

# Classification of Plant Diseases

Ghufran Makki Mohammed

Artificial Intelligence, Midocean University

Dr. Hager Saleh

## Abstract

The paper discusses the effectiveness of different machine learning (ML) models in classifying plant diseases. For this, we consider cherry and bell pepper plants. We use the PlantVillage dataset containing images of cherry plants infected with powdery mildew and healthy leaves and bell pepper plants infected with bacterial spots and healthy leaves. We preprocessed the dataset by resizing the images to 64x64 pixels and normalizing pixel values. Labels were one-hot encoded.

To improve this dataset, we have applied several data augmentation techniques: rotation, width and height shifts, and horizontal flipping. We implemented a Convolutional Neural Network(CNN) that consisted of two convolution layers, two max-pooling layers, one flattening layer, one dense layer with dropout, and one final output layer with SoftMax activation. The model was fitted to the data using the Adam optimizer and categorical cross-entropy loss for 20 epochs. Based on these considerations, the Convolutional Neural Network (CNN) thus realized a high level of accuracy of 95% when evaluated on a validation set.

We implemented the Convolutional Neural Network (CNN) and several other traditional machine learning models using features from the flattened images, including K-Nearest Neighbors (KNN), Decision Trees, and Random Forests. These experiments showed that the Convolutional Neural Network (CNN) performed better than other traditional models on the given classification problem. However, they gave insight into feature importance and decision processes.

The efficiency is quite good in Convolutional Neural Networks (CNN) to classify image-based plant diseases and can benefit from both data augmentation and deep feature extraction. The presented work is a step toward using deep learning models in agriculture and for plant disease diagnosis that may eventually provide rapid and efficient automation in such critical diagnostics, hence the early control and management of crops. This comparative performance analysis also highlights the choice of model for application requirements.

Keywords cherry and bell pepper plants diseases, Convolutional Neural Networks, PlantVillage dataset, Deep Learning including K-Nearest Neighbors (KNN), Decision Trees, and Random Forests, Image Classification.

## Table of Contents

|                                            |    |
|--------------------------------------------|----|
| Classification of Plant Diseases .....     | 1  |
| Abstract .....                             | 2  |
| 1. Introduction .....                      | 4  |
| 2. Related Work.....                       | 5  |
| 3. Methodology .....                       | 7  |
| 4. Result Analysis and Visualization ..... | 15 |
| 5. Conclusion .....                        | 15 |
| 6. References.....                         | 16 |

## 1. Introduction

By convention, plant diseases are classified by physical/manual inspection and visual assessment by agricultural experts. Farmers and agronomists visually inspect leaves, stems, and fruits for any visible symptoms such as color change, lesions, and other abnormal morphological changes. While effective in some ways, this method has also suffered from major limitations. It is highly dependent on expertise so consequently to human mistakes, to distinguish between diseases of similar visual symptoms. Besides, it is a labor-intensive process that can be slow, especially for large areas that need to be inspected. These challenges have risen and increased the demand for effective and scalable plant disease detection and classification.

The influence of increasing research in image processing, machine learning, and artificial intelligence affecting outlooks on the classifications of plant diseases has thus set in. These approaches ensure that the detection is efficient and done promptly, independent of human skills. Applications have involved the use of deep learning techniques, such as CNNs, that look toward the examination of plant images to enable disease classification. These features are automatically detected by CNNs from images to learn complex patterns that increase the accuracy of this technique over traditional methods.

This study, therefore, will be considering deep learning in the classification of plant diseases with the use of a CNN model and its performance evaluation with other traditional machine learning models such as K-Nearest Neighbors, Decision Trees, and Random Forests. The CNN model, which has proven quite efficient in handling image data, is trained on the dataset of images of various diseases of plants, among them Powdery Mildew on Cherry (including sour) and Bacterial Spot on Bell Peppers. The architecture will consist of several convolutional layers followed by max-pooling and dense layers with dropout regularization to handle overfitting.

Several studies demonstrated the potential of deep learning for plant disease classification. For example, Grünig et al. (2021) used deep neural networks to predict the incidence and phenology of plant pests and diseases, thus illustrating the potential of AI in agriculture. Sharma et al. (2022) also explored the diagnosis of plant diseases with deep learning and noted that CNNs effectively provide high accuracy in plant disease classification tasks. Akshai and Anitha (2021) demonstrated the deep learning model, which had successfully classified many plant diseases, and proved that these days, the concept has shifted from traditional methods toward automated systems.

These deep learning models, especially CNNs, used in this work provide a more robust and scalable solution for the classification of plant diseases. Unlike traditional methods, CNNs can learn features hierarchically from images with much precision and efficiency in identifying diseases. This approach significantly reduces the reliance on human expertise

and allows scaling the disease detection process to larger agricultural fields, thereby improving early detection and overall crop management.

In this work, the performance of CNNs is compared to other classical machine learning models. It was also assessed against the variability in images due to different environmental conditions and plant species. Its findings add to a growing body of work illustrating the potential of AI for modern agriculture and plant health management.

