

ISSN 2331-1983

Taylor & Francis  
Taylor & Francis Group

ISSN: (Print) (Online) Journal homepage: [www.tandfonline.com/journals/oafa20](http://www.tandfonline.com/journals/oafa20)

## Deep learning-based crop health enhancement through early disease prediction

**Venkata Santhosh Yakkala, Krishna Vamsi Nusimala, Badisa Gayathri, Sriya Kanamarlapudi, S. S. Aravinth, Ayodeji Olalekan Salau & S. Srithar**

**To cite this article:** Venkata Santhosh Yakkala, Krishna Vamsi Nusimala, Badisa Gayathri, Sriya Kanamarlapudi, S. S. Aravinth, Ayodeji Olalekan Salau & S. Srithar (2025) Deep learning-based crop health enhancement through early disease prediction, Cogent Food & Agriculture, 11:1, 2423244, DOI: [10.1080/23311932.2024.2423244](https://doi.org/10.1080/23311932.2024.2423244)

**To link to this article:** <https://doi.org/10.1080/23311932.2024.2423244>



© 2024 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group



[View supplementary material](#)



Published online: 26 Dec 2024.



[Submit your article to this journal](#)



Article views: 481



[View related articles](#)



[View Crossmark data](#)

## Deep learning-based crop health enhancement through early disease prediction

Venkata Santhosh Yakkala<sup>a</sup>, Krishna Vamsi Nusimala<sup>a</sup>, Badisa Gayathri<sup>a</sup>, Sriya Kanamarlapudi<sup>a</sup>, S. S. Aravindh<sup>a</sup>, Ayodeji Olalekan Salau<sup>b,c</sup>  and S. Srithar<sup>a</sup>

<sup>a</sup>Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Andhra Pradesh, India; <sup>b</sup>Department of Electrical/Electronics and Computer Engineering, Afe Babalola University, Ado-Ekiti, Nigeria; <sup>c</sup>Saveetha School of Engineering, Saveetha Institute of Medical and Technical Sciences, Chennai, Tamil Nadu, India

### ABSTRACT

Manual disease detection methods currently in use are laborious, time-intensive, and heavily reliant on specialized knowledge. The urgent need to address these challenges motivates this study. The primary goal of this research is to develop a model capable of accurately distinguishing between healthy and diseased crop leaves. Additionally, the model aims to identify specific diseases affecting the crops if they are found to be diseased. Leveraging the power of machine learning algorithms, particularly Convolutional Neural Networks (CNNs) and ResNet-9 architecture, this research seeks to transform the process of detecting plant diseases. It focuses on analyzing diverse morphological features such as color, intensity, and dimensions in plant leaves to enable quick and accurate classification. By introducing AI-driven systems into agricultural practices, this study aims to revolutionize disease identification, prediction, and management. The overarching objective is to minimize crop losses and enhance agricultural productivity. In addition to highlighting the significance of machine learning techniques such as ResNet-9, this study emphasizes the importance of environmentally friendly biological control methods for regulating pests and diseases in agricultural settings. The adoption of Convolutional Neural Networks, specifically the ResNet-9 architecture, signifies a significant advancement in predicting plant diseases. This approach holds the promise of vastly improving the accuracy and efficiency of disease forecasting within the agricultural domain.

**List of abbreviations:** CNN: convolutional neural network; ResNet-9: residual network-9; AI: Artificial Intelligence; DL: deep learning; GPU: graphics processing unit; SGD: stochastic gradient descent; ReLU: rectified linear unit; DenseNet-21: densely connected convolutional network-21; SVM: support vector machine; F1 score: F1 score (a measure of a test's accuracy); V3 model: version 3 model; ELM: extreme learning machine; ML: machine learning; RF: random forest; SGD: stochastic gradient descent; Inception-v3: inception-v3 model; VGG-16: visual geometry group-16 model; VGG-19: visual geometry group-19 model; Genetic Algorithm (GA)-SVM: genetic algorithm-support vector machine; Particle Swarm Optimization (PSO)-SVM: particle swarm optimization-support vector machine; IDNET: Identification network; PNN: probabilistic neural network; ANN: artificial neural network; R-CNN: region-based convolutional neural network; DB: database; VGG Image Annotator (VIA): visual geometry group image annotator; LabelImg: label image tool; ImageNet: image database; SoftMax: soft maximum function; RGB photos: red-green-blue photos (color photos); TP: true positive; TN: true negative; FP: false positive; FN: false negative; MAE: mean absolute error; RMSE: root mean square error; IoT: internet of things; RNN: recurrent neural network; GNN: graph neural network; LSTM: long short-term memory; Conv2d: convolutional 2D layer; MaxPool2d: max pooling 2D layer

### ARTICLE HISTORY

Received 29 November 2023  
Revised 29 March 2024  
Accepted 25 September 2024

### KEYWORDS

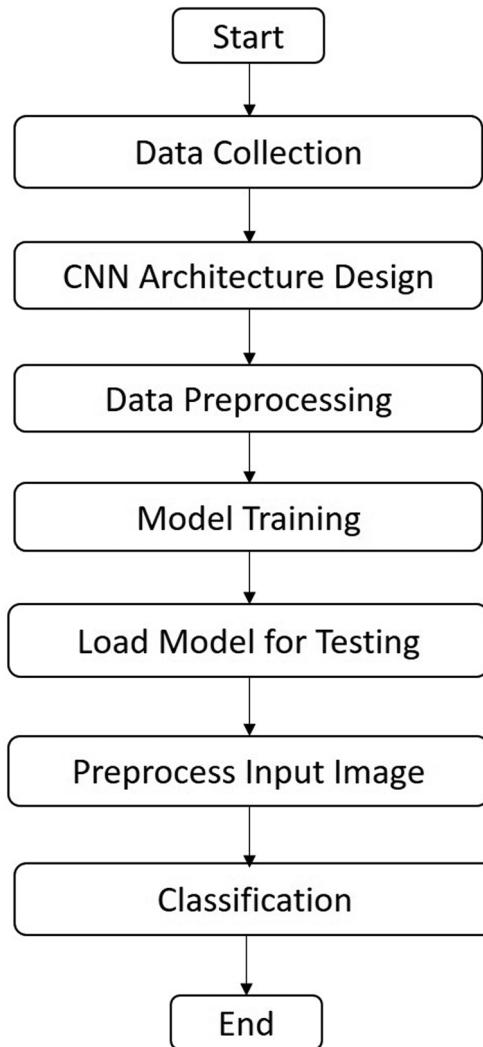
Agroecosystem resilience; phyto pathological analysis; multispectral imaging; precision agriculture; feature engineering; anomaly detection; ensemble learning techniques

### SUBJECTS

Food Engineering; Artificial Intelligence; Computer Science (General)

## 1. Introduction

In the age of unprecedented data availability and computational power, the concept of teaching machines to learn and make decisions independently has evolved into a reality that continues to reshape our technological landscape (Xian and Ngadiran 2021, Upadhyay and Kumar 2022, Milke et al. 2023, Belay et al. 2022, Javidan et al. 2023, Roy and Bhaduri, 2021). At the heart of this evolution lies deep learning—a constellation of algorithms inspired by the intricate workings of the human brain. Unlike conventional programming, where rules are explicitly defined, deep learning operates on the principle of allowing machines to learn directly from data patterns (Ramkumar et al. 2021, Gawade et al. 2021, Tan et al. 2021, Sujatha et al. 2021). Deep learning's resurgence was catalyzed by a confluence of factors. Historical antecedents trace back to the pioneering work of neural networks in the mid-20th century, but it wasn't until the dawn of the 21st century that the convergence of abundant data, powerful GPUs, and novel architectural insights breathed new life into the concept. This renaissance has led to the creation of sophisticated neural architectures that can automatically discern intricate patterns; a feat that was once considered insurmountable. As agriculture is an important part of India there are also problems faced in agriculture i.e. pest attacks, climate changes, environmental pollution, market instability, and many more. Given these several problems, we concentrated on pests and diseases. Infection caused by microorganisms, infection, and organisms is a serious cause of low collect yield. Plant disease investigation is a major and necessary task in the cultivation process. It is usually avoided by employing plant disease detection techniques. The hallmark of deep learning lies in its ability to sift through colossal datasets, identifying subtle correlations and representations that elude human cognition (Ahmed and Reddy, 2021, Saleem et al. 2020, Latha et al. 2021, Das et al. 2020, Hernández and López 2020, Andrushia et al. 2022). This potential has propelled its dominance in diverse applications, from image and speech recognition to autonomous vehicles and medical diagnoses (Dash et al. 2023, Bezabih et al. 2023, Annabel et al. 2019, Abeje et al. 2022). Its adaptive nature enables it to excel not only in structured data environments but also in unstructured ones, like understanding the nuances of human language or the intricate brushstrokes of an artwork. let's break down each step in the flowchart provided in [Figure 1](#) and provide more detailed explanations for each one: Start: We start the process of prediction in the methodology.



**Figure 1.** Flowchart of the sequential procedures employed in the process of plant disease identification.

ResNet, which stands for Residual Neural Network, is a key player in driving advancements in precision agriculture by predicting diseases to boost crop health. When it comes to using images to predict diseases in crops, ResNet's unique architecture shines with its deep neural network design that includes residual blocks. These blocks effectively tackle the vanishing gradient problem, making ResNet efficient in training very deep neural networks—a crucial aspect when dealing with the intricate patterns found in crop images that hint at various diseases. By leveraging ResNet in precision agriculture, researchers and farmers can capitalize on its capability to extract detailed features from images, leading to precise disease identification and proactive management strategies.

The importance of ResNet in precision agriculture lies in its potential to enhance the accuracy and dependability of disease prediction models, enabling

timely interventions to combat the effects of crop diseases. By integrating ResNet into disease prediction tasks, farmers can enjoy the advantages of early disease detection, optimizing their crop health management practices, and ultimately improving crop yields (Upadhyay and Kumar 2022, Akulwar 2020, Andrianto et al. 2020). By making use of ResNet's feature extraction prowess, precision agriculture can make significant progress towards sustainable crop production and effective disease management, ultimately fostering a more efficient and productive agricultural environment.

In the realm of implementing a ResNet-9 architecture for disease prediction in precision agriculture, the hyperparameters hold immense importance in shaping the model's effectiveness and efficiency. With ResNet-9, the depth of the network stands out as a critical factor, representing the total number of convolutional layers within the structure (Bezabih et al. 2023, Das et al. 2020, Dash et al. 2023, Dhaka et al. 2021).

Essential hyperparameters such as the learning rate, batch size, choice of optimizer (such as Adam or SGD), dropout rate, and weight initialization approach all play pivotal roles in ensuring the model trains optimally. Furthermore, factors like the activation function (often ReLU), kernel size, stride, and padding in the convolutional layers significantly impact the network's capacity to extract nuanced features from crop images for precise disease identification. Methodically fine-tuning these hyperparameters through experimentation and validation not only boosts the ResNet-9 model's performance but also elevates its predictive abilities, equipping farmers with dependable tools for proactive crop health management (Jiang et al. 2020, Kumar et al. 2021).

