# *Plant disease detection using Artificial Intelligence*

**Computer Engineering Informatics and Communications**

University of Zimbabwe

**Group Members**

| *Full names* | *Reg Number* | *Program* |
|---|---|---|
| Kennedy Sithole | R204992P | HCS |
| Tinotenda Nyamande | R192004C | HCS |
| Takunda M Gatakata | R219042J | HDSC |
| Chantel Munoyenda | R206399G | HDSC |


## Abstract

Plant diseases can significantly reduce crop yields and quality, leading to significant economic losses for farmers and food shortages for consumers and even for the country as a whole. Early detection and diagnosis of plant diseases is essential for effective disease management and control. However, traditional methods of disease detection, such as visual inspection by human experts, can be time-consuming, highly costing, and prone to errors. Therefore, there is a growing interest in developing automated and accurate plant disease detection systems using artificial intelligence (AI) and machine learning (ML) techniques despite that most of these systems either use millions of training parameters. In this research, we aim to develop an AI-based plant disease detection system that can accurately and efficiently identify plant diseases in real-time. Specifically, we will explore various AI and ML techniques, including deep learning, convolutional neural networks (CNNs), and image segmentation, to identify and classify plant diseases based on images of diseased leaves. This research's evaluation is based on the performance of a dataset of tomato plant images with and without diseases.

# 1. Introduction

Agricultural country, Zimbabwe, a well-known bread basket of Africa, and a major part of its economy depends on the agricultural sector like textile Industry, sugar industries. The share of agriculture in the Zimbabwe Gross Domestic Product (GDP) are 8.85% 2021(Global economy). Over 70% population of Zimbabwe depends on the agricultural sector either directly or indirectly. Therefore, disease-free good quality crop production is essential for the growth of the country's economy at large. Same as human beings, plants also suffocate from various kinds of diseases in their different stages. Consequently, the total crop yield and hence the net profit of the farmers are adversely affected. In order to address this issue, the early detection of plant diseases is a necessity. Manual disease detection in plants is done either by farmers or by agricultural scientists for example middlemen. However, this is very challenging and time consuming since human experience is not precise at a large scale. To curb this problem, many researchers across the globe presented different computer systems for automatic plant disease detection with the help of various Machine Learning and deep learning techniques. These computer systems use a very high number of training parameters. The training time and the prediction time of these systems are very high, or they require a machine with high computation powers to make the systems more accurate. This research work tries to reduce the number of features used for prediction using the CNN (Convolutional Neaural Network) network with a close accuracy of plant disease detection. This, in turn, reduces the quantity of training parameters by a significant factor resulting in the reduction of training and prediction time.

This paper provides a comprehensive review of recent research on plant disease detection using Artificial Intelligence, including the techniques used, the data-sets used for training and testing, the accuracy rates achieved, and the challenges and opportunities for future research. The paper also discusses the potential applications of AI-based systems in the agriculture industry and their implications for food security and sustainable agriculture for the country's economy.

# 2. Literature review and Related Work

There exists many researchers who have published papers for identifying and classifying the plant diseases specifically for recognizing the tomato leaf diseases. In Hanson, A.M.J., Joy, A., Francis, J 2017, a CNN-based model is used to detect various plant leaf diseases. We will summarize some of the papers as follows. Many researchers found that LeNet gives the best result in terms of accuracy after they used various state-of-the-art CNN architectures such as AlexNet, GoogleNet and LeNet (LeCun, Y., et a, 1989) for the detection and clasification of Tomato leaf diseases.

The authors Sladojevic.S, Arsenovic.M, Anderla.A, Culibrk. D and Stefanovic. D (p3, 2016) work on a research and discovers that CNN are crucial in the image classification and detecting features as they say,"... enabling the model to distinguish between diseased leaves and healthy ones or from the environment by using deep

CNN ”.  Punam Bedi, Pushkar Gole (2021) proposed a Deep Convolutional Neural Network system for seasonal crops disease identification. They proposed a model which obtained compressed domain representations of leaf images using the encoder network of CAE and then used the compressed domain representations for classification using CNN. They claimed that due to dimension reduction using CAE, the number of features, and hence the number of training parameters reduced significantly. To test the model, it was applied to detect Bacterial Spot disease in peach plants. The model achieved 99.35% training accuracy and 98.38% testing accuracy by using only 9,914 training parameters. This effectively proved the efficiency of the CNN as it requires fewer training parameters yet having great and almost accuracy disease detection.

Islam.M.Z (2019) proposed a CNN-based technique for detecting common tomato plant diseases, that are as follows, bacterial spot,  early blight, late blight, septoria leaf spot, and yellow leaf curl virus. Thy gathered a dataset of tomato leaf images with the above specified diseases they trained a CNN to classify the images into one of the above categories. Their model achieved 96% and over of training accuracy on their test dataset. In addition S. Ghosal(2018) also collected dataset of tomato leaf images with these diseases and trained a CNN to classify the images into one of the three categories and the research has an accuracy of over 90% on the test dataset.

Random forest is one of the most used algorithms used in plant disease detection due to its efficiency and accuracy. Several studies have been conducted to detect plant diseases using random forest. Chen et al (2018) developed a model to detect plant diseases using a dataset of 54000 images of 17 crop classes which achieved an accuracy of 97.13%.

Mishra et al (2021) created a model using a dataset of 38260 images of 15 different classes which achieved a 96.66% accuracy using random forest. Some studies have integrated the random forest classifier with sensors to detect diseases in plants. A system was developed by Li et al (2019) that used a spectral sensor with the random forest algorithm to detect powdery mildew on cucumber leaves, achieving a 91.9% accuracy. Anderson et al (2017) used a hyper-spectral imaging system together with random forest algorithm to detect fungal infection in barley plants.

Other researchers have combined the random forest algorithm with other algorithms to try and improve its accuracy. Zaman et al (2020) combined Local Binary Pattern features with random forest to detect diseases in potato plants, achieving an accuracy 98.57% which was higher compared to other algorithms.

Hatuwal et al (2021) did a comparison of CNN, support vector machine, random forest and KNN to determine the most accurate model image classification. CNN produced a higher accuracy compared to other models but the time and resource requirements were high. The table below shows a summary of the results

| Model Name | Accuracy |
| --- | --- |
| Convolutional Neural Network | 97.89% |
| Random Forest | 87.43% |
| Support Vector Machine | 78.61% |
| K Nearest Neighbor | 76.96% |

Guptal et al (2022) also did an image classification research using random forest classifier.

| | |
| --- | --- |
| Accuracy | 97.33% |
| Precision | 97.00% |
| Sensitivity | 98.9795 |
| Specificity | 94.2304 |

.