## 2. Related Work

In [1] Grünig et al. applied Deep neural networks (DNN) to predict plant diseases. They created a dataset consisting of environmental data and historical plant disease data. The model is trained on multiple data sources, which gives strength to the model. The model is characterized by high accuracy in the prediction and integration of different data sources. The limitation of this model is historical data requirements. Finally, the model achieved an accuracy of 85%, which helped manage plant diseases.

In [2] Sharma et al. applied Convolutional Neural Networks (CNNs) to classify plant diseases. They used the PlantVillage dataset containing more than 50,000 diverse images of diseased and healthy plant leaves. That gives a large training set by the dataset. This model works with a layered structure to extract the classification. This model gives very accurate results and provides automated diagnosis which makes it practical to implement in the real world. The Limitations of this model are low-resolution images, and it requires a large dataset. This model achieved 98% accuracy which shows great results.

In [3], Akshai and Anitha used a deep learning (DL) model using (CNN) to classify plant diseases. They collected them from local agricultural sources. The dataset contained thousands of images. The model is adaptive to a variety of plant species, making it suitable for various crops. It is also compatible with a mobile application, which has enhanced its use in science. The model has an accurate performance. One of the limitations of this application is that it relies on high-quality images. However, the model achieved an accuracy of 94%, which gives it reliability in the real world.

In [4] Chen, Zhang, and Nanekaran applied Deep Learning with a lightweight neural network to identify plant diseases. They used the PlantVillage dataset, which contains more than 50,000 images and many types, to train the model. The computational efficiency of this model was improved by reducing the computational requirements, speeding up the inference time, and giving more accurate results. This model is considered good, but its limitations include not working with new types of diseases. However, it showed 96% accuracy results, which were effectively used to reduce computational resources.

In [5] Suresh et al. developed the DeepPlantNet model, which uses deep learning to classify diseases from plant leaf images. The plant dataset used to train the model was collected from different sources, which gave the possibility of scaling to different plants. One of the

main limitations of this model is that training the data requires large computational resources and needs to be updated with new data to maintain a high level of accuracy. This model achieved an accuracy of up to 97%, which confirmed its effectiveness in classifying diseases.

In [6], Subasi, Gulam, and Sadikoglu use deep learning to build a model to detect maize leaf diseases. They collected a dataset from local sources consisting of maize leaf images and trained the model on these images. The advantage of this model is that it is a mobile-based system and can be used in real-time. This makes it practical and easy to use for farmers. The model was limited to maize leaves and relied on the accuracy of the image quality. However, it achieved a performance of 92%.

In [7] Kaur et al. developed a mobile application to detect maize leaf diseases using deep learning algorithms. They used a dataset from agricultural research centres to train the model. The model can be used in real-time to give immediate results, so the model is practical for farmers. The model has accurate results of 93%, but the performance is affected by environmental factors such as lighting conditions.

| Author                            | Methodology                                     | Dataset                                                             | Advantages                                                                      | Limitations                                                                   | Results                                                      |
|-----------------------------------|-------------------------------------------------|---------------------------------------------------------------------|---------------------------------------------------------------------------------|-------------------------------------------------------------------------------|--------------------------------------------------------------|
| <i>Grünig et al. (2021)</i>       | DNN to predict disease incidence                | consisting of environmental data and historical plant disease data. | High-accuracy data and analyzes of complex datasets                             | Data quality affects accuracy and historical data requirements                | Successful predictions for disease with an accuracy of 85%   |
| <i>Sharma et al. (2022)</i>       | CNNs for image classification of plant diseases | PlantVillage dataset.                                               | High accuracy, automates detection                                              | Requires large datasets, low-resolution images                                | High classification accuracy 98% achieved                    |
| <i>Akshai &amp; Anitha (2021)</i> | CNN models trained on leaf images               | collected dataset from local agricultural sources                   | efficient classification                                                        | Sensitivity to overfitting and depend on high-quality images                  | classification accuracy of 94%                               |
| <i>Chen et al. (2023)</i>         | Deep transfer learning with lightweight network | PlantVillage dataset and additional images from online sources.     | Faster training, reduced resource needs                                         | Performance dependent on base model so not working with new types of diseases | Effective disease identification showed 96% accuracy         |
| <i>Suresh et al. (2023)</i>       | DeepPlantNet for disease classification         | collected from different sources                                    | High accuracy and efficiency and the possibility of scaling to different plants | Requires substantial resources and large computational resources              | classification performance achieved an accuracy of up to 97% |

|                             |                                           |                                                  |                                   |                                                                    |                                                       |
|-----------------------------|-------------------------------------------|--------------------------------------------------|-----------------------------------|--------------------------------------------------------------------|-------------------------------------------------------|
| <i>Subasi et al. (2022)</i> | Mobile-based system using DL              | Collected dataset for maize plant leaf diseases. | Practical application for farmers | Internet access required, light condition issues and image quality | Successful real-time disease detection 92% accuracy   |
| <i>Kaur et al. (2023)</i>   | Mobile-based system for disease detection | Custom dataset for maize plant leaf diseases.    | Quick diagnosis, user-friendly    | Accuracy may vary in diverse conditions                            | Effective classification with accurate results of 93% |

### 3. Methodology

#### 1.3. Data Collection

In this study Cherry with Powdery Mildew and healthful and bell Pepper with Bacterial Spot and healthful datasets are used and taken from the PlantVillage Kaggle images dataset. We have focused on two categories of Cherry diseases: "Cherry\_(including\_sour)\_\_\_Powdery\_mildew" and "Cherry(including\_sour)\_\_\_healthy". And two categories of pepper "Pepper,\_bell\_\_\_Bacterial\_spot" and "Pepper,\_bell\_\_\_healthy."