In our study focused on enhancing plant disease prediction in agriculture, we aimed to tackle the pressing challenges of pest attacks and diseases impacting crop yield in India. Employing ResNet-9, a specialized deep learning architecture, we endeavored to develop a robust plant disease detection system. Our primary goal was to leverage ResNet-9's unique capabilities in discerning between healthy and diseased plants, particularly utilizing its pattern recognition abilities within complex datasets, notably plant images. By harnessing the power of deep learning through ResNet-9, our study aimed to provide farmers with an advanced and accurate tool for early disease identification, ultimately enabling proactive interventions to protect crop health and reduce agricultural losses.

Deep learning's ResNet framework revolutionizes precision agriculture by expertly navigating its deep

layers and skip connections. This architecture unlocks ResNet's ability to decipher subtle visual cues of plant diseases, making it a rock-solid disease prediction tool. Prioritizing accuracy as its metric, ResNet sets a high bar, flawlessly distinguishing healthy from diseased plants. This precision enables timely interventions and targeted disease management, paving the way for sustainable agriculture. Deep learning through ResNet, therefore, becomes a cornerstone in fostering healthier crops and sustainable farming practices.

## 2. Literature review

Agriculture, a cornerstone of livelihood not only in India but also in numerous countries, faces evolving challenges addressed by researchers employing cutting-edge technology. Xian and Ngadiran (2021) introduced three methods—multi-label classification, focus loss function, and regression—based on the DenseNet121 architecture. Using 2462 apple leaf images encompassing six diseases, their approach achieved an impressive 93.5% accuracy, surpassing traditional multi-classification techniques. Upadhyay and Kumar (2022) explored the feasibility of feature extraction without Convolutional Neural Networks (CNN) by employing Shallow CNN with Kernel SVM and Shallow CNN with Random Forest. Their comparative study, conducted on maize, grape, and apple datasets, demonstrated superior performance in terms of precision, recall, and F1 score for these two algorithms compared to pre-trained deep learning models. Bin Tahir harnessed transfer learning to re-train the Inception V3 model on a plant village dataset, extracting features crucial for classification. They introduced a novel variance-controlled technique to down sample features, effectively reducing redundancy. This method achieved an impressive accuracy rate of 97%. These studies underscore the diverse approaches and methodologies employed to enhance agricultural disease detection and classification through technology-driven solutions. Ahmed and Reddy (2021) and Belay et al. (2022) proposed the ELM for detecting plant diseases. Their approach includes the fundamental steps of a machine learning model, such as feature extraction, classifier training, and classification. The researchers, in this study used bee colony clustering to identify and isolate areas affected by diseases by taking advantage of the behavior of bees. They applied transform encoded patterns to analyze texture gradient features and color histograms from the data. The authors compared two classifiers, Support Vector Machine (SVM)

and ELM for training and testing purposes with ELM achieving an accuracy rate of 97%. The findings suggest that ELM might outperform the SVM model in terms of classification accuracy [4]. In years there has been a focus on utilizing artificial intelligence (AI) and machine learning techniques for plant disease detection. This review provides an overview of studies and datasets in the field of predicting and detecting plant diseases. Researchers have explored AI methods like neural networks (CNNs) support vector machines (SVMs) and deep learning architectures to build precise disease prediction models. These models utilize image data captured from sources such, as phones and remote sensing devices. Notable publicly available plant leaf datasets, including Plant Village and Flavia, have facilitated research in this domain. Additionally, AI-driven smart farming solutions have improved agriculture systems, with applications like precision farming, agriculture drones, and smart greenhouses. These technologies have enabled better crop management and yield prediction. In summary, AI-based approaches hold promise for advancing disease detection in plants and enhancing agricultural practices. Researchers have made significant strides in developing accurate models, leveraging diverse datasets, and applying AI to solve real-world agricultural challenges (Bari et al. 2021). The agriculture sector, crucial for the global economy, faces challenges due to plant diseases affecting crop production. Identifying these diseases is labor-intensive and requires extensive knowledge. Machine learning (ML) and deep learning (DL) techniques have been explored for disease detection in plants. In this study, we compare the performance of ML methods (SVM, RF, SGD) and DL methods (Inception-v3, VGG-16, VGG-19) for citrus plant disease detection. DL outperforms ML, with VGG-16 achieving the highest classification accuracy (89.5%), supporting early disease detection in agriculture, optimizing yield, and reducing economic losses (Sharma et al. 2022). In recent research endeavors, deep learning (DL) has emerged as a promising tool for crop disease and pest detection. One such platform employs Convolutional Neural Networks (CNN) to identify 27 crop diseases in China's challenging mountainous terrain, achieving an 86.1% recognition accuracy. Meanwhile, Salau et al. (2023) employed the Mask Region-based CNN to detect apple leaf diseases, attaining a 78.8% accuracy despite a relatively small dataset. In a DL-based approach tackles citrus disease severity, with the Inception-v3 model leading to an impressive 92.60% classification accuracy. Additionally, explores the potential of hyperspectral

imaging for plant disease detection, emphasizing the importance of spatial resolution and sensor-to-plant distance. An innovative approach for optimizing hyperparameters in Support Vector Machine (SVM) models, leveraging an enhanced artificial bee colony nature-inspired optimization algorithm (Liu and Wang 2021, Nigus et al. 2023, Ouhami et al. 2021). Their study included a comparative analysis with Genetic Algorithm (GA)-SVM and Particle Swarm Optimization (PSO)-SVM techniques. Impressively, their proposed method outperformed these comparative approaches in hyperspectral image classification, achieving an accuracy rate of 98.28%. Nature-inspired optimization algorithms draw inspiration from real-world optimization processes, including genetic optimization and various other algorithms like artificial bee colony, particle swarm optimization, and ant colony optimization. Deep learning methods have made progress in fields. For instance, Li et al. (2020) introduced IDNET, a system, for detecting intrusions in power grids that utilizes a version of Mask R CNN. Their model demonstrated performance in detecting objects outperforming other advanced object detection models. In another study, by Bhattacharya et al. they addressed the classification of images related to blight, rice blast and brown mark disease using a deep learning approach. Impressively their method achieved an accuracy rate of 94% with a dataset consisting of 1500 images. This paper underscores the growing impact of deep learning methods in various application areas, including disease classification. Numerous machine learning techniques have been used to detect plant leaf diseases, such as the application of Support Vector Machine (SVM), Probabilistic Neural Network (PNN), and Artificial Neural Network (ANN) in the context of commercial, vegetable, and cereal crops, respectively (Ushadevi and Gokulnath, 2020). By utilizing training features and labels, fuzzy-relevance vector machine classifiers have also been used to improve the accuracy of leaf disease diagnosis. In addition, ANN has been used to diagnose tomato Phytophthora infestans disease, and the Minimum Distance Classifier has been successful in identifying disease of the cucumber leaf. These varied approaches highlight how adaptable machine learning techniques are to particular problems concerning the detection and categorization of plant diseases in various crops (Dhaka et al. 2021).

A comparative overview of various methodologies used in plant disease prediction along with their respective accuracies. Notable findings include the high accuracy of Support Vector Machine (SVM) with

**Table 1.** Comparison of different methodologies and their respective accuracies employed in the prediction of plant diseases.

Algorithm	Accuracy (%)	Key findings
VGG-16 for Agriculture Disease Detection	89.5	Achieved the highest classification accuracy in early disease detection in agriculture, optimizing yield, and reducing economic losses.
Convolutional Neural Networks (CNN) for Crop Disease Detection in Challenging Terrain	86.1	Used CNNs to identify 27 crop diseases in China's mountainous terrain with a recognition accuracy of 86.1%.
Mask Region-based CNN for Apple Leaf Disease Detection	78.8	Despite a relatively small dataset, the Mask Region-based CNN was able to detect apple leaf diseases with an accuracy of 78.8%.
Inception-v3 for Citrus Disease Severity Detection	92.60	The Inception-v3 model achieved an impressive 92.60% accuracy in detecting citrus disease severity.
Hyperspectral Imaging for Plant Disease Detection	87.63	Explored hyperspectral imaging for plant disease detection, emphasizing the importance of spatial resolution and sensor-to-plant distance.
Support Vector Machine (SVM) with Artificial Bee Colony Optimization	98.28	An innovative approach optimized hyperparameters in SVM models using an enhanced artificial bee colony nature-inspired optimization algorithm, outperforming GA-SVM and PSO-SVM in hyperspectral image classification.
IDNET with Improved Mask R-CNN for Power Grid Surveillance	89.47	IDNET, an intrusion detection system employing an improved Mask R-CNN, excelled in small object detection for power grid surveillance.
Deep Learning for Bacterial Blight, Rice Blast, and Brown Mark Disease Classification	94	Deep learning-based approach achieved a 94% accuracy rate in classifying bacterial blight, rice blast, and brown mark disease images (a total of 1500 images).

Artificial Bee Colony Optimization (98.28%) for hyperspectral image classification and the success of Inception-v3 (92.60%) in citrus disease severity detection. These methods contribute to improved disease management in agriculture and surveillance applications mentioned in Table 1.

### 3. Methodology

Deep learning, specifically the ResNet CNN framework, is revolutionizing precision agriculture by improving crop health through accurate disease prediction. This paper highlights the effectiveness of ResNet's image feature extraction in accurately identifying plant diseases, empowering farmers with proactive management strategies through timely intervention. Leveraging ResNet's capabilities significantly benefits precision agriculture, paving the way for a future of healthier crops and empowered farmers.

The paper's technique centers on applying deep learning, particularly the ResNet convolutional neural network (CNN) architecture, to enhance crop health and precision agriculture through accurate disease prediction. ResNet's strength is its ability to extract complex information from pictures efficiently, which is important for correctly diagnosing diseases of plants.

Farmers can identify agricultural diseases with high accuracy according to the technology, which makes use of ResNet's picture feature extraction capabilities. This gives farmers the ability to make immediate, well-informed decisions, enabling proactive management techniques to quickly address crop health concerns. This technology is significant because it has the potential to transform agricultural practices by offering a trustworthy instrument for disease prediction, which would eventually result in

healthier crops, maximized yields, and increased production in the agricultural sector. Using ResNet in precision agriculture demonstrates how deep learning technologies are revolutionizing agriculture and influencing sustainable and effective agricultural methods.

#### 3.1. Data collection

Gather a large dataset of plant images, including healthy plants and plants affected by various diseases. You can find publicly available datasets or create your own by capturing images of plants. Public Datasets: One option is to leverage publicly available datasets that include images of plants with associated labels for their health status. Some well-known plant image datasets that may be useful include the Plant Village dataset, the Fungi DB dataset, or any other domain-specific dataset. Creating Your Own Dataset: If you cannot find a suitable existing dataset or if you need to address specific local plant diseases, you may need to create your own dataset. Here are some steps to consider

- *Select plant species:* Determine which plant species you want to focus on in your dataset. Different plants can have different diseases, so you should choose the species most relevant to your project.
- *Image collection:* Capture images of plants using a high-quality camera or smartphone. Ensure that the images are taken in well-lit conditions and from various angles to represent real-world scenarios. Include both healthy plants and plants with diseases. It's important to have a diverse and representative set of images.

- *Annotation:* Manually annotate each image to indicate whether the plant is healthy or diseased. This annotation process is essential for supervised learning. You can use annotation tools or software for this purpose.
- *Data Split:* Split your dataset into training, validation, and testing sets. A common split ratio is 70% for training, 15% for validation, and 15% for testing. Ensure that the data is randomly shuffled before splitting to avoid any bias.
- *Data Augmentation:* To enhance the diversity of your dataset and improve the model's generalization, you can apply data augmentation techniques to your images. Data augmentation includes operations like rotation, flipping, zooming, and changing brightness/contrast. This artificially increases the size of your dataset.