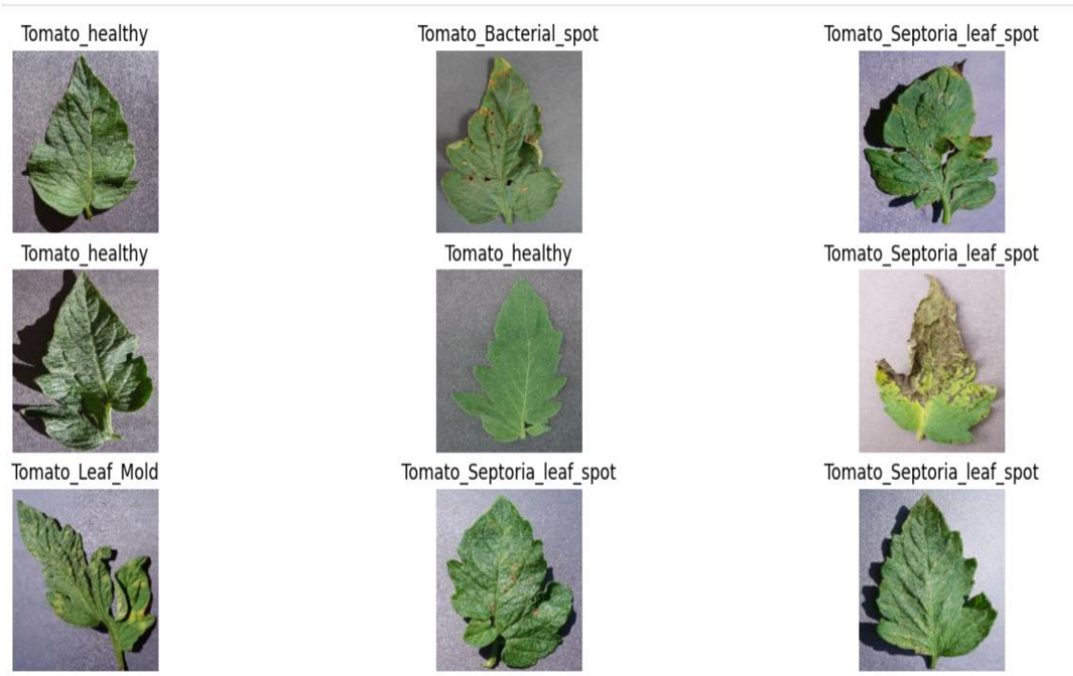
The results of the research saw a 97.33% accuracy and some of the metrics are summarized above.

# 3. Methodology

This section describes the stages of how the model recognizes the tomato leaves. The complete process in developing the plant disease detection model as follows : data acquisition is which is the collection images as data-sets ,image preprocessing, feature extraction and model training and testing as well as disease recognition.

## 3.1 Data acquisition

Datasets are the most required raw materials to be used when one want to  develop a project from training phase to the evaluation stage. All the images collected for this dataset were downloaded from the internet from Kaggle. Those images used  for this research consist of various tomato plant leaves which have been divided into eight categories to reflect their different conditions for easy identification of diseased leaves from healthy leaves. The dataset consist of 6952 images of tomato plant leaves. Below are some sample images and  their relative labels since it's a supervised data :




## 3.2 Data Preprocessing or data augmentation

Data preprocessing involves the preparation of raw data to make it more suitable for a machine learning model. This includes resizing , normalization: colour correction and noise reduction. The inputted images are of 256 x 256 sizes.

Useful geometric transformation on data preprocessing that maps the position of every object in the image to the new location of the final output image which is image shifting. For instance if an object is at the position of x,y, it gets shifted to new position y,x in the new image as shown below. Where dx and dy are the respective shifts along with the different directions. This will generalize the model and give a more variety to the model. The equations below illustrates how the images are translated with respect to  Fig 3.2.1a shows the transformation.

(x,y) -------shifting---------> (y,x)

$$X = x + dx$$

$$Y = y + dy$$

Fig 3.2.1a

Original Image Shifted image

Another popular preprocessing techniques we have employed is the noise removal. That's when we were adding noise to the images for the model to know when to separate the signal from the noise the images can have. As show below on Fig.3.2.1b we have the original image and other noised image.

Fig. 3.2.1b

Original Image before noising Image after image Noising

To make sure our model is capable of assessing the images in many forms, we also accounted for the image blurring as another preprocessing technique which involves blurring the original images as some of the images might be of higher quality and others might be very bad with low quality. It actually removes high frequency content for example noise and edges from the image resulting in edges being blurred when this is filter is applied. This technique is implemented in the project through using OpenCV. As shown by the Fig.3.2.1d below, we have the original image and the blurred image of tomato dataset.

Original image After Conversion

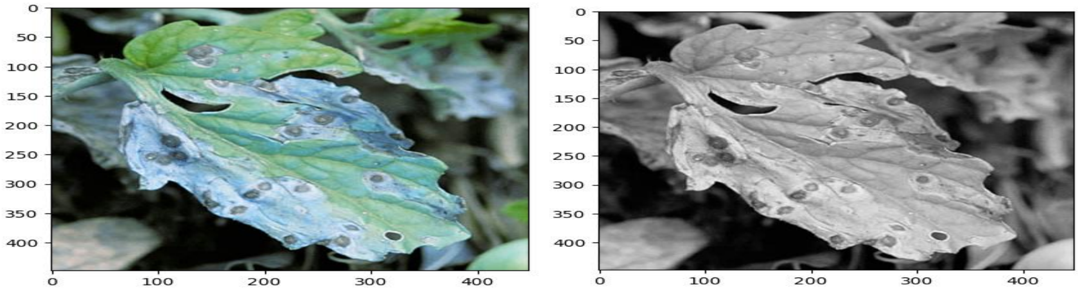



Table 1. Dataset augmentation stages descriptions of tomato leaf disease recognition

| ***Leaf Class name*** | ***Org. I mages*** | ***Augmentation techniques*** | | | ***Augmented images*** |
|---|---|---|---|---|---|
| | | Flipping | Rotation | Shifting | |
| Early_Blight | 1000 | 2000 | 2000 | 2000 | 7000 |
| Late_Blight | 1000 | 2000 | 2000 | 2000 | 7000 |
| Leaf_Mold | 952 | 1904 | 1904 | 2000 | 6460 |
| Septoria_leaf_spot | 1000 | 2000 | 2000 | 2000 | 7000 |
| Tomato_healthy | 1000 | 2000 | 2000 | 2000 | 7000 |
| Spider_mites_Two _spotted_spider_m ite | 1000 | 2000 | 2000 | 2000 | 7000 |
| Bacterial_spot | 1000 | 2000 | 2000 | 2000 | 7000 |
| **Total** | **6952** | **13904** | **13904** | **13904** | **48460** |

To ensure the accuracy of the neural network model, we need bigger amount of data. As Shorten, C., Khoshgoftaar, T.M, 2019 mention about data augmentation, we implemented the image data augmentation techniques as discussed above to enhance the dataset with slight distortion. To prepare the augmented data, we use 1000, 1000, 1000, 952, 1000, 1000 and 1000 original images of Tomato_Bacterial_spot, Tomato_Spider_mites_Two_spotted_spider_mite, Tomato_healthy, Tomato_Leaf_Mold,Tomato_Early_Blight ,Tomato_Late_Blight and Tomato_Septoria_leaf_spot respectively, and applied four types of augmentation techniques and it's summary is shown in table 1 above. The above data augmentation techniques were adopted through the use OpenCV library (Brahmbhatt, S, 2013).