#### 2.3. Data Preprocessing

The dataset is divided into 80:20 subsets and organized for training and validation purposes. where 80% of the images are used for training and 20% of the images are used for testing. The leaves dataset containing a total of 4381 images is used in the study. Some of the leaves are shown in Fig 1. Tab. 1 represents the number of training and testing images of the datasets.

| Image Dataset and class                         | Total Image | Training split | test split |
|-------------------------------------------------|-------------|----------------|------------|
| <b>Cherry_(including_sour)___Powdery_mildew</b> | 1052        | 841            | 210        |
| <b>Cherry(including_sour)___healthy</b>         | 854         | 683            | 170        |
| <b>Pepper,_bell___Bacterial_spot</b>            | 997         | 797            | 199        |
| <b>Pepper,_bell___healthy</b>                   | 1478        | 1182           | 295        |

Table 1: Cherry and Pepper leaf datasets

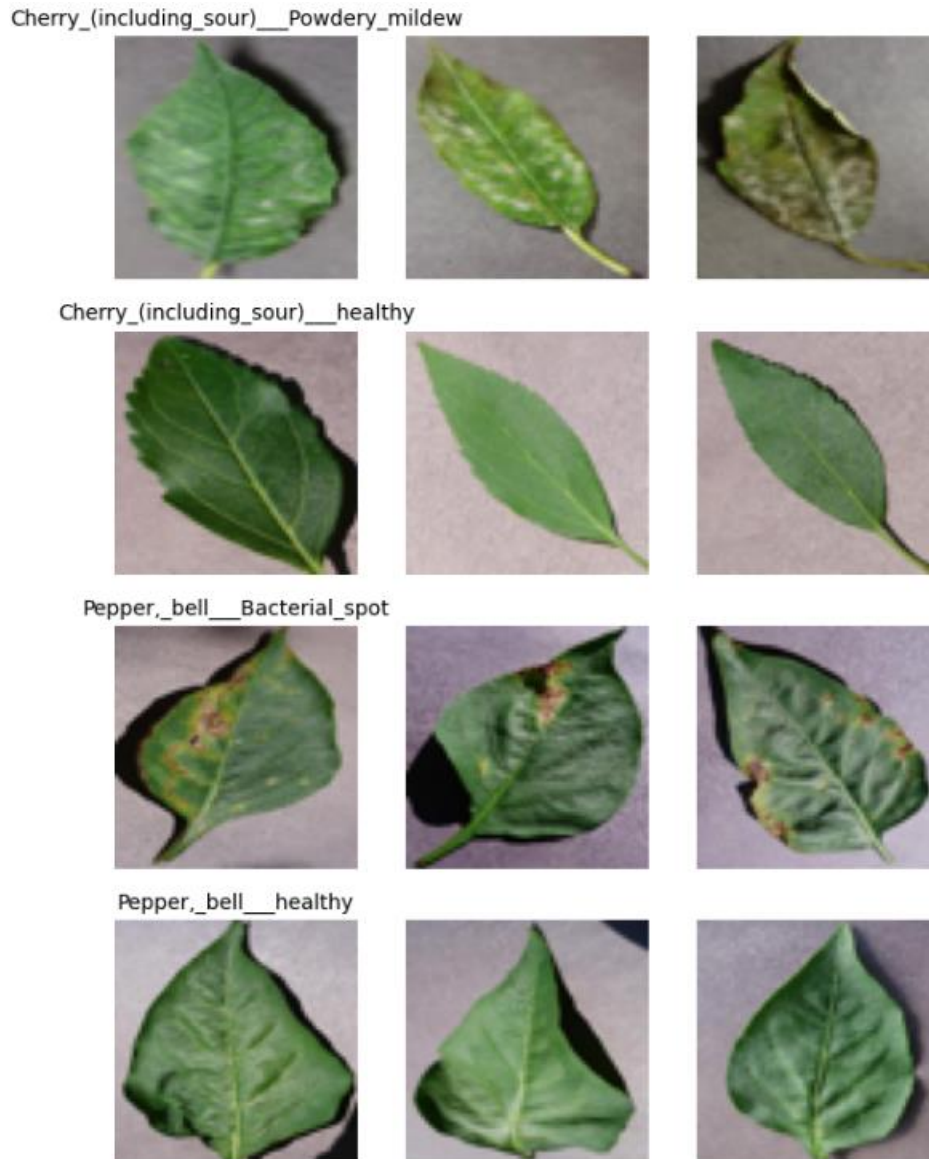


Figure 1: Sample image of Cherry with Powdery Mildew and healthful and bell Pepper with Bacterial Spot and healthful bell peppers.

