum_j e_j^2 (n) \quad (6)$$

The CNN network has the two types of layers that are convolution and pooling layers. Each layer has a group of specialized neurons that are used to perform one of these operations. The convolution operation basically means detecting the visual features of objects in the input image such as edges, lines, color drops, etc. The pooling process helps the CNN network to avoid the learning irrelevant features of objects by focusing only on learning the essential ones. The pooling operation is applied to the output of the convolutional layers for down sampling the generated feature maps by summarizing the features into patches. Two common pooling methods that are used are average-pooling and max-pooling. In this paper, we have used the max-pooling method, which calculates the maximum value for each patch of the feature map as the dominant feature.

The output of every Conv2D and MaxPooling2D layer is a 3D form tensor (with height, width, channels). The width and height dimensions tend to shrink as we go deeper into the network. The third argument (eg. 16, 32 or 64) controls the number of output channels for each Conv2D layer. During the training phase, the CNN model generates around 4 million trainable parameters.

Implementation Platform :

FLUTTER :

Flutter is a cross-platform UI toolkit that is designed to allow code reuse across operating systems such as iOS and Android, while also allowing applications to interface directly with underlying platform services. The goal is to enable developers to deliver highperformance apps that feel natural on different platforms, embracing differences where they exist while sharing as much code as possible. Flutter comes with beautiful and customizable widgets for high performance and outstanding mobile application. It fulfils all the custom needs and requirements.

ANDROID STUDIO :

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems or as a subscription-based service in 2020. It is a replacement for the Eclipse Android Development Tools (E-ADT) as the primary IDE for native Android application development.

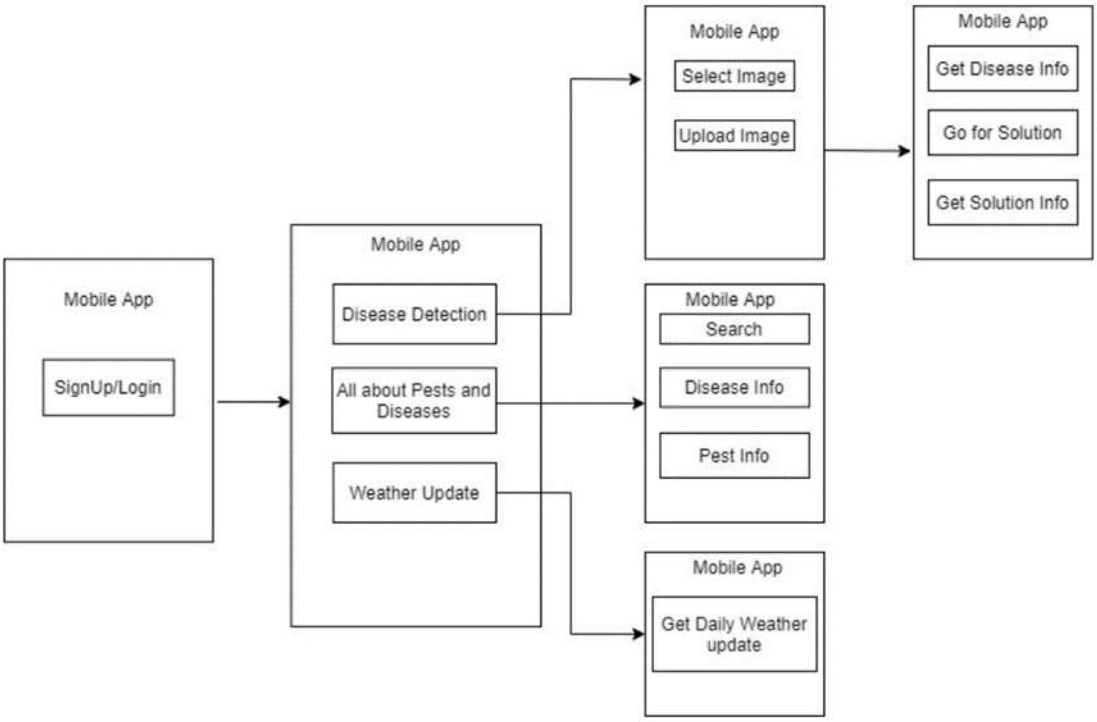
We have used Flutter for the developing the user interface of the mobile application. Flutter helps the user to develop mobile applications for Android as well as iOS. Flutter makes the user interface very attractive for the clients.