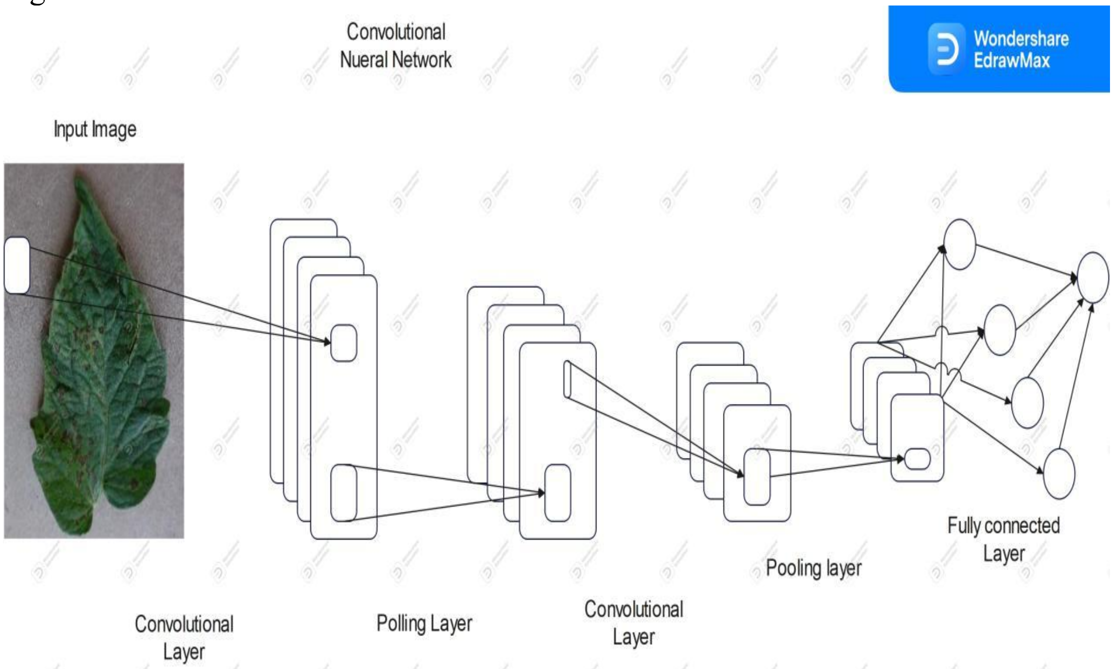
## 3.3 Feature extraction

This section simply tells much more about the feature engineering or extraction on the processed data of different images of tomatoes. That when we can use the *Support Vector Machine (SVM),* Convolutional Neural Networks (CNN) under Tensor Flow, K-Nearest Network as the Random Forest algorithms.

3.4.1 Convulutional Nueral Networks

The Convolutional Neural Network (CNN) model is trained using the preprocessed dataset. These CNNs uses classification as the Machine Leraning techniques. As Sarkakar.G, Paril.G and Dutta.P (p53,2021) says the classification is the process of finding a model that separates input data into multiple discrete classes or labels. The training process involves feeding the model with input images and their respective labels (Tomato_healthy or Tomato_Late_blight) and adjusting the model parameters to minimize the prediction error and maximize the prediction accuracy of the models. This technique uses convolutional , Rectified Linear Unit and pooling processes. The Fig.3.4.1a below shows the stages on how the features were extracted from the leaves.

Fig.3.4.1a




Input image is just the original image of size $W_0 \times H_0$, where $W_0$ is the width(256) and $H_0$ is the height(256) of the image, respectively.

As soon as the data-centric steps are done, the images are represented in the form of matrices. These are input layer, convolution layer 1, max pooling layer, convolution layer 2, max pooling layer 2, convolution layer 3, max pooling layer 3 and an output layer as shown in Fig.3.4.1a above So before CNN convoluational layer, the algorithm starts by introducing padding which is used to preserve the size of the image as it passes through the network by adding extra pixels to the images and reducing the border effect when CNN reduces the dimensions of the images. Padding is used to expand the input matrix by appending the layers of zeroes to the input matrix's border (Punam Bedi, Pushkar Gole, 2021).

The convolution layer is the heart of the Convolutional Nueral network where linear operation used for feature extraction takes place, where a small array of numbers, called a kernel, is applied across the input, which is an array of numbers, called a tensor. Since we used three channels on this research.

Table 2. Below are the parameters of our CNN-based model for detecting rice leaf diseases

| ***Layers*** | ***Functions*** | ***Pools*** | ***Filter*** | ***Output*** | ***Parameters*** |
|---|---|---|---|---|---|
| *Input* | ----- | ----- | ----- | 256 x 256 | 0 |
| *Conv1* | Convolution | 3 x 3 | 32 | 32 x 254 x 254 | 896 |
| *Pooling 1* | Max pooling | 2 x 2 | ----- | 32 x 127 x 127 | 0 |
| *Conv2* | Convolution | 3 x 3 | 32 | 32 x 125 x 125 | 18496 |
| *Pooling 2* | Max pooling | 2 x 2 | ----- | 64 x 62 x 62 | 0 |
| *Conv3* | Convolution | 3 x 3 | 64 | 64 x 60 x 60 | 36928 |
| *Pooling3* | Max pooling | 2 x 2 | ----- | 64 x 30 x 30 | 0 |
| *Conv4* | Convolution | 3 x 3 | 64 | 64 x 28 x 28 | 36928 |
| *Pooling4* | Max pooling | 2 x 2 | ----- | 64 x 14 x 14 | 0 |
| *Conv5* | Convolution | 3 x 3 | 64 | 64 x 12 x 12 | 36928 |
| *Pooling5* | Max pooling | 2 x 2 | ----- | 64 x 6 x 6 | 0 |
| *Conv6* | Convolution | 3 x 3 | 64 | 64 x 4 x 4 | 36928 |
| *Pooling6* | Max pooling | 2 x 2 | ----- | 64 x 2 x 2 | 0 |
| *Dense* | ------ | ------ | ------- | 1 x 1 x 64 | 16448 |
| *Dense1* | ------- | ------- | ------ | 1 x 1 x 7 | 455 |
| *Output* | Fully connected | ------ | ----- | *1 x 1 x 7* | *0* |
| **Total trainable parameters** | | | | | 184, 007 |

As shown above, we implemented nonlinear activation function ReLU as explained below. We use 32, 64, 64, 64, 64 and 64 filters for Conv1, Conv2 ,Conv3, Conv4, Conv5 and Conv6 respectively.

Rectified Linear Unit : This sets all the negative values of the feature to zero and enhancing the network to more complex and meaningful representation of the input images of tomatoes. It overcome the problems of vanishing gradients as it has the constant gradient positive input values and does not saturate ah high or low values hence helping to distinguish between healthy and diseased features on the leaves. This function learns the parameters of rectifiers and improves accuracy at negligible extra computational cost. It is defined as

$$f(z_i) = \max(0, z_i),$$

where $z_i$ represents the input of the nonlinear activation function on the the channel.(Srdjan Sladojevic, Marko Arsenovic, Andras Anderla , Dubravko Culibrk, and Darko Stefanovic, 2016).

Pooling layer helps by reducing the dimensionality of the feature map whilst maintaining the most significant features from the leaves and this will reduce the computational burden and the risks of over-fitting. We actually implemented max pooling on this research. For instance our activation map where of size W x W x D, a pooling kernel of spatial size F, and stride S (the size of the sliding kernel), then the size of output volume can be determined by the following formula as :

$$\mathbf{W}_{out} = \frac{\mathbf{W - S}}{\mathbf{S}} + \mathbf{1}$$

This will yield an output volume of size $W_{out}$ x $W_{out}$ x D.