During the preprocessing stage, data augmentation rescales the images to 64x64 pixels as a fixed size as input into the model. Then each image is normalized by dividing the pixel values by 255 so that the pixel values lie in a range of 0 to 1.

Data augmentation- including random rotation up to 20°, up to 20% of the width/height for horizontal/vertical shifts, and random horizontal flipping utilized to increase the robustness of the model and to avoid overfitting. These are ways to artificially increase the diversity in the training set so that the model generalizes better. Images of the dataset after preprocessing are shown in Fig 2

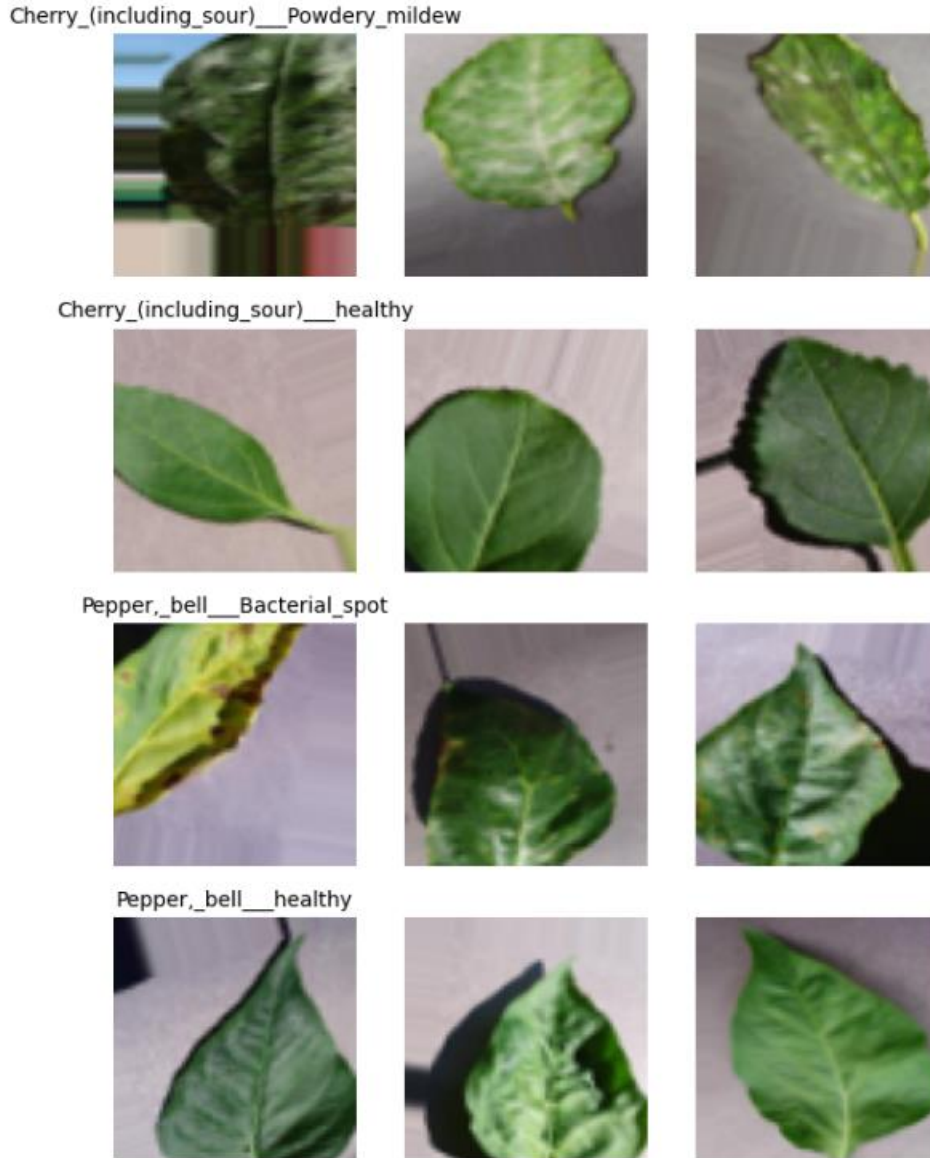


Figure 2: Resized Cherry and Pepper leaf dataset

### 3.3. Model architecture

In this study, we build three different models to classify plant diseases

Several deep learning models are used in this research to classify disease in plants amongst them, the CNN plays the prime role. The architecture involves many layers that help to generate features from the inputted image to categorize diseases. It first considers an input layer that supports the dimensions of 64 x 64 pixels with 3 color channels. It is followed by two convolutional layers, each with 32 and 64 filters of size (3,3), respectively, that incorporate the ReLU for the extraction of low-level features such as edges and textures. The convolutional layers are further supported by max-pooling layers, which reduce the image in spatial dimensions to enhance computational efficiency, hence

retaining the salient features. These are followed by flattening the data into a 1D vector to pass through a dense layer of 128 units, ReLU-activated, for high-level feature learning. For this purpose, a dropout rate of 0.5 is added to prevent network overfitting during the network training process. Finally, there is the output layer with a softmax activation function with 4 units representing the four classes of diseases. Besides CNN, the other classical machine learning models used for comparison include K-Nearest Neighbors (KNN), Decision Tree, and Random Forest. All these models operate on feature vectors derived from the flattened images, but CNN outperforms them in accuracy since it can automatically learn hierarchical features directly from the raw image data. See in Fig 3 and Fig 4.