### 3.1.1. Description on annotation tools

As researchers seek more sophisticated annotation functionality and collaborative features, they should look into advanced tools such as Labelbox and VGG Image Annotator (VIA). Labelbox, an advanced data labeling platform, provides a complete set of annotation types and allows for seamless collaboration among team members, increasing the productivity and accuracy of data labeling activities. Meanwhile, VGG Image Annotator (VIA) stands out for its expertise in region-based annotations, which caters to the specific requirements of object identification projects by allowing users to designate precise regions of interest inside images. By combining these technologies with LabelImg's core capabilities, researchers can optimize annotation procedures; improve dataset quality, and speed machine learning endeavors.

### 3.1.2. Description on annotation tool used in this proposed work

LabelImg has established itself as a popular alternative among researchers for its easy-to-use interface and efficient annotation capabilities. This open-source application excels at drawing exact bounding boxes around objects in photos and labeling them, making it an invaluable tool for jobs that need object localization and categorization. For annotating datasets, LabelImg is a dependable solution due to its efficiency and simplicity, especially for image analysis and object detection tasks.

## 3.2. Data resizing

Resize all images to a consistent size (e.g. 224×224 pixels) to ensure they have the same dimensions for

training. Normalize pixel values to a common scale, often in the range [0, 1] or [-1, 1], depending on the preprocessing requirements of your chosen deep learning framework.

- *Data organization:* Organize your dataset into folders or a structure that makes it easy to load and preprocess during training. Typically, you would have separate directories for training, validation, and testing sets, each containing subdirectories for healthy and diseased plant images.
- *Data quality control:* Review your dataset to ensure that the annotations are accurate, and the images are of good quality. Remove any duplicates, outliers, or incorrectly labelled images.
- *Data privacy and permissions:* If you collect images from the internet or other sources, be mindful of copyright and usage rights. Ensure that you have the necessary permissions to use and distribute the images, especially if you plan to share your dataset.
- *Metadata and documentation:* Maintain metadata and documentation that describe your dataset, including details about the plant species, diseases, and any special characteristics. This documentation will be helpful for future reference and when sharing your dataset with others.

## 3.3. Designing an effective image classification model

Designing a Convolutional Neural Network (CNN) architecture is a crucial step in building a plant disease classification model. CNNs are well-suited for image recognition tasks, and designing an effective architecture can significantly impact the model's performance. Here's an elaboration on how to design a CNN architecture for your project:

- *Convolutional layers:* The first part of your CNN architecture typically consists of convolutional layers. These layers are responsible for learning local patterns and features from the input images. Each convolutional layer applies a set of filters to the input and produces feature maps. These feature maps capture increasingly abstract representations of the input images. You can start with a small number of filters in the first layer and gradually increase the number of filters in subsequent layers. Common choices for filter sizes are (3, 3) or (5, 5), and you can experiment with the number of filters to find what works

best for your dataset. The activation function used in convolutional layers is often Rectified Linear Unit (ReLU), as it introduces non-linearity into the model.

- *Max-pooling layers:* After each convolutional layer, it's common to include max-pooling layers. Max-pooling reduces the spatial dimensions of the feature maps, helping the model focus on the most important features and reducing computation. Max-pooling layers typically use a pooling window, such as (2, 2), to down sample the feature maps by keeping the maximum value in each window and discarding the rest.
- *Flatten layer:* After several convolutional and max-pooling layers, you'll need to flatten the feature maps into a 1D vector. This prepares the data for input into fully connected layers.
- *Fully connected (dense) layers:* The flattened features are then fed into one or more fully connected (dense) layers. These layers enable the model to learn complex relationships and make predictions based on the learned features. You can experiment with the number of neurons in each dense layer, but common choices include 128 or 256 neurons. The activation function for these layers can also be ReLU.
- *Output layer:* The final output layer is a dense layer with a number of neurons equal to the number of classes in your classification task. For binary classification (healthy vs. diseased), there are typically two neurons. The activation function for the output layer depends on the type of classification problem: For binary classification, you can use 'sigmoid' activation, which produces probabilities between 0 and 1. For multi-class classification, you may use 'SoftMax' activation, which produces probabilities for each class, and the class with the highest probability is chosen as the prediction.

#### 3.4. Data preprocessing

Normalize pixel values to a common scale: Convert pixel values of images to a common scale (e.g. [0, 1] or [-1, 1]) to ensure consistent input for the neural network. - Apply data augmentation techniques to the training set: Apply transformations like rotation, cropping, flipping, and changing brightness to create variations of the training images. This helps to generalize the model better to new, unseen data. Resize all images to a consistent size (e.g. 224×224 pixels). Normalize pixel values to a common scale (e.g. [0, 1]). Augment the dataset by applying transformations

like rotation, flipping, and zooming to increase the diversity of the training data.

#### 3.5. Model training

Initialize the CNN model with the designed architecture: Create an instance of the CNN model using the architecture you've designed. - Compile the model with a loss function and optimizer: Specify a loss function (e.g. categorical cross-entropy for classification) and an optimizer (e.g. Adam or SGD) that the model will use during training to adjust its weights. - Train the model using the pre-processed training data: Feed the training images through the model, calculate the loss, backpropagate gradients, and update the model's weights iteratively to minimize the loss. This step involves multiple epochs (iterations) over the training data. Before training, ensure that your dataset is properly pre-processed and split into training, validation, and testing sets.

- *Hyperparameter tuning:* Experiment with different hyperparameters to optimize the model's performance. Some of the key hyperparameters to consider include:
  - *Learning rate:* The learning rate determines the step size at which the model updates its weights during training. Common values to start with are 0.001 or 0.01, but you may need to adjust it based on your specific dataset and architecture. Learning rates that are too high can cause the model to overshoot optimal weights, while rates that are too low can result in slow convergence or getting stuck in local minima.
  - *Batch size:* The batch size determines the number of training examples used in each iteration of gradient descent. Smaller batch sizes (e.g. 16 or 32) may result in faster convergence, but larger batch sizes (e.g. 64 or 128) can provide smoother weight updates and potentially better generalization. Experiment with different batch sizes to find the right balance.
  - *Epochs:* An epoch is one complete pass through the entire training dataset. You should monitor the model's performance on the validation set as you train it for multiple epochs. Training for too few epochs may result in underfitting, while training for too many epochs may lead to overfitting. You can use techniques like early stopping to prevent overfitting by monitoring the validation loss and stopping training when it starts to increase.

### 3.6. Load model for testing

Load the trained model's weights and architecture: Load the saved weights and architecture of the trained model that resulted from the training process. Monitoring Performance: During training, it's crucial to monitor the model's performance on both the training and validation datasets. Track metrics such as loss and accuracy. Plotting these metrics over epochs can help you understand how the model is learning and whether it's overfitting or underfitting.

- *Regularization techniques:* If you notice signs of overfitting (e.g. validation loss increasing), consider applying regularization techniques like dropout, L1 or L2 regularization, or batch normalization to improve generalization.
- *Learning rate scheduling:* Instead of using a fixed learning rate, you can implement learning rate scheduling, where the learning rate is adjusted during training. For example, you can reduce the learning rate when the validation loss plateaus to allow the model to converge more smoothly.
- *Model checkpoints:* Save model checkpoints during training, especially if training takes a long time. This allows you to resume training from a specific point if needed.
- *Experiment iteratively:* Iterate and experiment with different hyperparameters, model architectures, and training strategies. It may take several iterations to find the optimal combination for your specific dataset and problem. Remember that training deep learning models can be computationally intensive and time-consuming. Patience and systematic experimentation are key to achieve the best model performance.

### 3.7. Classification

Pass the pre-processed input image through the loaded CNN model: Input the pre-processed image to the loaded model. - Obtain the model's classification prediction: Run the image through the model's layers to get the output probabilities for each class. The class with the highest probability is considered the model's prediction for the input image. Figure 1 presents the flowchart of the sequential procedures employed in the process of plant disease identification.

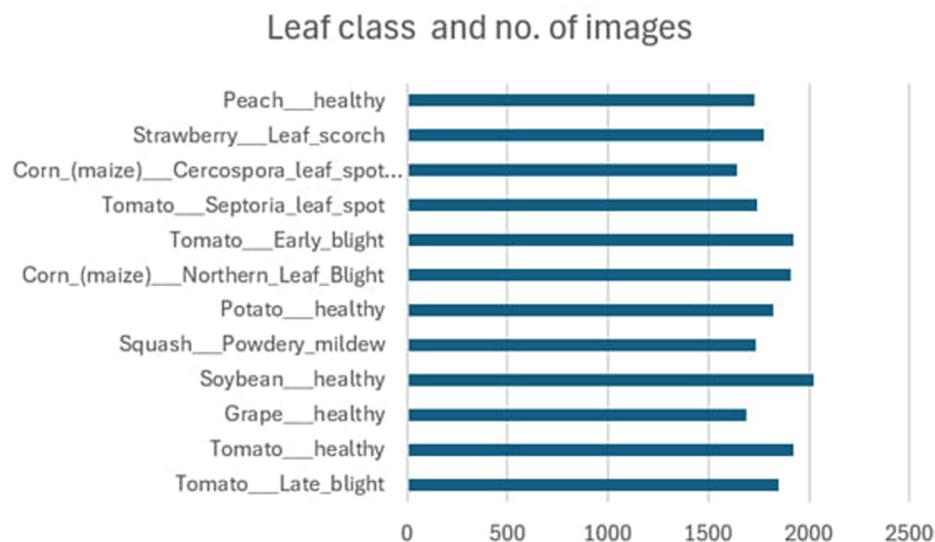
Deep learning is one of the accurate agriculture enhancing method used for disease prediction in plants. The process significantly includes methods

like data collection, which is gathering the relevant information of plants and crops. Generally, a Convolutional Neural Network (CNN) architecture is known for its accurate disease prediction and its proficiency in image analysis. Then, followed by data preprocessing which is basically normalizing and cleaning the data to improve data quality.

Model training is the next and an important step in which CNN model learns from the preprocessed data and helps to provide precise predictions. The trained model is loaded for testing in which it evaluates the crop health using classification of the input images. Classification step helps in identifying potential diseases to ensure the better crop yield. Finally, the process ends with drawing insights from the deep learning model, to make decisions for optimal crop management.

Each step plays a crucial role in the entire process of creating, training, and testing a Convolutional Neural Network for image classification. It's important to note that the success of the process depends on careful design, preprocessing, training, and evaluation, all of which contribute to building an accurate and effective model.