Fully connected Layer : The output feature map of the final pooling layer is typically transformed into one-dimensional array of numbers as a 1 × 64 vector and connected to one or more fully connected layers which are called dense layers where every input is connected to every output by trained weight as shown the Fig.3.4.1a above. That feature map is the passed into another dense layer and be a vector of 1 x 7 then finally the Soft-max layer will finalize the classification of the diseases into bacterial spot or healthy leaves.

### 3.4.2 Random Forest

Random forest is a supervised machine learning algorithm that makes use of the training dataset to build a set of decision trees.Random forest is based on the ensemble learning concept where multiple classifiers are combined in solving a problem.The dataset is split to form several decision trees on various datasets of the given dataset.

A decision tree is a supervised machine learning algorithm with a tree-like structure where the root node represents the dataset and each internal node represents the features and the leaf nodes represents the results of the classifier. The decision to split the data along certain features is made by calculating the information gain for each feature.Information gain refers to the measure of reduction in impurity achieved by splitting the dataset along a particular feature and can be calculated using the formula :

$$\text{Information Gain(H, A)} = H - \sum \frac{|H_V|}{|H|} H_v$$

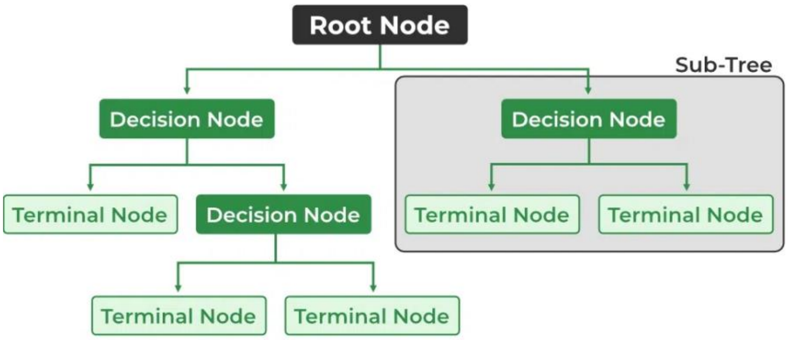
where

- A is the specific attribute or class label
- |H| is the entropy of dataset sample S
- |Hv| is the number of instances in the subset S that have the value v for attribute A

Entropy refers to the measure of the degree of randomness or uncertainty in the dataset which can be calculated using the below formula :

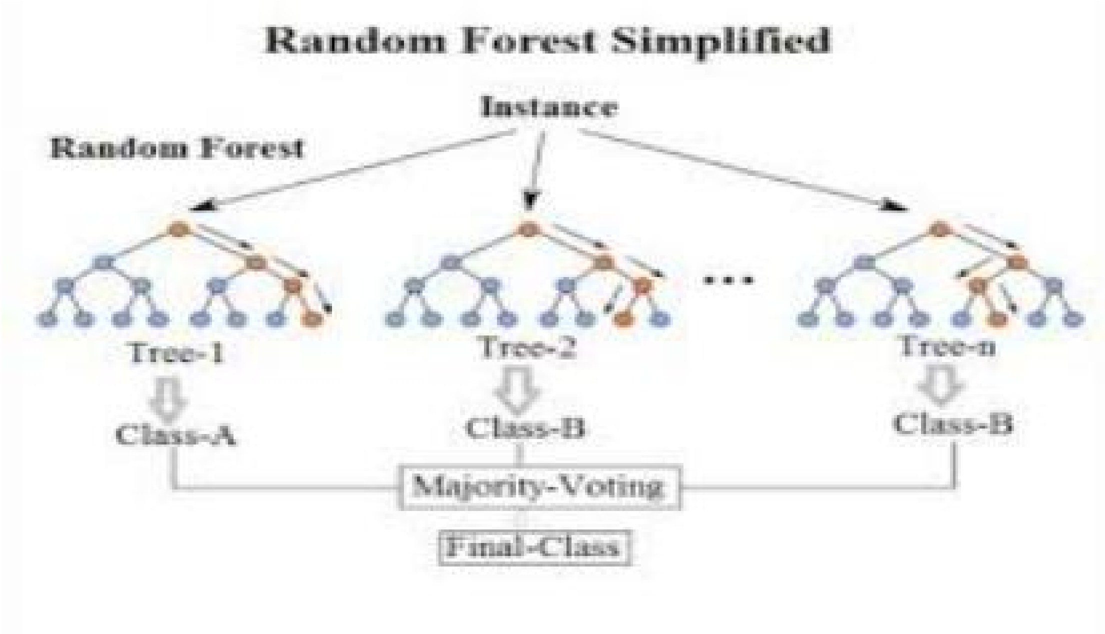
$$H_i = -\sum_{k \epsilon K}^{n} p(i,k) \log_2 p(i,k)$$

The working of the decision tree is illustrated by the diagram below:




The random forest classifier is created by creating several decision trees and combining their results.To classify data, the random forest classifier takes into account the prediction of all the decision trees and makes the final classification based on the majority of votes of predictions from the trees.

The diagram below illustrates how the random forest classifier works:




For this research , the dataset is split into 100 decision trees. Having many decision trees usually results in higher accuracy. The max_depth which determines the maximum height up to which trees can grow inside a forest has been set to None which means the nodes will continue to grow until all leaves become pure.The accuracy of the random forest model usually grows as the depth increases .

# 4. Experimental Results and Discussion

This chapter illustrates our experiments with possible diagrams and graphs as result analysis of our research.

Coding Environment :

The experiment is conducted on an Intel(R) Core(TM) i5-4300U CPU @2.50 GHz Processor with 8 GB RAM. The whole model was developed using Python with Keras and TensorFlow as packages on Windows 10 Operating System.

The dataset was splitted into the training dataset, validation and the test dataset as follows :

Training_dataset = Plant_Disease * 80/100
Validation_data = Plant_Disease * 10/100
Test_dataset = Plant_Disease * 10/100

Table 3 : The Summary of training, validation and testing datasets

| *Leaf Class name* | *#Training Images* | *#Validation Images* | *#Testing Images* |
|---|---|---|---|
| Early_Blight | 800 | 100 | 100 |

| | | | |
|---|---|---|---|
| Late_Blight | 800 | 100 | 100 |
| Leaf_Mold | 761 | 95 | 95 |
| Septoria_leaf_spot | 800 | 100 | 100 |
| Tomato_healthy | 800 | 100 | 100 |
| Spider_mites_Two_spotted_spider_mite | 800 | 100 | 100 |
| Bacterial_spot | 800 | 100 | 100 |
| **Total** | **5561** | **695** | **695** |

Several models were trained and tested with this data. By reserving a portion of data for testing, this can allow us to evaluate the model's performance by reducing the inaccuracy predictions even for unseen data and ensure that the model learned the underlying patterns instead of simply memorizing the training data. This is supported by Dr Doshi.R, Dr Hiran.K, Ritesh Kumar.J and Dr Lakhwani.K (p49, 2022) as they say ,"This is important because using the same data sets for both training and evaluation would not give a fair assessment of the model's performance in real-world scenarios".

4.1 Hyper-Parameters of the CNN-Based Model

During training the model, we used the categorical cross-entropy as the loss function. We ran the our model for 50 epochs as there are no further improvements in training and validation ac-curacies observed by the graphs below on Fig4.2 1 and Fig4.2.2. For the loss function we applied Optimizer Adam. As shown by the table 2 in the methodology section, we 32, 64, 64, 64, 64 and 64 filters for Conv1, Conv2 ,Conv3, Conv4, Conv5 and Conv6 respectively with epochs of 50 and a batch size of 32.