| Layer (type)                    | Output Shape       | Param #   |
|---------------------------------|--------------------|-----------|
| conv2d_12 (Conv2D)              | (None, 62, 62, 32) | 896       |
| max_pooling2d_12 (MaxPooling2D) | (None, 31, 31, 32) | 0         |
| conv2d_13 (Conv2D)              | (None, 29, 29, 64) | 18,496    |
| max_pooling2d_13 (MaxPooling2D) | (None, 14, 14, 64) | 0         |
| flatten_6 (Flatten)             | (None, 12544)      | 0         |
| dense_12 (Dense)                | (None, 128)        | 1,605,760 |
| dropout_6 (Dropout)             | (None, 128)        | 0         |
| dense_13 (Dense)                | (None, 4)          | 516       |

Total params: 1,625,668 (6.20 MB)

Trainable params: 1,625,668 (6.20 MB)

Non-trainable params: 0 (0.00 B)

Figure 3: Model layer

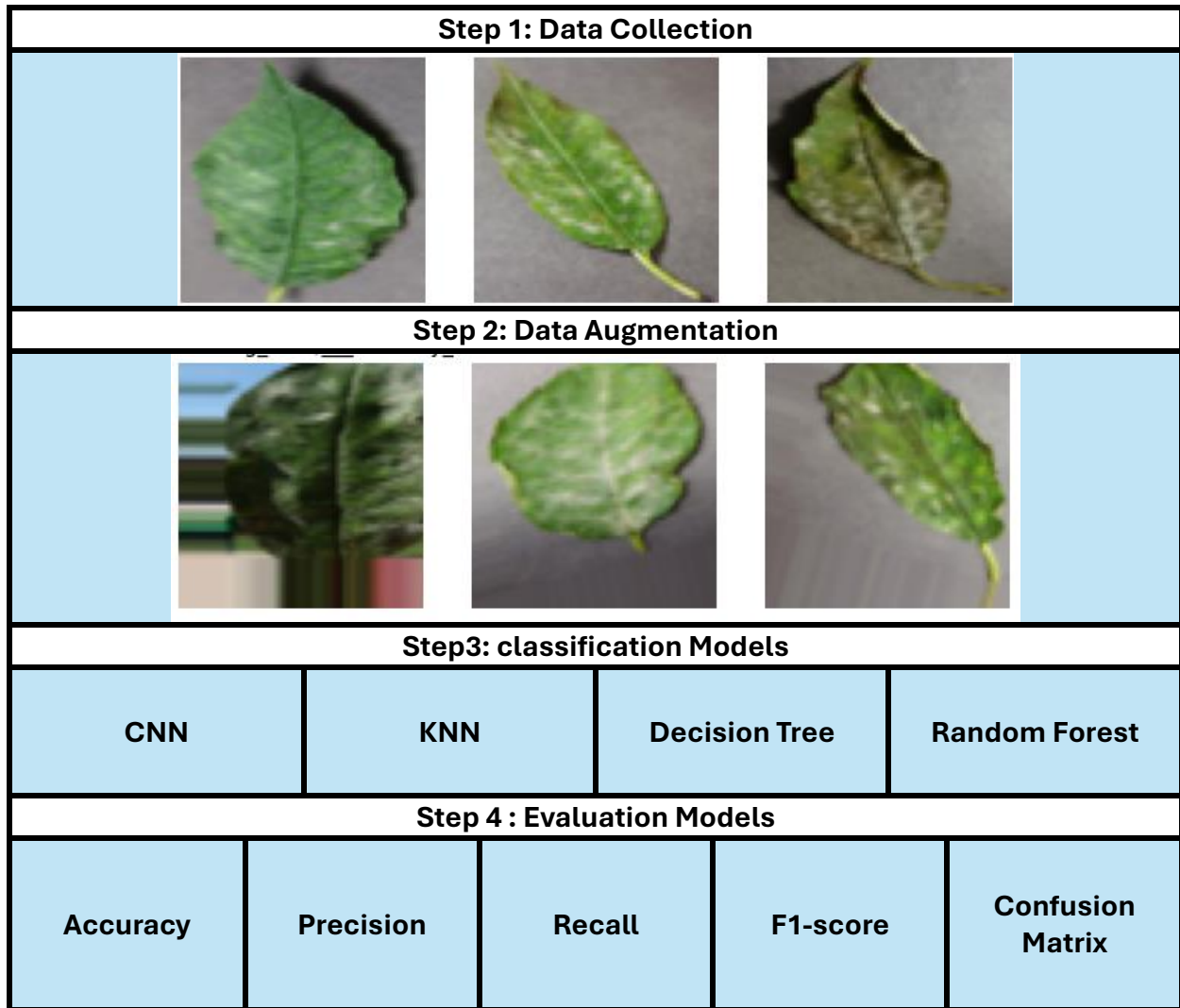


Figure 4: Model architecture

### 3.4. Model Training

Training the CNN model, where the 'Adam' optimizer is used, embedding the loss function of a categorical cross-entropy regarding multi-class classifications. Training will last for 20 epochs where the model itself learns from the input given in the picture the relations in the classes of plant diseases. While in training, checking will be exercised on the performance of this model on the validation set to know the trends in accuracy or loss on each epoch run. Although very simple, the CNN model reaches an accuracy as high as 94.06%, showing efficiency in the task of plant disease classification from the image of a leaf. Other models, including KNN, Decision Tree, and Random Forest, were also trained using flattened image features, and their performances were compared against the CNN model. These results showed variable precisions, recall, and F1-scores for each model.