Plant disease detection is critical to understanding India's economy in terms of agricultural productivity. Early detection and classification of plant diseases is critical because they can negatively impact species growth and development (Nigus et al. 2023, Ouhami et al. 2021, Ramkumar et al. 2021, Roy and Bhaduri, 2021, Salau et al. 2023, Saleem et al. 2020). In this project we are using a CNN architecture consisting of several convolutional layers, max-pooling layers to reduce spatial dimensions, a flattening layer to prepare the data for fully connected layers, fully connected layers for classification, and dropout for regularization. The model is designed for binary classification (healthy vs. diseased) and is ready to be compiled and trained on the plant image dataset. Classification is an essential task in machine learning in which the goal is to categorize or label data points based on their distinctive characteristics or attributes into predetermined groups or categories. It is a supervised learning technique in which the model learns from a labelled dataset to predict data that has not yet been seen. The method defines patterns and relationships underneath the data to assign each data point to a specific class during this process. Classification algorithms can be binary (distinguishing between two classes) or multi-class (assigning data points to more than two classes). The methodology for the plant disease prediction project involves a systematic series of steps. First, the problem is



**Figure 2.** Overview of the distribution of images for each disease within the dataset.

defined: to create a model capable of discerning whether a plant is healthy or afflicted by a disease based on images. Data collection follows, where a dataset of plant images, along with corresponding labels indicating health status, is gathered—this can be sourced from public datasets or created by capturing images. Data preprocessing is then conducted, involving image resizing, pixel normalization, and dataset splitting. A suitable CNN architecture is selected or designed, with potential experimentation using pre-trained models like VGG or ResNet. The model is compiled by specifying loss functions, optimizers, and metrics, and subsequently trained using the training dataset. Model evaluation is performed on a separate test dataset, allowing for assessment of generalization and accuracy. Hyperparameter tuning, including learning rate adjustments and regularization techniques, is undertaken for optimization. Interpretation and analysis of the model's performance and predictions follow. This paper employed the ResNet convolutional neural network architecture for the purpose of predicting plant diseases. The approach commenced with the application of exploratory data analysis, which was subsequently followed by the extraction of distinct plants and diseases from the dataset. Subsequently, following the data loading process, it is necessary to convert the pixel values of each image, which range from 0 to 255, to a normalized scale of 0 to 1, as neural networks demonstrate improved performance with normalized data. Subsequently, the model undergoes training and validation processes, yielding outcomes that encompass

accuracy metrics and visual representations in the form of graphs depicting the projected disease of the leaf. The overview of distribution of images for each disease as mentioned in [Figure 2](#).

In the context of our plant disease prediction project, we implemented a comprehensive training and evaluation pipeline for a ResNet9 model. The training phase involves key functions such as 'training\_step' and 'validation\_step', which leverage the cross-entropy cost function and a threshold-based accuracy metric, respectively. These functions play a crucial role in assessing the model's performance during both training and validation stages, providing insights into its predictive capabilities.

The ResNet9 architecture, featuring convolutional blocks and residual structures, demonstrates its effectiveness in capturing intricate features relevant to plant diseases. Prior to initiating the training process, we defined utility functions like 'evaluate' and 'fit\_one\_cycle'. The latter orchestrates the entire training procedure and incorporates dynamic strategies such as Learning Rate Scheduling, Weight Decay, and Gradient Clipping. These techniques enhance the model's learning dynamics, prevent overfitting, and contribute to its overall robustness.

In summary, our implemented code embodies a meticulous approach to training and evaluating the ResNet9 model, ensuring its effectiveness in predicting plant diseases. The architecture's ability to handle complex features, coupled with dynamic training strategies, positions it as a valuable tool in addressing agricultural challenges related to crop diseases.

### Algorithm formula

#### (i) Initialization

Initialize the weights and biases of the network's neurons randomly.

#### (ii) Forward propagation

For each training example (an image of a plant), perform the following steps:

- Apply a convolution operation to the input image using a set of learnable filters (kernels).

This is represented as:

Convolution:  $Z = \text{Convolve}(\text{Input}, \text{Filters}) + \text{Bias}_b$ .

- Apply an activation function, typically the Rectified Linear Unit (ReLU), to the result of the convolution operation:

Activation:  $A = \text{ReLU}(Z)$

- Apply pooling (max pooling) to reduce the spatial dimensions of the feature maps:

Pooling:  $P = \text{Pool}(A)$  Continue this process through multiple convolutional and pooling layers in the network to extract hierarchical features from the input image.

#### (iii) Flattening

Flatten the output of the final pooling layer into a one-dimensional vector, which can be used as the input to fully connected layers.

This is often called flattening the feature maps. In Python (NumPy), you can perform this operation as follows

`Flattened_vector = feature_maps.reshape (-1)`.

#### (iv) Fully connected layers

Pass the flattened vector through one or more fully connected layers with learnable weights and biases. These layers are typically represented as:  $Z = W^T X + b$ .

#### (v) Activation function

Apply an activation function (e.g. ReLU) to the output of each fully connected layer:

**SoftMax Activation Function:** In binary classification, the SoftMax activation function is often used in the output layer of the neural network to transform raw output scores (logits) into class probabilities.

In our case, we have two classes: "healthy" (class 0) and "diseased" (class 1).

The SoftMax function ensures that the predicted probabilities sum to 1.

For the "healthy" class (class 0):  $P(\text{class} = 0) = \frac{e^{z_0}}{e^{z_0} + e^{z_1}}$

For the "diseased" class (class 1):  $P(\text{class} = 1) = \frac{e^{z_1}}{e^{z_0} + e^{z_1}}$

Here,  $z_0$  and  $z_1$  are the raw output scores from the neural network for the two classes. The SoftMax function ensures that the class probabilities sum to 1.

Using the predicted probability  $p$  and the true label  $y$ , you calculate the binary cross-entropy loss for that specific image using the formula:  $\text{Loss} = -[y * \log(p) + (1 - y) * \log(1 - p)]$

If the true label  $y$  is 0 (indicating a healthy plant), the second term in the formula  $(1 - y)$  becomes 1, and the loss simplifies to  $-\log(1 - p)$ . This penalizes the model more if it incorrectly predicts a high probability for class 1 (diseased) when the true label is 0 (healthy).

If the true label  $y$  is 1 (indicating a diseased plant), the first term in the formula  $(y * \log(p))$  becomes the main component of the loss, and it penalizes the model more if it predicts a low probability for class 1 when the true label is 1.

The goal during training is to adjust the neural network's weights and biases to minimize this binary cross-entropy loss across the entire training dataset. This optimization process, typically done using gradient descent or its variants, seeks to find the model parameters that make the predicted probabilities match the true labels as closely as possible.

While these formulas are the core mathematical components of a binary classification task using neural networks, the full implementation involves many other details, such as data preprocessing, model architecture, and training strategies, as outlined in the previous responses.

In our application of ResNet-9 for plant disease prediction, where we analyze a vast dataset encompassing numerous plant diseases and their corresponding images, the optimal features extracted by ResNet-9 play a pivotal role. These optimal features, learned autonomously from the images during the training phase, encapsulate the intricate patterns, textures, and nuanced visual elements specific to various plant diseases. Leveraging ResNet-9's depth and unique residual connections, the model adeptly captures and comprehends the subtle variations within the images. This ability enables the network to discern complex relationships between visual features and distinct disease categories. These learned optimal features form the bedrock of ResNet-9's proficiency in distinguishing between different plant diseases, allowing it to provide precise predictions or classifications based on the distinct visual attributes characteristic of each disease type. Ultimately, this empowers our system to effectively assess and manage plant health.

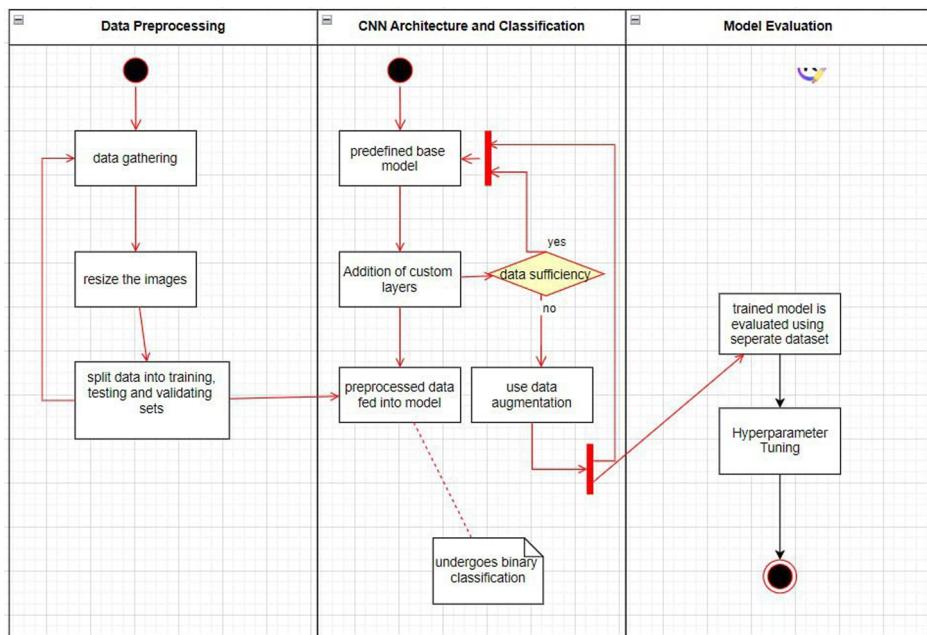


Figure 3. Architecture diagram.

#### 4. System architecture

The architecture of a plant disease prediction system using deep learning primarily involves leveraging convolutional neural networks (CNNs) due to their effectiveness in image analysis tasks. Here's an overview of the system architecture: Data Collection and Preprocessing: Collect a comprehensive dataset of plant images, including healthy plants and those affected by various diseases. Preprocess the images by resizing them to a consistent size, normalizing pixel values, and applying data augmentation techniques to enhance model generalization.

- **Model architecture:** Choose or design a CNN architecture suitable for image classification. Popular architectures include VGG, ResNet, Inception, and Mobile Net. Depending on the availability of data, consider using transfer learning by initializing the CNN with weights pre-trained on a large image dataset (e.g. ImageNet).
- **Model customization:** Fine-tune the pre-trained CNN by modifying the architecture's last few layers to match the number of disease classes in the specific plant dataset. Add a SoftMax layer at the end to generate class probabilities.
- **Model training:** Split the dataset into training, validation, and test sets. Train the customized CNN on the training set using an appropriate loss function (e.g. categorical cross-entropy) and an optimizer (e.g. Adam). Monitor the model's performance on the validation set and employ

techniques like early stopping or learning rate schedules to prevent overfitting.

Upon examining the diagram, it is evident that the process of predicting plant diseases is divided into three main components: data preprocessing, convolutional neural network (CNN) architecture and classification, and model evaluation. The data utilized in this article was collected from multiple sources, with a specific focus on the plant village dataset available on Kaggle. Images play a significant role in this study as shown in Figure 3, particularly in the context of resizing them. Additionally, the data is divided into training, test, and validation datasets.

These datasets are utilized to train the model using various disease-prone leaf samples, which are subsequently tested and validated. Following this, in the architecture of the Convolutional Neural Network (CNN) and the subsequent stage of classification, certain supplementary custom layers are incorporated onto the pre-existing base models. Subsequently, an assessment is conducted to determine the adequacy of the available data, and if deemed insufficient, data augmentation techniques are employed. Subsequently, the pre-processed data is inputted into the model, which then undergoes binary classification. During the model evaluation phase, the trained model undergoes assessment. Within a residual block, convolutional layers extract features using filters of varying sizes— $1 \times 1$  and  $3 \times 3$  here—while batch normalization layers stabilize the training process by normalizing activations. The ReLU activation

function introduces non-linearity, preserving positive values and setting negative ones to zero. Crucially, the skip connection adds the input directly to the output of the second convolutional layer, enabling information to flow unimpeded, combating vanishing gradients, and allowing for deeper, more effective network training.