4.2 Results on the Epochs.

The graphs below shows the training and validation loss as well as the training and validation accuracy. In Fig4.2.2, it is shown that the validation loss is almost the same as the training loss in 15th to 18th epochs. But in In Fig. 4.2.1 at that time, the validation accuracy is greater than the training accuracy. Around 47th to 48th epochs, validation loss is closer to the training loss and they are fluctuating and the validation accuracy is at its highest level. The training accuracy is still increasing and with those observations, we stop training our model at the 50th epoch.

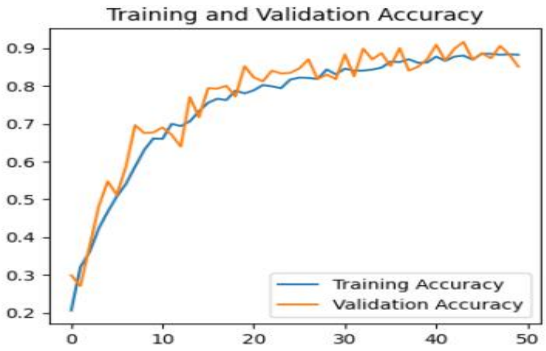




**Fig 4.2.1** Training accuracy vs validation accuracy in recognizing recognizing tomato leaf diseases

**Fig 4.2.2** Training loss vs validation loss in tomato leaf diseases

The model achieves the best training, validation and test accuracies for 32, 64, 64, 64, 64 and 64 filters for Conv1, Conv2 ,Conv3, Conv4, Conv5 and Conv6 respectively, with 3 × 3 filters.

4.2 Performance Analysis.

Dataset of 6952 tomato leaf disease images of seven classes is used to evaluate our CNN-based model. We also implemeted the confidence of the model on trying to indicates how well the model is performing toward achieving its goal (Brain J.A 2023) Model accuracy, on the other hand, refers to the model's skill in the percentage of predictions it gets right for a certain use-case. We predicted the images using the designed model and the class predicted class was mentioned as shown below

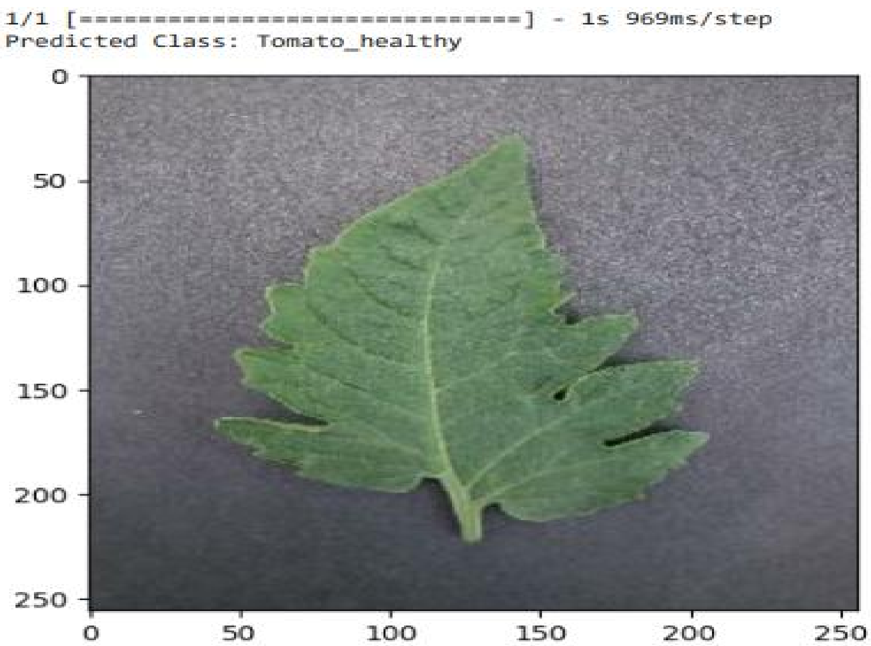
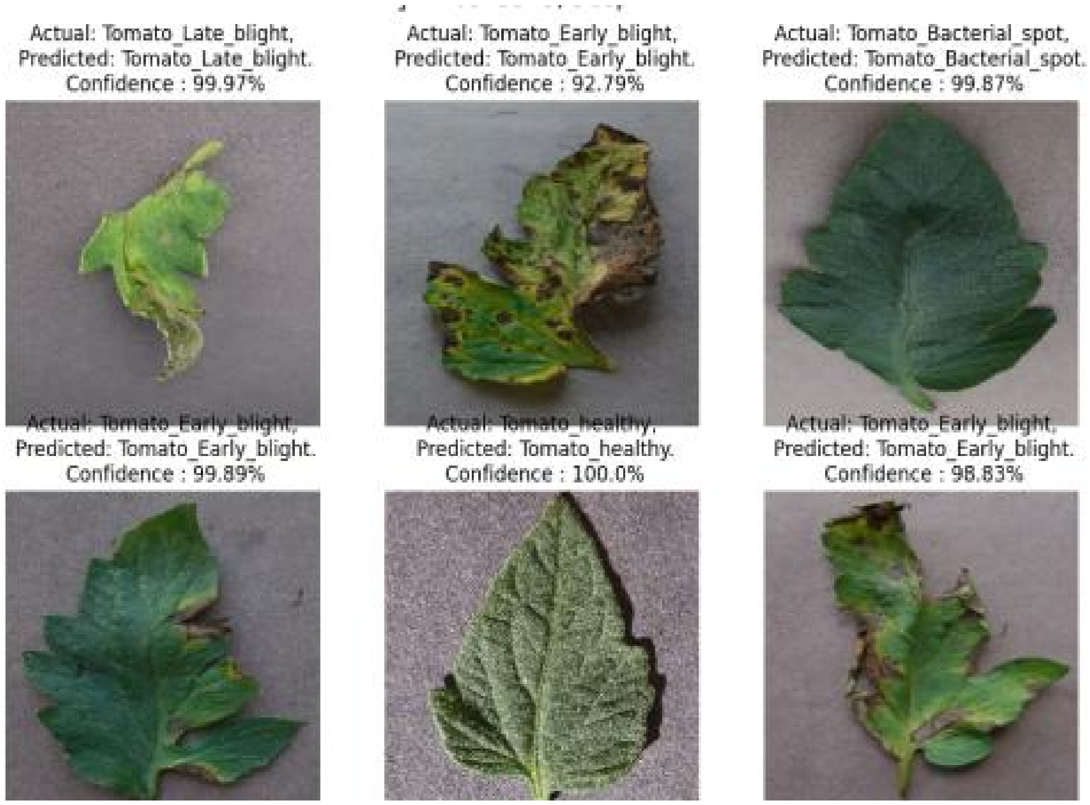



Fig4.3 : Below are the predicted images and their predicted class and original class name as well as the level of confidence




As shown above, only one image has the lowest confidence of 92.79% but all of them has the maximum expected percentages with the accurate predicted class name which best reflects the accuracy of the model.