Work flow of Mobile Application :

The mobile application starts with the Welcome Page where user is given two options one is Login option and Registration option. If the user has already created a account or has already registered himself to the app he can direct go to login page and get himself logged in. Otherwise the user can get himself registered to the app.

After logging in home page will appear home page has two modules one the Detection of Plant Diseases and the other is Weather info. Plant Disease detection section has camera option and gallery option through with user can capture the image and upload the image to the server. After uploading the image the image preview page will appear. It has the send button which sends the image to the server. Server applies the processing algorithms to the image and detects the disease for the plant and then it displays the detected disease with plant name, disease name and the remedy for the detected disease.

The Weather info section provides the current status of weather according to the users live location. If the user wants know about the current weather about any other cities he can check by using the search option given. The Weather info section provide the weather information with Humidity, Temperature and the wind per km.



III.RESULT AND DISCUSSION

Dataset :

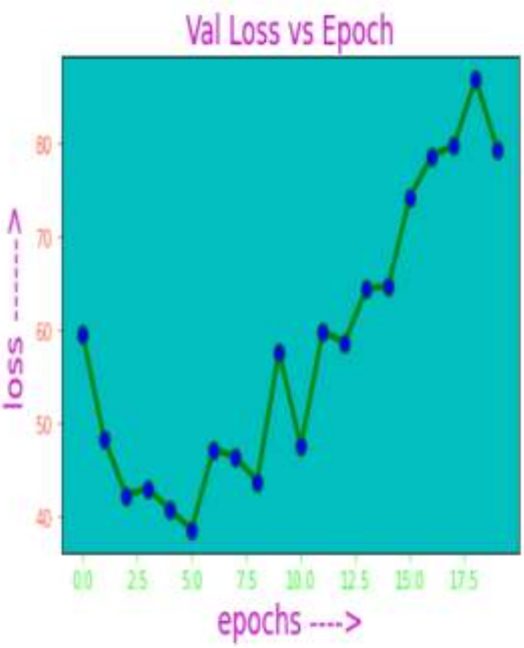
We have made a dataset of around 50 thousand images. The dataset is divided into three parts: training, validation and testing. Table shows the number of images used in the three phases across the 38 disease classes in 14 crop species. The number of images in each phase are determined based on the fine-tuned hyperparameters and structure of the CNN model.
We conducted set of controlled experiments to estimate the hyperparameters to improve the prediction accuracy and performance. In particular, we progressively tested some random combinations of hyperparameter values until we achieved satisfactory results. To increase the training accuracy and to minimize training loss of the CNN model, we applied some series of image pre-processing transformations to the training dataset.

Plant Disease Classes	Training	Validation	Testing	Total
Rice blast	80	80	40	200
Rice tungro	80	80	50	210
Rice blight	80	80	40	200
Rice healthy	85	85	50	220
Sugarcane Healthy	100	100	60	260
Sugarcane Bacterial	100	100	60	260
Sugarcane Red Rot	100	100	60	260
Wheat healthy	607	279	150	1036
Wheat brown rust	319	226	100	645
Wheat yellow rust	319	232	200	651
Tomato target spot	1680	457	300	2437
Tomato yellow rust	912	490	290	1692
Tomato bacterial spot	1184	425	300	1909