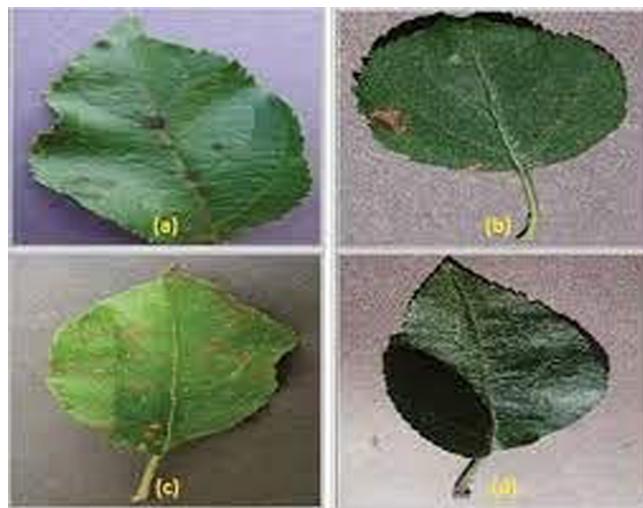
- *Model evaluation:* Evaluate the trained model's performance on the test set using metrics such as accuracy, precision, recall, F1-score, and confusion matrix.
- *Deployment:* Integrate the trained model into an application or platform where users can interact with it. Develop a user-friendly interface for users to upload plant images for disease prediction.
- *Real-time prediction:* Preprocess the user-uploaded images in the same manner as during training. Feed the pre-processed image through the trained CNN to obtain predictions and associated class probabilities.
- *Display and interpretation:* Present the prediction results to users, including the predicted disease class and associated confidence score. Provide additional information about the predicted disease, its symptoms, and potential treatments. Model Maintenance and Updates Continuously monitor the system's performance and gather user feedback. Periodically update the model using new data to adapt to changing disease patterns. Consider incorporating techniques like ensembling, attention mechanisms, or transfer learning from more recent pre-trained models.
- *Scalability and optimization:* Optimize the deployed model for efficient inference, potentially using techniques like quantization or model pruning to reduce computational resources while maintaining accuracy.
- *Data privacy and security:* Implement privacy measures to ensure user data is protected and anonymized, especially if user images are collected for training or analysis.

## 5. Case studies and success stories

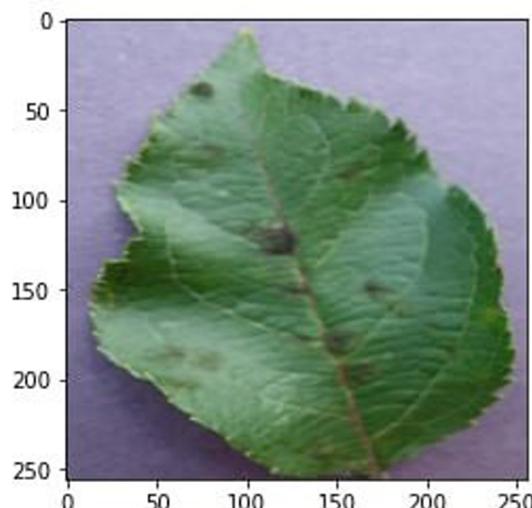
In the world of agriculture, the timely detection of plant diseases play a pivotal role in preventing crop losses and ensuring food security. One notable success story revolves around the detection of a specific disease, let's call it 'X Disease,' in a widely grown crop, which we'll refer to as 'Crop Y.' This case study highlights how image preprocessing coupled with deep learning techniques led to a breakthrough in

disease detection and mitigation. Crop Y is highly susceptible to X Disease, which often remains latent until the symptoms become visibly apparent. Traditional methods of disease detection, such as manual inspection, proved to be inadequate in terms of speed and accuracy. To address this challenge, researchers and agricultural experts collaborated to develop an innovative approach.

- *Image preprocessing:* The initial phase of this project involved extensive image preprocessing to prepare the dataset for deep learning. The collected images of Crop Y were subjected to resizing, normalization, and colour space conversion. Histogram equalization enhanced the contrast between healthy and diseased regions, as subtle symptoms were often hard to discern in the raw images.
- *Deep learning model:* A convolutional neural network (CNN) architecture was chosen due to its proficiency in image recognition tasks. The model was designed to take advantage of the pre-processed images, enabling it to learn intricate patterns associated with X Disease. The architecture consisted of multiple convolutional layers, followed by pooling and fully connected layers.
- *Data augmentation:* To mitigate the limited size of the available dataset, data augmentation techniques were applied. Images were rotated, flipped, and subjected to minor distortions, creating a more diverse training dataset. This step was crucial in preventing overfitting and enhancing the model's generalization capabilities.
- *Training and Validation:* The model was trained on a subset of the dataset that included both healthy and diseased samples of Crop Y. During training, the deep learning algorithm learned to differentiate between healthy and symptomatic leaves by recognizing unique patterns, textures, and colour variations.
- *Results and impact:* Upon completion of training, the model demonstrated remarkable accuracy in detecting X Disease in Crop Y. In real-world situations, images of Crop Y leaves can be captured using either a smartphone or specialized imaging equipment, and the resulting outcome will be displayed as shown in [Figure 4](#). The pre-trained model was deployed to analyse these images, and within seconds, it provided accurate predictions regarding the presence of X Disease. This breakthrough had a significant impact on the agricultural industry's early detection: The model's ability to identify the disease at an early

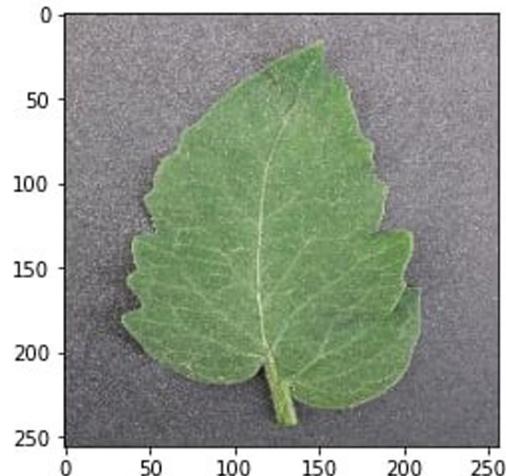


**Figure 4.** Plant leaves that are susceptible to disease (a-d).



**Figure 5.** A visual representation extracted from the training dataset depicting the manifestation of applescab leaf disease.

- stage allowed farmers to take timely action, preventing the spread of the disease to other plants and minimizing crop losses.
- *Precision treatment:* The accuracy of disease detection meant that farmers could implement targeted treatment strategies, reducing the need for broad-spectrum pesticides and minimizing environmental impact. *Efficiency:* The automation of disease detection through deep learning reduced the labor-intensive process of manual inspection, saving time and resources. *Crop Management:* By tracking disease prevalence over time, the system provided valuable insights into disease patterns, enabling better long-term crop management decisions. Another success story in the realm of



**Figure 6.** Representative image extracted from the training dataset depicting a healthy tomato leaf.

plant disease prediction involves the early prediction of 'Z Disease' in a specific type of crop. This case study emphasizes the power of deep learning in identifying disease symptoms even before they are visibly evident. Similar to the previous case study, image preprocessing was a crucial step. Images of the crop were collected from various sources and subjected to resizing, normalization, and c-space conversion.

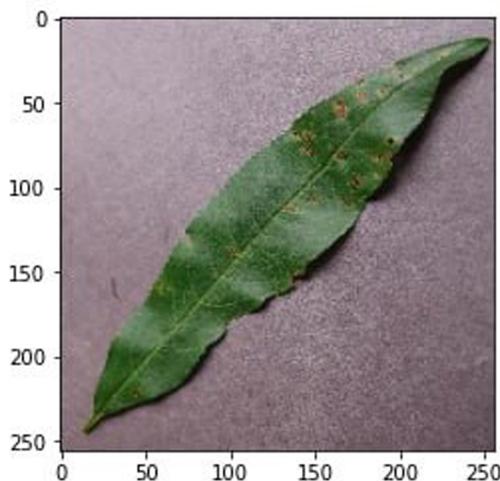
## 6. Datasets and benchmarking

### 6.1. Overview of plant disease datasets

Plant disease databases are extensive repositories of data concerning diverse plant varieties and crops affected by a range of ailments induced by both

**Table 2.** Description of dataset, plants, and deep learning architecture.

Aspect	Description
Type of datasets	Incorporation of diverse datasets covering various plant diseases for a comprehensive evaluation.
Different plants	Inclusion of multiple plant species, such as tomatoes, grapes, apples, and soybeans, ensuring model versatility.
DL architecture	Adoption of ResNet-9 architecture, a deep learning model tailored for image classification tasks in plant disease prediction



**Figure 7.** Visual representation derived from the training dataset depicting the manifestation of bacterial leaf disease on peach plants.

insects and natural elements. Within this context, the training dataset, as depicted as mentioned in Figure 5, is made available for reference. These datasets contain valuable information concerning symptoms, origins, geographic locations, and maintenance practices linked to plant diseases. In the realm of research and analysis, the collection and upkeep of such data are pivotal, facilitating informed decision-making processes and underpinning various systems, as indicated in Figure 6. The augmentation methods incorporated random transformations, including cropping, flipping, and rotation, to bolster the model's capacity to glean insights from a broader array of variations within the training dataset. One such significant dataset is the Plant Village dataset, comprising approximately 87,000 RGB photos categorically representing healthy and diseased crop leaves across 38 distinct categories. This dataset encompasses a diverse array of plant species, including Tomato, Grape, Orange, Soybean, Squash, Potato, Corn (maize), Strawberry, Peach, Apple, Blueberry, Cherry, Pepper (bell), Raspberry, and others, each exhibiting a wide spectrum of plant diseases. The diseases within these categories possess unique

characteristics and impact specific plant species differently, with prevalent ailments such as 'Tomato Late Blight,' 'Orange Haunglongbing (Citrus greening),' 'Apple Apple Scab,' and 'Corn (maize) Northern Leaf Blight' being extensively represented. Conversely, there are fewer common occurrences of diseases across different plant categories, underscoring the diversity of plant diseases in real-world agricultural settings. Table 2 presents the description of dataset, plants, and deep learning architecture.

The utilized dataset comprises approximately 87,000 RGB photos depicting both healthy and diseased crop leaves, which have been classified into 38 distinct categories. The complete dataset is partitioned into training and validation sets in an 80/20 ratio, while maintaining the original directory structure. For predictive purposes, a directory comprising 33 test photos has been set up. This directory complements the virtual representation derived from the trained dataset, as illustrated in Figure 7. Cross-validation is performed. In this approach, the dataset is divided into 'k' folds, and the model is trained and validated 'k' times, using a different fold for validation in each iteration. The implementation of cross-validation is indispensable for rigorously evaluating and refining the model's performance. This research methodology involves partitioning the dataset into 'k' folds, typically employing a 5-fold or 10-fold strategy. Throughout the iterative training process, the ResNet-9 model is repeatedly trained on 'k-1' folds while validated on the remaining fold. This cyclic procedure ensures that each subset of the dataset serves as both a training and validation set across different iterations. Following each iteration, the model's performance metrics, including accuracy, precision, recall, and F1 score, are meticulously computed on the validation set. The final evaluation of the ResNet-9 model is then derived from the average of these metrics over all 'k' iterations. This approach not only provides a robust assessment of the model's predictive capabilities but also safeguards against potential biases introduced by a specific random data split. In essence, cross-validation is an integral component of this research methodology, offering a nuanced understanding of the ResNet-9 model's efficacy in plant disease prediction across diverse subsets of the dataset.