**<u>Tomato Leaf Diseases Recognition Using Convolutional Neural Networks</u>**

<u>Table 4. Quantity of network images in different models</u>

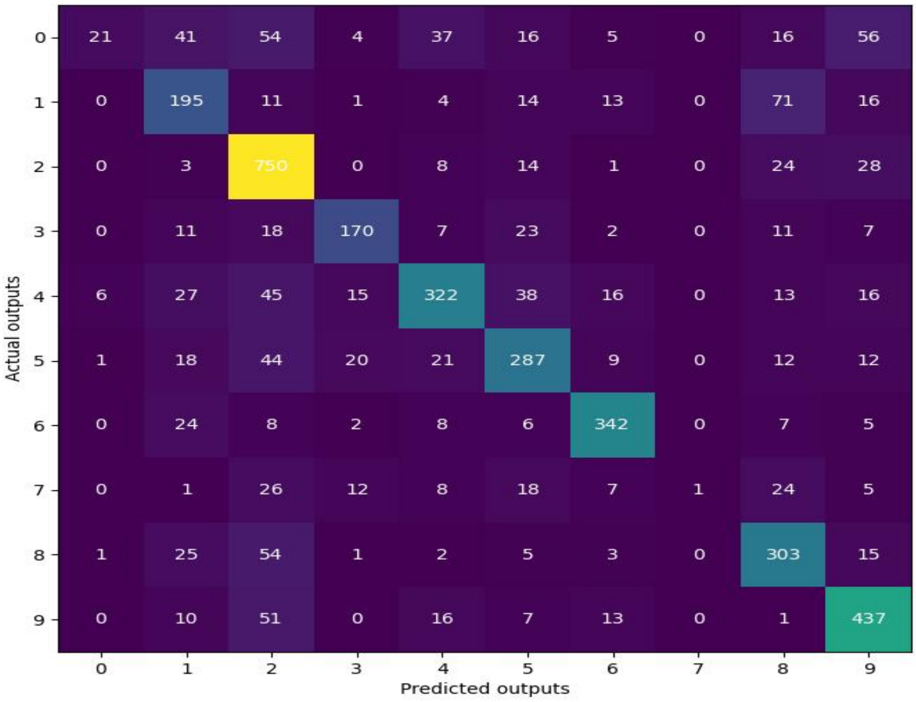
| ***Model*** | ***# Images*** | ***% Accuracy*** |
|---|---|---|
| Our Model | 6952 | 90.01 % |
| Model developed by Punam.B, Pushkar.G (2021) | 9,914 | 99.35 % |
| Model by Srdjan. S, Arsenovic.M (2016) | 7900 | 96.3% |

The above is just a summary of some of the researchers that have worked on detecting tomato leaf diseases using CNN. The differences on the accuracy of the model best shows that the higher the number of training images the higher the accuracy percentage.

<u>4.2 Random Forest</u>

The research was done with tomato plant leaves affected by different diseases as well as some normal leaves.For random forest we achieved the accuracy of 70.3% with number of decision trees (n_estimators) being set to 100.

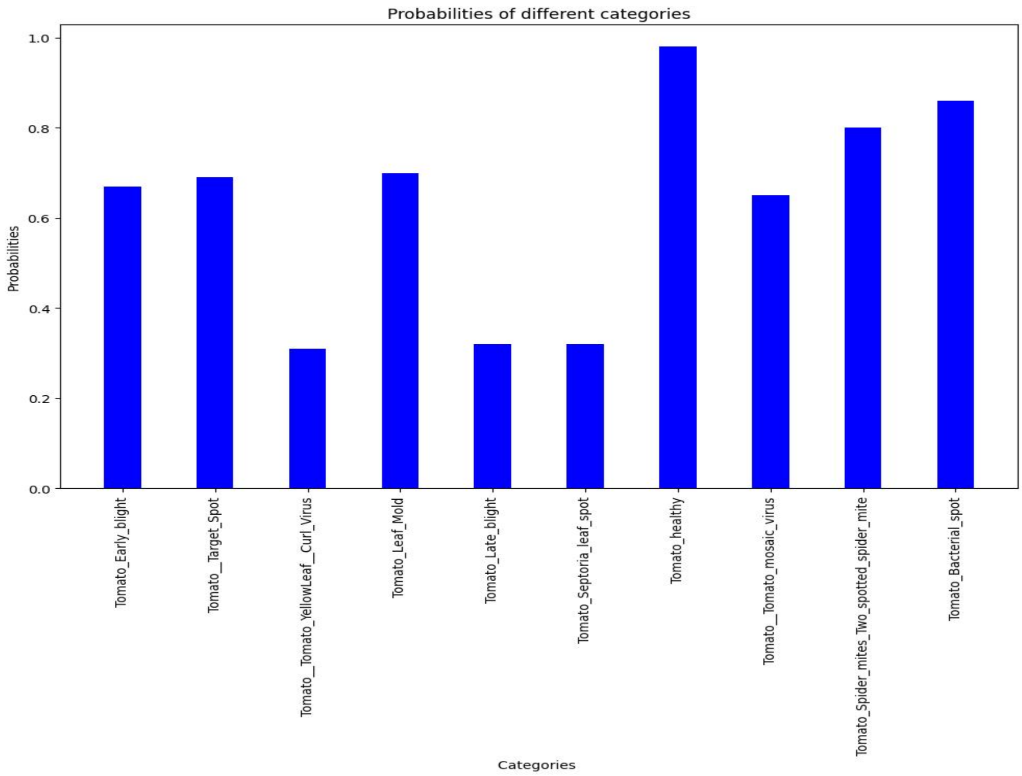
The results can be visualized using the heat map below which shows diagonal entries being the number of correct predictions vs other incorrect predictions :