| Parameter           | Value                    |
|---------------------|--------------------------|
| Optimizer           | Adam                     |
| Loss                | Categorical Crossentropy |
| Batch Size          | 32                       |
| Metrics             | Accuracy                 |
| Activation Function | ReLU<br>Softmax          |

Table 2: CNN model training parameters

### 3.5 Training (Accuracy & Loss) Graphs of the CNN Model

The model for the designed CNN models to classify datasets was trained for 20 epochs. The training graphs show model accuracy and loss in Figs 4 for the respective models.

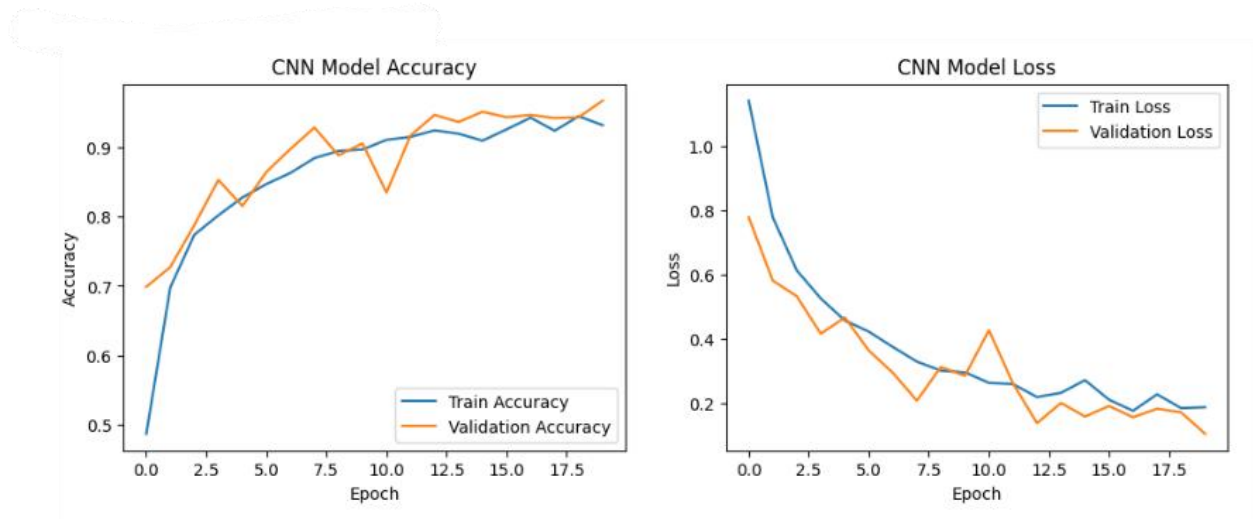


Figure 4: Training-progress graph for CNN model

### 3.6. Performance Evaluation

For evaluation we use the Precision, accuracy, recall, and F1 scores and to calculate and evaluate the model's performance as shown in Table 3. We create a confusion matrix to see the classification performance across different classes in Fig 5.

| Model | Class                                     | Precision (%) | Recall (%) | F1-Score (%) | Support |
|-------|-------------------------------------------|---------------|------------|--------------|---------|
| CNN   | Cherry_(including_sour)____Powdery_mildew | 100.00        | 91.00      | 95.08        | 224     |
|       | Cherry_(including_sour)____healthy        | 99.41         | 94.41      | 96.85        | 179     |
|       | Pepper,_bell____Bacterial_spot            | 84.36         | 97.27      | 90.36        | 183     |
|       | Pepper,_bell____healthy                   | 93.84         | 94.48      | 94.16        | 290     |