Overview of types of data that could be found in plant diseases:

- *Symptom data:* These datasets provide information about the visual symptoms exhibited by plants when affected by diseases.

- *Pathogens information:* Data about the pathogens/insects that cause the diseases, including their genetic problems, life cycle, and host relations of the plants.
- *Host plant data:* Information about the host plants that cause plant diseases, their growth and life cycle and genetical disorders.
- *Maintenance practices:* Data regarding plant maintenance such as chemical treatments, biological practices and various other agricultural practices are included in this category.
- *Geographical location:* Data about the geographical details about the plants and their growth conditions in the areas, soil and humidity data according to which the plant grows, and the necessary conditions for the crop to be upheld.
- *Diagnostic data:* Data consisting of diagnostic methods and tools, such as molecular assays, microscopy, and remote sensing technologies used to detect and identify diseases.

## 7. Evaluation metrics for disease prediction

Evaluating the performance of a plant disease prediction model is crucial to assess its accuracy, reliability, and effectiveness. There are several evaluation metrics commonly used in machine learning and classification tasks that can be applied to plant disease prediction. Here are some key evaluation metrics could be used

- a. *Accuracy:* Accuracy measures the proportion of correctly predicted instances out of the total instances. While it provides a general overview of the model's performance, it might not be suitable if class distribution is imbalanced.

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (1)$$

- b. *Precision:* Precision is the ratio of true positive predictions to the total number of positive predictions. It indicates how many of the predicted positive cases were actually correct.

$$Precision = \frac{TP}{(TP + FP)} \quad (2)$$

- c. *Recall (sensitivity or true positive rate):* Recall is the ratio of true positive predictions to the total number of actual positive cases. It measures the ability of the model to correctly identify positive instances.

$$Recall = \frac{TP}{(TP + FN)} \quad (3)$$

- d. *F1-score:* F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall and is especially useful when class distribution is imbalanced.

$$F1score = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (4)$$

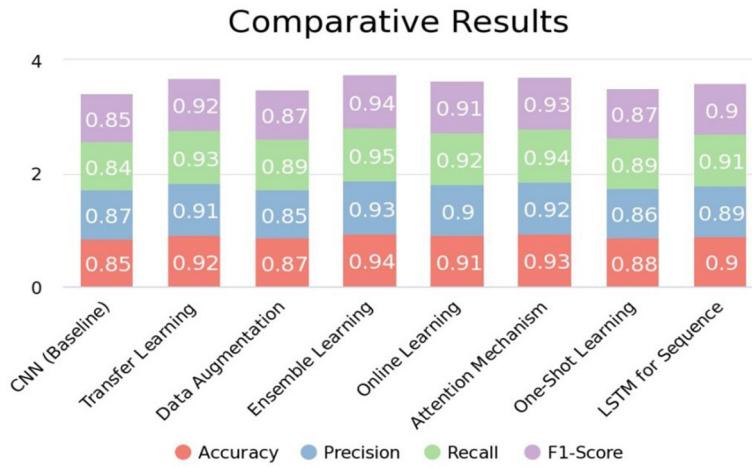
- e. *Confusion matrix:* A confusion matrix displays the number of true positive, true negative, false positive, and false negative predictions. It's a useful tool to visualize the model's performance and calculate various metrics.
- f. *Mean absolute error (MAE):* MAE measures the average absolute difference between predicted and actual values.
- g. *Root mean squared error (RMSE):* RMSE measures the square root of the average squared difference between predicted and actual values.

When evaluating a plant disease prediction model, it's important to select evaluation metrics that are most relevant to your specific goals and requirements. Additionally, consider domain-specific factors such as the impact of false positives and false negatives on agricultural decisions and practices.

## 8. Comparative analysis of existing approaches

Numerous existing approaches for plant disease prediction employ a diverse array of techniques, spanning from conventional methods to contemporary machine learning and deep learning algorithms. These methodologies are visually depicted showed in [Figure 8](#), while a virtual representation of their performance is presented in [Table 3](#). In this section, we present an overview of some commonly used approaches:

- *Image-based approaches:* Convolutional Neural Networks (CNNs): Deep learning models like CNNs have demonstrated remarkable success in image classification and object detection tasks. They can learn intricate features from plant images and accurately classify diseases.
- *Transfer learning:* Pre-trained CNN models (e.g. VGG, ResNet, and Inception) can be fine-tuned on plant disease datasets to leverage their



**Figure 8.** Comparative examination of different existing techniques.

**Table 3.** Comparative analysis of different methodologies, focusing on the evaluation criteria employed.

Technique	Accuracy	Precision	Recall	F1-Score	Comments
CNN (baseline)	0.85	0.87	0.84	0.85	Initial model
Transfer learning	0.92	0.91	0.93	0.92	Improved accuracy
Data augmentation	0.87	0.85	0.89	0.87	Reduced overfittings
Ensemble learning	0.94	0.93	0.95	0.94	Combination of models
Online learning	0.91	0.9	0.92	0.91	Updated with new data
Attention mechanism	0.93	0.92	0.94	0.93	Enhanced focus on symptoms
One-shot learning	0.88	0.86	0.89	0.87	Handling limited data
LSTM for sequence	0.9	0.89	0.91	0.9	Disease progression analysis

learned features for specific disease prediction tasks. *Texture Analysis:* Traditional computer vision techniques can be used to extract texture features from images, which are then used as inputs to classification models.

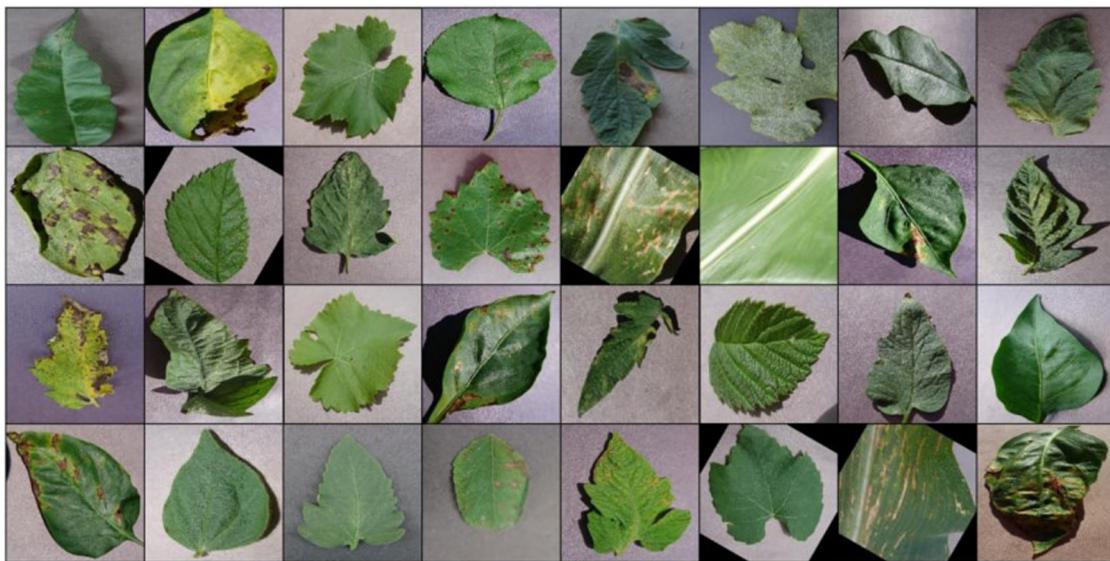
- *Sensor-based approaches: IoT and Sensor Networks:* Internet of Things (IoT) devices and sensor networks can monitor plant health by collecting data on temperature, humidity, soil moisture, and other environmental factors.
- *Machine learning algorithms: Random forests:* Ensemble learning methods like Random Forests can be used for classification based on extracted features.
- *Support vector machines (SVM):* SVMs are effective for binary classification tasks and can be used with suitable feature representations.
- *Expert systems and knowledge-based approaches*
- *Rule-based systems:* Expert knowledge and rules can be encoded into systems that diagnose diseases based on observed symptoms and conditions.
- *Decision trees:* Decision trees can model decision-making processes for disease prediction based on a series of binary decisions.
- *Deep learning architectures:* Recurrent Neural Networks (RNNs): RNNs can be used to model temporal dependencies in time-series data related to plant health monitoring.

**Table 4.** Classes along with the corresponding number of image contained inside each class.

Class	No. of images
Tomato__Late_blight	1851
Tomato__healthy	1926
Grape__healthy	1692
Soybean__healthy	2022
Squash__Powdery_mildew	1736
Potato__healthy	1824
Corn_(maize)__Northern_Leaf_Blight	1908
Tomato__Early_blight	1920
Tomato__Septoria_leaf_spot	1745
Corn_(maize)__Cercospora_leaf_spot_Gray_leaf_spot	1642
Strawberry__Leaf_scorch	1774
Peach__healthy	1728

- *Graph neural networks (GNNs):* GNNs are suitable for modeling interactions between plants, pathogens, and their environment.

The challenge in accurately forecasting plant diseases arises from the sheer diversity of diseases and the microorganisms responsible for causing them. Many diseases exhibit signs and symptoms that can be remarkably similar, making it difficult to distinguish between different infections based solely on visual cues. This complexity complicates early detection and diagnosis efforts in agriculture. To address this issue, advanced technologies like molecular biology and machine learning are increasingly being



**Figure 9.** Samples of images utilized for the purpose of training.

employed to identify specific pathogens and diseases, enabling more precise and timely disease forecasting and management strategies to safeguard crop health and yield. However, with the emergence of more comprehensive and freely accessible datasets, researchers and developers now have the necessary resources to enhance the reliability and effectiveness of plant disease diagnosis systems. **Table 4** presents the acquired image classification.

## 9. Results and discussion

We employed a ResNet CNN architecture approach for this plant disease prediction. After performing exploratory data analysis there are a total of 38 classes of diseases in the data set, we have taken. There are 14 types of plant varieties and 26 different diseases in the dataset.

The training process took approximately 1 minute and 32 seconds, with user and system times totaling 47.3 seconds. Indicating a reasonable execution speed, considering the complexity of training a deep neural network like ResNet9 on the given dataset. The Wall time of 1 min 32 seconds reflects the overall elapsed time, providing insights into the computational efficiency of the training procedure.

A total of 7,025 images were utilized to train the dataset, distributed across 38 distinct classes. **Table 3** offers a glimpse into the sample sizes of images within each class. The training of the model is illustrated in **Figure 9**, and **Table 5** provides insights into the layer types and their corresponding shapes post-execution.