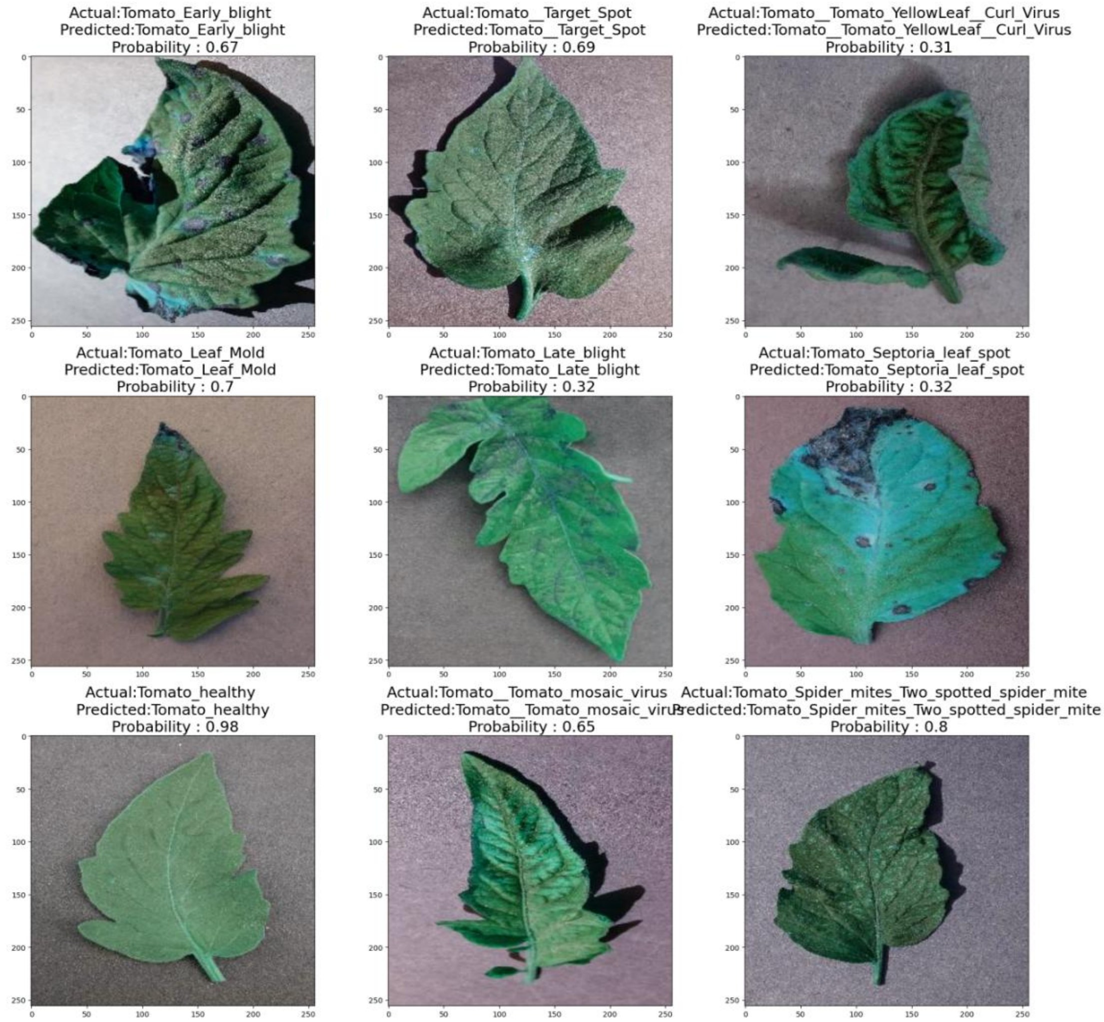

Several other performance measurement metrics were also obtained from the results. Precision measures the accuracy of positive predictions and is calculated by dividing the true positives with all positive outputs. Recall is a measure of how the model correctly identifies true positive outputs and can be calculated by dividing the true positives by all outputs that should have been predicted as positives (i.e True positives + False negatives)The F1-score shows a weighted average of precision and recall. The precision , recall and f1-score of the random forest classifier are summarized in the table below:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.72 | 0.08 | 0.15 | 250 |
| 1 | 0.55 | 0.60 | 0.57 | 325 |
| 2 | 0.71 | 0.91 | 0.79 | 828 |
| 3 | 0.76 | 0.68 | 0.72 | 249 |
| 4 | 0.74 | 0.65 | 0.69 | 498 |
| 5 | 0.67 | 0.68 | 0.67 | 424 |
| 6 | 0.83 | 0.85 | 0.84 | 402 |
| 7 | 1.00 | 0.01 | 0.02 | 102 |
| 8 | 0.63 | 0.74 | 0.68 | 409 |
| 9 | 0.73 | 0.82 | 0.77 | 535 |
| accuracy | | | 0.70 | 4022 |
| macro avg | 0.73 | 0.60 | 0.59 | 4022 |
| weighted avg | 0.71 | 0.70 | 0.68 | 4022 |

The dataset used had  10 different categories of leaf conditions . The random forest classifier splits the dataset into random decision trees and each decision tree will make independent predictions on the data . Each tree predicts class probabilities and these probabilities are averaged for the predictions.The bar chart below shows the the probability for different categories of the dataset.




A single image from each category were randomly chosen in order to test the model’s ability to predict their category. All the images were correctly classified despite the different range of probabilities. The images below shows the images that were tested along the respective probabilities :




# 5.Conclusion and future work

In this research topic,we have developed a customized CNN-based model for agriculture which can be used to categorize seven mainly tomato leaf diseases that can be found in Zimbabwe. The model is trained to visualize the features on the leaves then recognize the tomato leaf diseases in different image backgrounds and capture conditions. Our has reached an average of 90.9% accuracy on independent test images.This model has a great advantage as it has reduced storage with few number of training parameters. We will try to modify the model so that it can can be used for

all other plant leaves to detect leaves not only basing on the tomato leafs. Again, the classification accuracy is not 100 % accurate which means there is need to adjust on the parameters (Doshi-Velez, F., Kim, B.) of our CNN-based model to present features clearly for which diseases will be categorized.
The random forest classifier showed an accuracy of 71% . This was significantly less accurate compared to the CNN model. The model was however able to correctly classifier all the images that were tested on it indicating its suitability for image classification. In future , the random forest classifier parameters need to be fine-tuned in order to achieve the maximum accuracy possible.

References

1. Mohanty, S. P., Hughes, D. P., & Salathe, M. (2016). Using Deep Learning for Image-Based Plant Disease Detection. Frontiers in plant science, 7, 1419
2. Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. Computational intelligence and neuroscience, 2016.
3. Singh, A., Ganapathysubramanian, B., & Singh, A. K. (2016). Machine learning for high-throughput stress phenotyping in plants. Trends in plant science, 21(2), 110-124
4. Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. Computers and electronics in agriculture, 145, 311-318.
5. Mohanty, S. P.,. (2017). Automated crop disease diagnosis using mobile capture devices and machine learning. Frontiers in plant science, 8, 1808.

6. Punam.B, Pushkar.G(2021) Plant disease detection using hybrid model based on convolutional autoencoder and convolutional neural network
7. Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. Computers and electronics in agriculture, 145, 311-318.
8. Chen , P. ,Yuan,P.,Liu , Z., Wang , T, Xue , J. ,and Sun , Z.(2018). Plant disease recognition from images using deep Convolutional neural networks . Sustainable Computing : Informatics and Systems,18,13-18.
9. Mishra,R.,Tripathy,B.K. and Sharma , R. (2021). AI based crop disease detection using hybrid feature extraction method. Computers and Electronics in Agriculture,185,106016
10. Gonzalez-Dugo, V, (2019). High-throughput field phenotyping using hyperspectral reflectance and partial least squares regression (PLSR) machine learning. Remote Sensing, 11(3), 261.
11. Singh, A., & Ganapathysubramanian, B. (2017). Machine learning for plant stress phenotyping: Trends and future perspectives. Frontiers in plant science, 8, 1182.
12. Li ,L. , Wang , Y. , Wu , X. , Zhang , J. ,Shan , J. and Zhao , C.(2019). Detection of powdery mildew on cucumber leaves based on a VNIR hyperspectral sensor and machine learning algorithms. Remote Sensing , 11(6), 692
13. Andreasson , P.R. , MNSSON , P.E. , and Lilinthal, A. (2017). Early disease detection in barley with deep convolutional neural networks and hyperspectral imaging. Sensors , 17(7),1520.
14. Bhattacharya, U., et al. (2020). Deep learning approach for plant disease detection and diagnosis through leaf image analysis: A review. Archives of computational methods in engineering, 27(5), 1655-1681.
15. Barbedo, J. G. A. (2019). Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. Computers and electronics in agriculture, 162, 184-189.
16. Zaman ,B . Guvenc , I. , Adak ,M.K. (2020). Potato leaf disease classification using LBP features and random forest. Journal of Ambient Intelligence and Humanized Computing, 11(3),1163-1173.
17. Yu Han Liu.J (2018) Feature Extraction and Image Recognition with Convolutional Neural Networksp4 .