|                      |                                          |       |       |       |     |
|----------------------|------------------------------------------|-------|-------|-------|-----|
| Accuracy             |                                          | 94.06 |       |       | 876 |
| Macro avg            |                                          | 94.00 | 94.00 | 94.00 | 876 |
| Weighted avg         |                                          | 94.86 | 94.02 | 94.09 | 876 |
| <b>KNN</b>           | Cherry_(including_sour)___Powdery_mildew | 66.37 | 99.55 | 79.64 | 224 |
|                      | Cherry_(including_sour)___healthy        | 64.75 | 94.41 | 76.82 | 179 |
|                      | Pepper,_bell___Bacterial_spot            | 91.89 | 18.58 | 30.91 | 183 |
|                      | Pepper,_bell___healthy                   | 84.71 | 70.69 | 77.07 | 290 |
| Accuracy             |                                          | 72.03 |       |       | 876 |
| Macro avg            |                                          | 77.09 | 71.27 | 66.07 | 876 |
| Weighted avg         |                                          | 77.13 | 72.05 | 67.89 | 876 |
| <b>Decision Tree</b> | Cherry_(including_sour)___Powdery_mildew | 80.26 | 83.48 | 81.84 | 224 |
|                      | Cherry_(including_sour)___healthy        | 64.56 | 74.30 | 69.09 | 179 |
|                      | Pepper,_bell___Bacterial_spot            | 59.62 | 50.82 | 54.87 | 183 |
|                      | Pepper,_bell___healthy                   | 72.60 | 70.34 | 71.45 | 290 |
| Accuracy             |                                          | 70.43 |       |       | 876 |
| Macro avg            |                                          | 69.76 | 69.61 | 69.61 | 876 |
| Weighted avg         |                                          | 70.22 | 69.90 | 69.85 | 876 |
| <b>Random Forest</b> | Cherry_(including_sour)___Powdery_mildew | 91.60 | 97.32 | 94.37 | 224 |
|                      | Cherry_(including_sour)___healthy        | 87.85 | 88.83 | 88.33 | 179 |
|                      | Pepper,_bell___Bacterial_spot            | 86.50 | 77.05 | 81.50 | 183 |
|                      | Pepper,_bell___healthy                   | 86.39 | 87.59 | 86.99 | 290 |
| Accuracy             |                                          | 88.13 |       |       | 876 |
| Macro avg            |                                          | 88.09 | 88.07 | 88.07 | 876 |
| Weighted avg         |                                          | 88.98 | 88.74 | 88.74 | 876 |

Table 3: Performance result of each class of CNN on dataset

The diagonal values of the confusion matrix represent the true positives of our model classification. To calculate correctly the false negatives, we will have to sum up all the values in the row with the corresponding class of interest, excluding the values from the true positives. In doing this, we will then have the total number of testing samples for any class computed as TP + FN. For FP, we would be able to calculate this as the sum of the values in the corresponding column, excluding, of course, the true positives for that class. Furthermore, to obtain the number of true negatives for a given

class, we would sum all values in the two columns and rows, excluding of course the respective row and column of the class in question. Accuracy is a strong indicator and is defined as the ratio of correct classification concerning the total number of classifications. It is the sum of all the diagonal elements divided by the total count of items in the confusion matrix. This metric provides the overall effectiveness and reliability of the predictive model. Using the confusion matrix in Fig. 5 we can infer several key performance measures using the following equations.