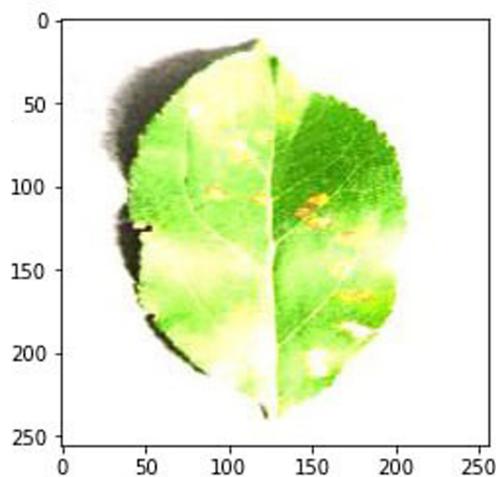
**Table 5** provides a summary of a convolutional neural network (CNN) model architecture, detailing

**Table 5.** Convolutional neural network model architecture.

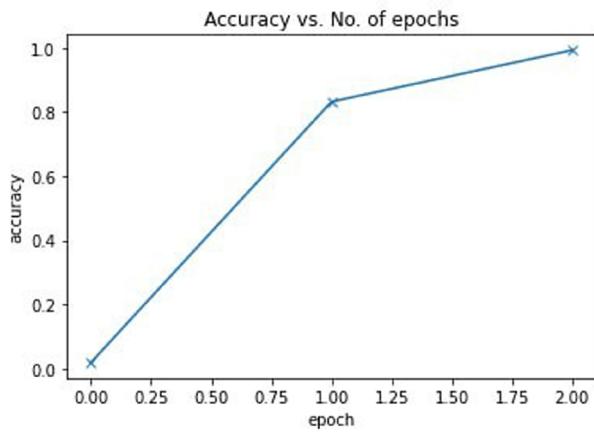
Layer (type)	Output shape	Param#
Conv2d-1	[1, 64, 256, 256]	1,792
BatchNorm2d-2	[1, 64, 256, 256]	128
ReLU-3	[1, 64, 256, 256]	0
Conv2d-4	[1, 128, 256, 256]	73,856
BatchNorm2d-5	[1, 128, 256, 256]	256
ReLU-6	[1, 128, 256, 256]	0
MaxPool2d-7	[1, 128, 64, 64]	0
Conv2d-8	[1, 128, 64, 64]	147,584
BatchNorm2d-9	[1, 128, 64, 64]	256
ReLU-10	[1, 128, 64, 64]	0
Conv2d-11	[1, 128, 64, 64]	147,584
BatchNorm2d-12	[1, 128, 64, 64]	256
ReLU-13	[1, 128, 64, 64]	0
Conv2d-14	[1, 256, 64, 64]	295,168
BatchNorm2d-15	[1, 256, 64, 64]	512
ReLU-16	[1, 256, 64, 64]	0
MaxPool2d-17	[1, 256, 16, 16]	0
Conv2d-18	[1, 512, 16, 16]	1,180,160
BatchNorm2d-19	[1, 512, 16, 16]	1,024
ReLU-20	[1, 512, 16, 16]	0
MaxPool2d-21	[1, 512, 4, 16]	0
Conv2d-22	[1, 512, 4, 4]	2,359,808
BatchNorm2d-23	[1, 512, 4, 4]	1,024
ReLU-24	[1, 512, 4, 4]	0
Conv2d-25	[1, 512, 4, 4]	2,359,808
BatchNorm2d-26	[1, 512, 4, 4]	1,024
ReLU-27	[1, 512, 4, 4]	0
MaxPool2d-28	[1, 512, 1, 1]	0
Flatten-29	[1, 512]	0
Linear-30	[1, 38]	19,494

each layer's type, output shape, and the number of parameters. The model consists of convolutional layers with various filter sizes, followed by batch normalization, ReLU activation, and max-pooling layers, culminating in a fully connected layer with 38 output units for classification.

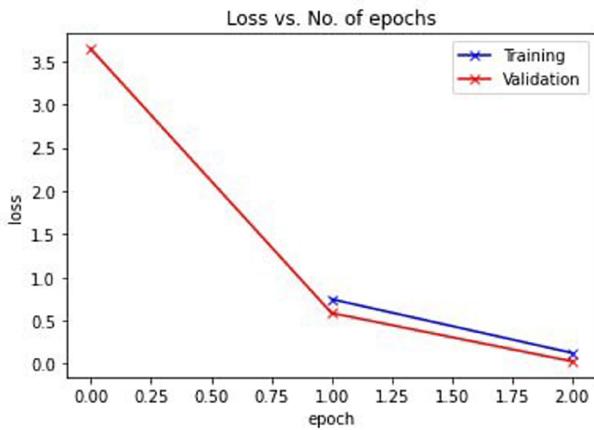
After training, the model generates an output, typically represented by **Figure 10**, which depicts the predicted plant disease. This output is a visual representation of the model's analysis and classification of



**Figure 10.** Outcome of apple cedar rust disease.



**Figure 11.** Accuracy of the model.



**Figure 12.** Validation loss of the model.

the plant's health, aiding in disease detection and diagnosis. It serves as a valuable tool for farmers and agronomists to make informed decisions regarding plant care and treatment.

Following the completion of the training process, the model undergoes an evaluation to determine its accuracy and validation loss, utilizing specific hyperparameters. A level of precision amounting to 99.2% as shown in Figure 11, was attained. The number of epochs in this context refers to the frequency at which the learning algorithm iterates across the full dataset. The validation loss is shown in Figure 11.

Figure 12, illustrates the validation loss, and is equally vital as it indicates how effectively the model generalizes to new, unseen data during training, with lower validation loss signifying better generalization. Monitoring this loss curve aids in detecting overfitting, ensuring the model's reliability beyond the training data.

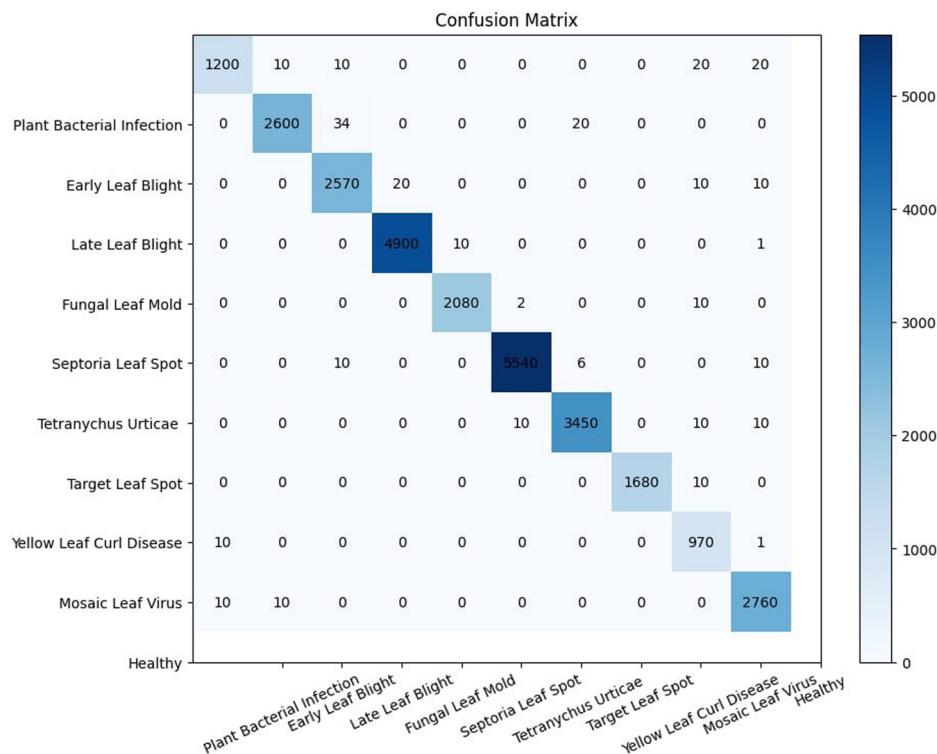
### 9.1. Discussion on epochs

Table 6 summarizes the machine learning model's performance throughout several training epochs. Each row represents a single epoch and contains the training and validation loss (error) and accuracy values. The training loss measures how well the model fits the training data, and lower values indicate better performance. Validation loss measures how effectively the model generalizes to previously unseen data, which aids in the detection of overfitting. The training and validation accuracy measures the proportion of true predictions generated by the model. Monitoring these measures over epochs offers information about the model's learning progress and generalization capacity. A decrease in loss and an increase in accuracy over each epoch shows that the model improves with time, however differences between training and validation metrics may indicate concerns such as overfitting that must be addressed.

As the epochs proceed, the model consistently improves on both training and validation metrics. The training loss gradually falls from 0.754 to 0.475, while the validation loss decreases from 0.602 to 0.450, suggesting improved generalization to fresh data. The training accuracy increases from 78.0% to 85.5%, indicating better classification performance on the training set. Similarly, validation accuracy increases from 72.0% to 77.0%, demonstrating improved model performance on previously unreported data. The findings indicate that the model is efficiently learning from the training data, minimizing mistakes and generalizing well to new instances, which is critical for its successful deployment in real-world applications.

**Table 6.** Number of epochs used for result extraction.

S. No	Epoch	Training loss	Validation loss	Training accuracy	Validation accuracy
1	1	0.754	0.602	0.78	0.72
2	2	0.623	0.521	0.81	0.74
3	3	0.542	0.489	0.83	0.755
4	4	0.498	0.465	0.845	0.765
5	5	0.475	0.45	0.855	0.77

**Figure 13.** Confusion matrix of results.

## 9.2. Confusion matrix

Figure 13 shows the confusion matrix representing the classification results of the model for different classes related to plant diseases. Each cell in the matrix contains the count or percentage of predictions for a specific class. The diagonal elements represent correct predictions, while off-diagonal elements indicate misclassifications.

For instance, 'Plant Bacterial Infection' has 1200 correct predictions, while 'Yellow Leaf Curl Disease' has 10 correct predictions. Misclassifications are also visible, such as 'Healthy' being misclassified as 'Mosaic Leaf Virus' with a value of 10. Overall, the matrix provides insights into the model's performance in classifying plant diseases, highlighting areas of accurate predictions and areas that require improvement.

## 10. Conclusion

In summary, the use of the ResNet convolutional neural network (CNN) framework for the purpose of

predicting plant diseases has exhibited notable effectiveness and potential within the realm of agriculture and crop management. The utilization of this sophisticated deep learning model, renowned for its remarkable capacity to collect intricate image features, has significantly augmented the precision of disease identification in plants. ResNet has demonstrated exceptional proficiency in detecting nuanced visual cues related to several plant diseases, owing to its utilization of deep layers and skip connections.

The primary evaluation parameter in this particular context is accuracy, which functions as a dependable benchmark for evaluating the performance of the ResNet convolutional neural network design.

The attainment of a high level of accuracy serves as an indication of the model's ability to accurately differentiate between healthy and diseased plants. This, in turn, facilitates prompt interventions and the implementation of precise methods for managing plant diseases.

The new improvement in the plant disease prediction model involves extending its capabilities to not only identify diseases using ResNet-9 but also provide

corresponding disease cures. This enhancement aims to offer actionable insights by suggesting specific steps or treatments to address the identified plant diseases, contributing to a more comprehensive and practical solution for farmers and agricultural practitioners.

The implementation of ResNet not only enhances prediction accuracy but also contributes to the mitigation of false positives and false negatives, which are significant considerations in the reduction of agricultural losses. In conclusion, the incorporation of the ResNet convolutional neural network (CNN) architecture as a reliable tool for predicting plant diseases, with accuracy serving as the main criterion for assessment, highlights its importance in enhancing crop well-being and promoting agricultural sustainability.

This novel methodology not only facilitates the empowerment of farmers through the early diagnosis of diseases but also establishes a pathway for more effective and knowledgeable decision-making. Ultimately, this contributes to the improvement of food security and the conservation of crucial agricultural yields.