A. Camargo, J. Smith Computers and Electronics inAgriculture, 66 (2) (2009), p121-125

18. Li, C., et al. (2020). A deep learning model for cucumber disease recognition under complicated cultivation environment. Precision agriculture, 21(2), 258-277.
19. Pande, S., et al. (2018). Artificial intelligence (AI) in agriculture: A comprehensive review. Sensors, 18(8), 2674.
20. X.Wu, V. Kumar, R.J. Quinlan, Ghosh, Y.Q. Joydeep, H. Motoda, J. McLachlan, N.A. Geofirey, B. Liu, S.P. Yu, Z.H. Zhou, M. Steinbach, J.D. Hand, D. Steinberg( Knowledge and Information Systems, 14 (1) (2008), pp. 1-37)
21. Varsha P. Gaikwad, Dr. Vijaya Musande, "Wheat Disease Detection Using Image Processing", Aurangabad, India, pp-110-112, 2017
22. Rong Zhou, Shun"ichi Kaneko, Fumio Tanaka, Miyuki Kayamori, Motoshige Shimizu, "Early Detection and Continuous Quantization of Plant Disease Using Template Matching and Support Vector Machine Algorithms", pp. 300-304, Sapporo, Japan 2013
23. Sladojevic, S., et al. (2019). Plant disease identification using explainable 3D deep learning on hyperspectral images. Plant methods, 15(1), 144.
24. Liakos, K. G, . (2018). Machine learning in agriculture: A review. Sensors, 18(8), 2674.
25. Gonzalez-Dugo, V, (2019). High-throughput field phenotyping using hyperspectral reflectance and partial least squares regression (PLSR) machine learning. Remote Sensing, 11(3), 261.
26. Singh, A., & Ganapathysubramanian, B. (2017). Machine learning for plant stress phenotyping: Trends and future perspectives. Frontiers in plant science, 8, 1182.
27. Bhattacharya, U., et al. (2020). Deep learning approach for plant disease detection and diagnosis through leaf image analysis: A review. Archives of computational methods in engineering, 27(5), 1655-1681.
28. Louis.O (2022)Hyperparameter Tuning with Python(p228)
29. Sarkakar.G, Paril.G and Dutta.P(p53) Internet of Things and Machine Learning
30. Dr Doshi.R,Dr Hiran.K,Ritesh Kumar.J and Dr Lakhwani.K (p49, 2022) Master Supervised and Unsupervised Learning Algorithms with Real Examples
31. Islam.M.Z (2019) Tomato Plant Diseases Detection Using Convolutional Neural Networks

32. S. Ghosal et al. (2018) Tomato plant disease identification using deep convolutional neural network
33. Wu X and Kumar V, 2009 *The Top Ten Algorithms in Data Mining London*
34. Han J and Kamber M, 2006 *Mining Stream, Time-Series and Sequence Data* 54
35. Leidiyana H, 2013 *Penerapan Algoritma K-Nearest Neighbor Untuk Penentuan Resiko Kredit Kepemilikan Kendaraan* Bermotor J. Penelit. Ilmu Komputer, Syst. Embed. Log. 1, 1 p. 65–76.
36. Mustakim G O, 2016 *Algoritma K-Nearest Neighbor Classification* J. Sains, Teknol. dan Ind. 13, 2 p. 195–202.
37. Yunita D, 2017 *Perbandingan Algoritma K-Nearest Neighbor dan Decision Tree untuk Penentuan Risiko Kredit Kepemilikan* Mobil J. Inform. Univ. Pamulang 2, 2 p. 103.
38. Gorunescu F, 2011 *Data Mining "Concepts, Models and Techniques*" 12 Berlin, Heidelberg: Springer Berlin Heidelberg.
39. Jani K and Noor A H, 2018 *Klasifikasi Penyakit Daun Padi Berdasarkan Hasil* Ekstraksi Fitur GLCM Interval 4 Sudut J. Teknol. Inf. dan Komun. STMIK ProVisi Semarang 03, 01 p. 1–6
*40. T. Gupta, "Plant leaf disease analysis using image processing technique with modified SVM-CS classier," International Journal of Engineering Managanagement Technology, vol. 5, pp. 11–17, 2017.*

41. E. Mwebaze and G. Owomugisha, "Machine learning for plant disease incidence and severity measurements from leaf images," in *Proceedings of the 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, IEEE, Anaheim, CA, USA, December 2016.

42. H. Wang, G. Li, Z. Ma, and X. Li, "Application of neural networks to image recognition of plant diseases," in *Proceedings of the 2012 International Conference on Systems and Informatics (ICSAI2012)*, Yantai, China, May 2012.

    A. Camargo and J. S. Smith, "An image-processing based algorithm to automatically identify plant disease visual symptoms," *Biosystems Engineering*, vol. 102, no. 1, pp. 9–21, 2009.

43. Meunkaewjinda, P. Kumsawat, K. Attakitmongcol, and A. Srikaew, "Grape leaf disease detection from color imagery using hybrid intelligent system," in *Proceedings of the 2008 5th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, Krabi, Thailand, May 2008.

44. M. Pagola, R. Ortiz, I. Irigoyen et al., "New method to assess barley nitrogen nutrition status based on image colour analysis: comparison with SPAD-502," *Computers and Electronics in Agriculture*, vol. 65, no. 2, pp. 213–218, 2009.

45. Chen , P. ,Yuan,P.,Liu , Z., Wang , T, Xue , J. ,and Sun , Z.(2018). Plant disease recognition from images using deep Convolutional neural networks . Sustainable Computing : Informatics and Systems,18,13-18.
46. Mishra,R.,Tripathy,B.K. and Sharma , R. (2021). AI based crop disease detection using hybrid feature extraction method. Computers and Electronics in Agriculture,185,106016
47. Li ,L. , Wang , Y. , Wu , X. , Zhang , J. ,Shan , J. and Zhao , C.(2019). Detection of powdery mildew on cucumber leaves based on a VNIR hyperspectral sensor and machine learning algorithms. Remote Sensing , 11(6), 692.
48. Andreasson , P.R. , MNSSON , P.E. , and Lilinthal, A. (2017). Early disease detection in barley with deep convolutional neural networks and hyperspectral imaging. Sensors , 17(7),1520.
49. Zaman ,B . Guvenc , I. , Adak ,M.K. (2020). Potato leaf disease classification using LBP features and random forest. Journal of Ambient Intelligence and Humanized Computing, 11(3),1163-1173.
50. Hatuwal , K.B. , Shakya, A. ,Joshi B (2021). Plant leaf disease recognition using Random Forest,KNN,SVM and CNN . https://www.researchgate.net/publication/351708837
51. Guptal ,S. N. ,Venkata , P. , Triveni ,M.A. (2022) . Detection of plant diseases using random forest classifier. International Journal of innovative research in technology..https://ijirt.org/master/publishedpap
52. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. J. Big Data 6(1), 60 (2019)
53. Brahmbhatt, S.: Practical OpenCV. Apress, New York (2013)
54. Srdjan Sladojevic, Marko Arsenovic, Andras Anderla , Dubravko Culibrk, and Darko Stefanovic, : Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification 2016
55. Brain J.A Testing your NLP Models, 2023
56. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. DIO:1702.08608 (2017)
57. Hanson, A.M.J., Joy, A., Francis, J.: Plant leaf disease detection using deep learning and convolutional neural network, vol. 7 (2017)

58. LeCun, Y., et al.: Backpropagation applied to handwritten zip code recognition. Neural Comput. 1(4), 541–551 (1989)