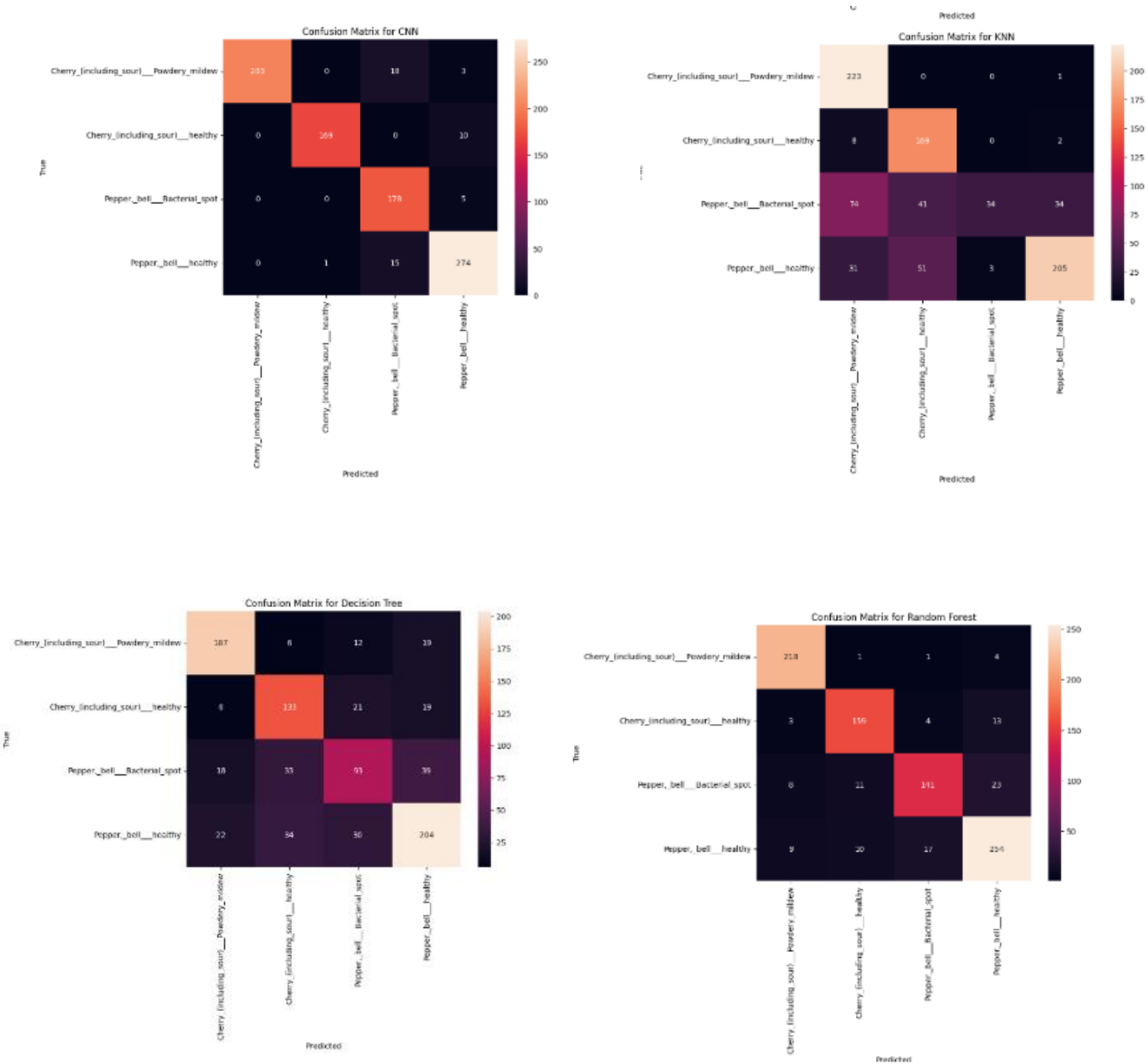


Figure 5: Confusion matrix

## 4. Result Analysis and Visualization

The CNN model performed best, with 94.06% accuracy, with a high and very strong performance for the model about precision, recall, and F1-scores within all classes. Moreover, training history also depicted enhancement concerning both accuracy and loss. It is followed by the Random Forest model, which performed well in all classes with balanced precision and recall at an accuracy of 88.13%. For the KNN model, performance was good, though it has a high recall but very low precision and recall for \*Pepper\_bell\_\_Bacterial\_spot\*, giving a general accuracy of 72.03%. The Decision Tree performed moderately, with 70.43%, though poor for the \*Pepper\_bell\_\_Bacterial\_spot\* classification. This was further confirmed by visualizing the training and validation accuracy and loss that CNN is more efficient in learning than classical models. On the whole, deep learning models such as CNN significantly outperformed classical machine learning models in plant disease classification.

## 5. Conclusion

The work has focused on deep learning model efficiency, more specifically, CNNs for image classification of plant diseases, reaching an accuracy of 94.06%. Also, CNN outperforms other traditional machine learning models tested, like K-Nearest Neighbors, Decision Trees, and Random Forests, where their performances were low and did not generalize well for some classes. This is further supported by the high precision, recall, and F1-scores obtained from the CNN model for critical categories like \*Cherry\_(including\_sour)\_\_Powdery\_mildew\* and \*Pepper\_bell\_\_healthy\*. These reflect how deep learning techniques may be used to enhance the detection and classification of plant diseases with higher precision than has ever been possible. The study is a highbrow case for the wide adoption of state-of-the-art, AI-driven techniques to guarantee food security through the control of crop health. The study will be continued for optimizing models and other expansions to be made on the already provided dataset for different plant species and diseases to provide holistic solutions.

## 6. References

1. Grünig, Marc, et al. "Applying deep neural networks to predict incidence and phenology of plant pests and diseases." *Ecosphere* 12.10 (2021): e03791.
2. Sharma, Rahul, et al. "Plant Disease Diagnosis and Image Classification Using Deep Learning." *Computers, Materials & Continua* 71.2 (2022).
3. Akshai, K. P., and J. Anitha. "Plant disease classification using deep learning." *2021 3rd International Conference on Signal Processing and Communication (ICPSC)*. IEEE, 2021.
4. Chen, J., Zhang, D., & Nanekaran, Y. A. "Identifying plant diseases using deep transfer learning and enhanced lightweight network." *IEEE Access* 11 (2023): 10286031.
5. Suresh, J., Barath, S., Kumar, D. V. S., Janakiraman, P., & Sekar, D. R. "An Effective Approach for Plant Leaf Diseases Classification Based on a Novel DeepPlantNet Deep Learning Model." *IEEE Access* 11 (2023): 10410019.
6. Subasi, A., Ghulam, A., & Sadikoglu, F. "A Mobile-Based System for Maize Plant Leaf Disease Detection and Classification Using Deep Learning." *IEEE Access* 10 (2022): 9855925.
7. Kaur, H., Ghosal, S., & Singh, V. "A Mobile-Based System for Maize Plant Leaf Disease Detection and Classification Using Deep Learning." *Frontiers in Plant Science* 14 (2023): 1079366.