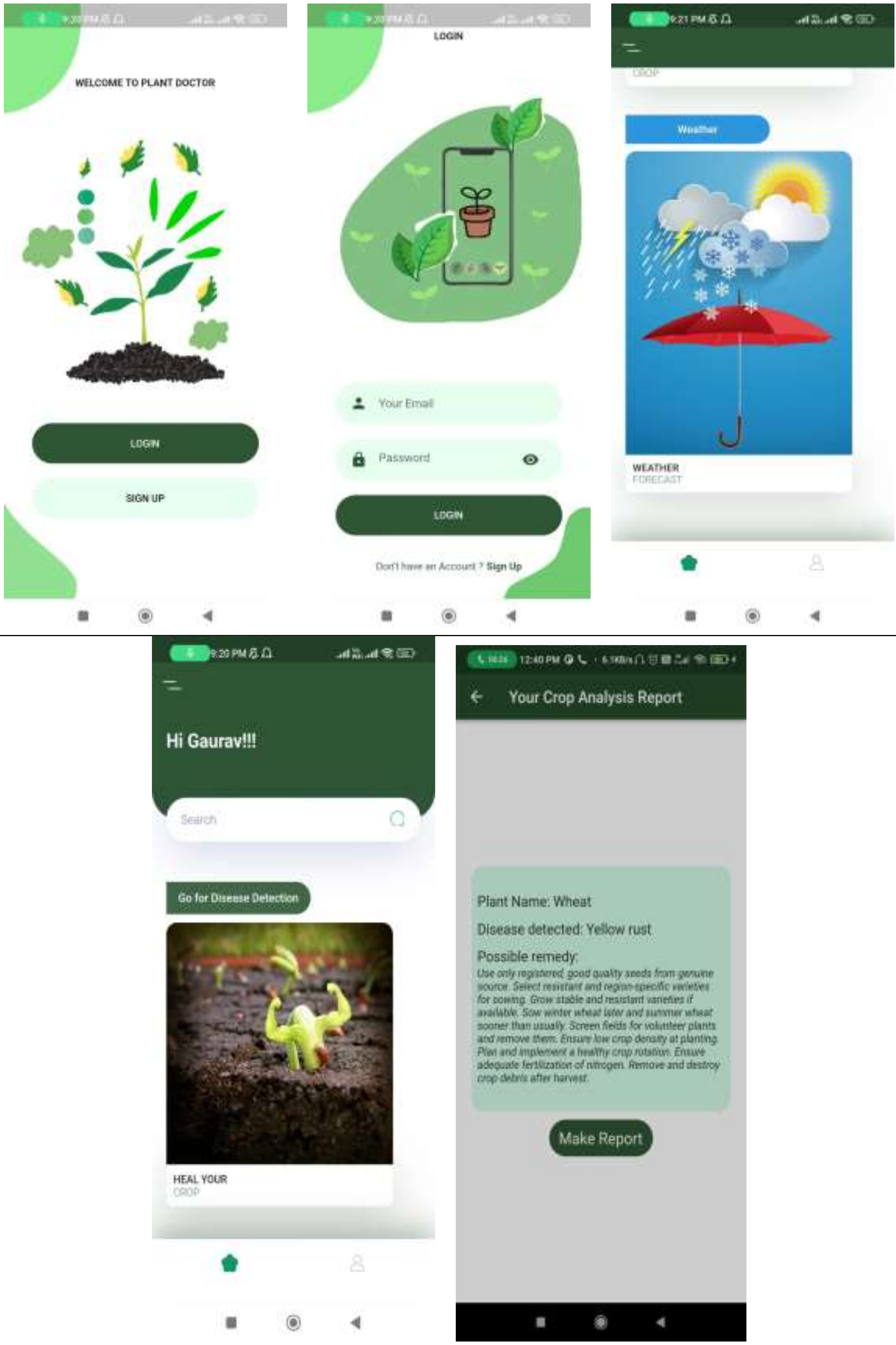
Tomato mosaic virus	1184	448	200	1832
Tomato early blight	1636	480	300	2416
Tomato septoria leaf spot	912	436	300	1648
Tomato leaf mold	912	470	250	1632
Tomato healthy	1440	481	300	2221
Tomato late blight	1851	463	300	2614
Tomato two spotted spider mite	1741	435	300	2476
Pepper healthy	1904	507	300	2711
Pepper bacterial spot	624	478	200	1402
Potato early blight	1680	487	300	2467
Potato late blight	1939	485	300	2724
Potato healthy	1440	456	200	2096
Corn healthy	1680	465	300	2445
Corn common rust	956	477	340	1773
Corn northern leaf blight	1440	477	350	2267
Corn cercospora leaf spot grey leaf spot	624	410	400	1434
Total				44,144

Epochs:
We examine different no of epochs for adam optimizer to check for better accuracy. We observed the same accuracy at around 19 epochs while training epochs, after which there was no noticeable change in the loss. As a result, we'll have to stop iterating at 20.

```
Epoch 0 total_correct 97 loss: 59.59567320346832
Epoch 1 total_correct 235 loss: 48.41190576553345
Epoch 2 total_correct 260 loss: 42.32168257236481
Epoch 3 total_correct 250 loss: 43.08985352516174
Epoch 4 total_correct 329 loss: 40.81555938720703
Epoch 5 total_correct 280 loss: 38.664438366889954
Epoch 6 total_correct 283 loss: 47.24228870868683
Epoch 7 total_correct 247 loss: 46.44645130634308
Epoch 8 total_correct 296 loss: 43.852198362350464
Epoch 9 total_correct 286 loss: 57.68193054199219
Epoch 10 total_correct 362 loss: 47.5207736492157
Epoch 11 total_correct 320 loss: 59.821959137916565
Epoch 12 total_correct 358 loss: 58.61583399772644
Epoch 13 total_correct 218 loss: 64.38009786605835
Epoch 14 total_correct 354 loss: 64.65620458126068
Epoch 15 total_correct 358 loss: 74.22239434719086
Epoch 16 total_correct 370 loss: 78.51256108283997
Epoch 17 total_correct 369 loss: 79.66621744632721
Epoch 18 total_correct 382 loss: 86.67768430709839
Epoch 19 total_correct 389 loss: 79.15445637702942
```



Mobile App results :



IV.CONCLUSION

The detection and classification of the plant diseases is very important for the successful cultivation of crop and this can be done using image processing. This paper discussed automated technique to segment the disease part of the plant. In the existing system includes only the detection of diseases. And in our proposed system we will be providing the solution/cure to the diseased plants. The proposed system is implemented on the Mobile Application.

References

[1]. *Shambhuraj Porob, Gajant Naik, Dheeraj Amonkar, Rahul Patil, Prof. Harish Velingkar, Prof. Prathamesh Bhat "Plant Health Monitoring using Digital Image Processing" International Journal of Emerging Trends in Engineering and Development Year 2017.*

[2]. *Mrs. G. Saranya, Deepthi M, Bhuvaneshwari M, Gomathy R "PLANT DISEASE DETECTION USING IMAGE PROCESSING" International Journal of Scientific Research and Review Year 2019*

[3]. *Saradhambal G , Dhivya R , Latha S , R. Rajesh "PLANT DISEASE DETECTION AND ITS SOLUTION USING IMAGE CLASSIFICATION" International Journal of Pure and Applied Mathematics Year 2018*

[4]. *Mr. Ashish Nage, Prof. V.R. Raut "Detection and Identification of Plant Leaf Diseases based on Python" International Journal of Engineering Research & Technology (IJERT) year 2019*

[5]. *Tejaswini Devram, Komal Hausalmal, Juby Thomas, Pranjal Arote, S.P.Pattanaik "DISEASE DETECTION OF TOMATO PLANT LEAF USING ANDROID APPLICATION" International Engineering Research Journal (IERJ) Year 2017*

[6]. *Vijai Singh, A.K. Misra "Detection of plant leaf diseases using image segmentation and soft computing techniques" INFORMATION PROCESSING IN AGRICULTURE 4 (2017) 41–49 Year 2016*

[7]. *Naveen Chandra Gowda, Sunil Kumar, Subham Majumbdar, Koneti Naga Abhishek, Parikshit Sarode "International Journal of Engineering and Advanced Technology (IJEAT)" year 2019*

[8]. *Sharath D M, Akhilesh, S Arun Kumar, Rohan M G and Prathap C "Image based Plant Disease Detection in Pomegranate Plant for Bacterial Blight" International Conference on Communication and Signal Processing Year 2019*

[9]. *Mr. V. Suresh, D Gopinath, M Hemavarthini, K Jayanthan, Mohana Krishnan "Plant Disease Detection using Image Processing '' International Journal of Engineering Research & Technology (IJERT) Year 2020*

[10]. *Sammy V. Militante , Bobby D. Gerardoij , Nanette V. Dionisio "Plant Leaf Detection and Disease Recognition using Deep Learning" IEEE Eurasia Conference on IOT, Communication and Engineering Year 2019*

[11]. *Melike Sardogan, Adem Tuncer, Yunus Ozen Plant Leaf Disease Detection and Classification Based on CNN with LVQ Algorithm "International Conference on Computer Science and Engineering" International Conference on Computer Science and Engineering Year 2020*

[12]. *Ms. Kiran R. Gavhale , Ujwalla Gawande Plant Leaves Disease detection using Image Processing "IOSR Journal of Computer Engineering (IOSR-JCE)" Plant Leaves Disease detection using Image Processing Year 2014*

[13]. *B. Rajesh, M. Vishnu Sai Vardhan, L. Sujihelen "Leaf Disease Detection and Classification by Decision Tree" Proceedings of the Fourth International Conference on Trends in Electronics and Informatics (ICOEI 2020) Year 2020*

[14]. *Achyut Morbekar, Ashi Parihar, Rashmi Jadhav "Crop Disease Detection Using YOLO" International Conference for Emerging Technology (INCET) Year 2020*

[15]. *Eftekhar Hossain, Md. Farhad Hossain and Mohammad Anisur Rahaman "A Color and Texture Based Approach for the Detection and Classification of Plant Leaf Disease Using KNN Classifier" Year 2020*