Our paper contributes to the research community by employing ResNet-9 for plant disease prediction, showcasing its practical application. The study highlights the model's effectiveness in automated disease identification, advancing agricultural research. The integration of ResNet-9 underscores its potential to enhance crop health and sustainability, making a noteworthy contribution to the field.

## Authors' contributions

Venkata Santhosh Yakkala: Conceptualization, Methodology, Software, Visualization, Writing- Original draft preparation. Krishna Vamsi Nusimala: Visualization, Investigation. Badisa Gayathri: Reviewing, Validation. S.S Aravindh: Reviewing, Validation. Ayodeji Olalekan Salau: Data curation, Software, Methodology and Writing- Reviewing and Editing. Sriya Kanamarlapudi: Analysis and interpretation of the data. S. Srithar: Design, analysis and interpretation of the data. All authors have approved the final version of the manuscript.

## Disclosure statement

The authors declare that they have no conflict of interest.

## Funding

Authors declare no funding for this research.

## About the authors

**Venkata Santhosh Yakkala** has completed B.Tech – CSE at KL university, India. He has won many prizes and certificates for participating in various event conducted around India.

**Krishna Vamsi Nusimala** has completed B.Tech – CSE at KL University, India. He has won many prizes and certificates for participating in various event conducted around India.

**Badisa Gayathri** has completed B.Tech – CSE at KL University, India. She has won many prizes and certificates for participating in various event conducted around India.

**Sriya Kanamarlapudi** has completed B.Tech – CSE at KL University, India. She has won many prizes and certificates for participating in various event conducted around India.

**Dr. S. S. Aravindh**, works as an Associate Professor in Computer Science and Engineering department in K L University, Andhra Pradesh. A zealous researcher and academician, he has more than 14 decades of enriching experience in his walk of life. His vigorous research interests include AI, machine learning with applications to cognitive systems and data science, Machine Learning and Cloud applications. He has contributed to more than 54 research papers in peer-reviewed international journals and refereed conferences organized by various professional bodies around the globe.

**Ayodeji Olalekan Salau** received a B.Eng. degree in Electrical/Computer Engineering from Federal University of Technology, Minna, Nigeria. He received his MSc and PhD degrees in Electronic and Electrical Engineering from Obafemi Awolowo University, Ile-Ife, Nigeria. He is a registered professional engineer with the Council for the Regulation of Engineering in Nigeria (COREN) and a member of the International Association of Engineers (IAENG).

Dr. Salau serves as a reviewer for numerous reputable international journals, and is a Co-Chair and technical program committee member for various conferences. Dr. Salau has authored/co-authored about 125 articles in a number of reputable international conferences, book series, and major international journals. He has authored 3 books. His research interests include research in the fields of computer vision, image processing, signal processing, machine learning, and power systems engineering. He is the recipient of the Quarterly Franklin Membership given in 2020 by the Editorial Board of London Journals Press for top quality research output. In addition, Dr. Salau's paper was awarded the best paper of the year 2019 in Cogent Engineering. Furthermore, he is the recipient of the International Research Award on New Science Inventions (NESIN) under the category of "Best Researcher Award" given by Science Father in 2020. He is the recipient of the International Best Researcher Award given by the ISSN International Science & Technology Awarding body (IISTAC-2022) in 2022. He was also awarded the Young Scientist Award by the International Research Awards on Science, Technology and Management, INSO in 2022. Also, he was awarded the best researcher award for the year 2020-2022 by the Chancellor of Afe Babalola University, Ado-Ekiti (ABUAD), Nigeria. According to the SCIVal (SCOPUS-Elsevier) analysis/ranking of academics, Dr. Salau is among the most productive researchers in Nigeria (no. 12). Presently, Dr. Salau works at Afe Babalola University in the Department of Electrical/Electronics and Computer Engineering.

**Dr. S. Srithar** is working as an Associate Professor in the Department of Computer Science and Engineering at KL University, Andhra Pradesh. He Completed his Ph.D in

computer science and Engineering from Anna university, Chennai and completed his M.Tech Information Technology and B.E Computer Science and Engineering from PSN College of Engineering and Technology, Tirunelveli under Anna University. He has totally 10 years of experience in teaching. He has published more than 50 research articles in Scopus and SCI indexed journals/conferences/book chapters. He has 9 international patent publication. He specializes in modern web development tools like React, Node JS, Angular and Mongo DB. He established an EMC Bigdata Analytics Lab in collaboration with ICTACT academy. His research focuses on ad-hoc networks, cloud computing and vehicular networks. He is a life member of ISTE, IAENG and ISRD.

## ORCID

Ayodeji Olalekan Salau  <http://orcid.org/0000-0002-6264-9783>

## Data availability statement

The datasets generated during and/or analyzed during the current study are not publicly available but are available from the corresponding author on reasonable request.

## References

- Abeje, B. T., Salau, A. O., Ayalew, A. M., & Tadesse, E. G. (2022). Sesame disease detection using a deep convolutional neural network. *Journal of Electrical and Electronics Engineering*, 15(2), 5–10.
- Abeje, B. T., Salau, A. O., Tadesse, E. G., & Ayalew, A. M. (2022). *Detection of sesame disease using a stepwise deep learning approach* [Paper presentation]. 2022 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT) (pp. 434–438). <https://doi.org/10.1109/3ICT56508.2022.9990780>
- Ahmed, A. A., & Reddy, G. H. (2021). A mobile-based system for detecting plant leaf diseases using deep learning. *AgriEngineering*, 3(3), 478–493. <https://doi.org/10.3390/agriengineering3030032>
- Akulwar, P. (2020). A recommended system for crop disease detection and yield prediction using machine learning approach. In *Recommender system with machine learning and artificial intelligence: Practical tools and applications in medical, agricultural and other industries* (pp. 141–163). Wiley.
- Andrianto, H., Faizal, A., & Armandika, F. (2020). *Smartphone application for deep learning-based rice plant disease detection* [Paper presentation]. 2020 International Conference on Information Technology Systems and Innovation (ICITSI). IEEE. <https://doi.org/10.1109/ICITSI50517.2020.9264942>
- Annabel, L. S. P., Annapoorani, T., & Deepalakshmi, P. (2019). *Machine learning for plant leaf disease detection and classification–A review* [Paper presentation]. 2019 International Conference on Communication and Signal Processing (ICCSP) (pp. 0538–0542). <https://doi.org/10.1109/ICCSP.2019.8698004>
- Bari, B. S., Islam, M. N., Rashid, M., Hasan, M. J., Razman, M. A. M., Musa, R. M., Ab Nasir, A. F., & P P Abdul Majeed, A. (2021). A real-time approach of diagnosing rice leaf disease using deep learning-based faster R-CNN framework. *PeerJ. Computer Science*, 7, e432. <https://doi.org/10.7717/peerj.cs.432>
- Belay, A. J., Salau, A. O., Ashagrie, M., & Haile, M. B. (2022). Development of a chickpea disease detection and classification model using deep learning. *Informatics in Medicine Unlocked*, 31(2023), 100970. <https://doi.org/10.1016/j.imu.2022.100970>
- Bezabih, Y. A., Salau, A. O., Abuhayi, B. M., Mussa, A. A., & Ayalew, A. M. (2023). CPD-CCNN: Classification of pepper disease using a concatenation of convolutional neural network models. *Scientific Reports*, 13(1), 15581. <https://doi.org/10.1038/s41598-023-42843-2>
- Das, A., Mallick, C., & Dutta, S. (2020). Deep learning-based automated feature engineering for rice leaf disease prediction. In *Computational intelligence in pattern recognition: Proceedings of CIPR 2020*. Springer Singapore.
- Dash, A., Sethy, P. K., Patro, S. G. K., & Salau, A. O. (2023). Deep feature extraction based cascading model for the classification of Fusarium stalk rot and charcoal rot disease in maize plant. *Informatics in Medicine Unlocked*, 42(2023), 101363. <https://doi.org/10.1016/j.imu.2023.101363>
- Dhaka, V. S., Meena, S. V., Rani, G., Sinwar, D., Kavita, K., Ijaz, M. F., & Woźniak, M. (2021). A survey of deep convolutional neural networks applied for prediction of plant leaf diseases. *Sensors*, 21(14), 4749. <https://doi.org/10.3390/s21144749>
- Diana Andrushia, A., Mary Neebha, T., Trephena Patricia, A., Umadevi, S., Anand, N., & Varshney, A. (2022). Image-based disease classification in grape leaves using convolutional capsule network. *Soft Computing*, 1–14.
- Gawade, A., Deolekar, S., & Patil, V. (2021). Early-stage apple leaf disease prediction using deep learning. *Bioscience Biotechnology Research Communications*, 14(5), 40–43. <https://doi.org/10.21786/bbrc/14.5/8>
- Hernández, S., & López, J. L. (2020). Uncertainty quantification for plant disease detection using Bayesian deep learning. *Applied Soft Computing*, 96, 106597. <https://doi.org/10.1016/j.asoc.2020.106597>
- Javidan, S. M., Banakar, A., Vakilian, K. A., & Ampatzidis, Y. (2023). Diagnosis of grape leaf diseases using automatic K-means clustering and machine learning. *Smart Agricultural Technology*, 3, 100081. <https://doi.org/10.1016/j.atech.2022.100081>
- Jiang, F., Lu, Y., Chen, Y., Cai, D., & Li, G. (2020). Image recognition of four rice leaf diseases based on deep learning and support vector machine. *Computers and Electronics in Agriculture*, 179, 105824. <https://doi.org/10.1016/j.compag.2020.105824>
- Kumar, M., Kumar, A., & Palaparthi, V. S. (2021). Soil sensors-based prediction system for plant diseases using exploratory data analysis and machine learning. *IEEE Sensors Journal*, 21(16), 17455–17468. <https://doi.org/10.1109/JSEN.2020.3046295>
- Latha, R. S. (2021). *Automatic detection of tea leaf diseases using deep convolution neural network* [Paper presentation]. 2021 International Conference on Computer Communication and Informatics (ICCCI). IEEE. <https://doi.org/10.1109/ICCCI50826.2021.9402225>

- Li, K., Lin, J., Liu, J., & Zhao, Y. (2020). Using deep learning for image-based different degrees of ginkgo leaf disease classification. *Information*, 11(2), 95. <https://doi.org/10.3390/info11020095>
- Liu, J., & Wang, X. (2021). Plant diseases and pests detection based on deep learning: A review. *Plant Methods*, 17(1), 22. <https://doi.org/10.1186/s13007-021-00722-9>
- Milke, E. B., Gebremariam, M. T., & Salau, A. O. (2023). Development of a coffee wilt disease identification model using deep learning. *Informatics in Medicine Unlocked*, 42(2023), 101344. <https://doi.org/10.1016/j.imu.2023.101344>
- Nigus, E. A., Taye, G. B., Girmaw, D. W., & Salau, A. O. (2023). Development of a model for detection and grading of stem rust in wheat using deep learning. *Multimedia Tools and Applications*, 83(16), 47649–47676. <https://doi.org/10.1007/s11042-023-17434-y>
- Ouhami, M., Hafiane, A., Es-Saady, Y., El Hajji, M., & Canals, R. (2021). Computer vision, IoT and data fusion for crop disease detection using machine learning: A survey and ongoing research. *Remote Sensing*, 13(13), 2486. <https://doi.org/10.3390/rs13132486>
- Ramkumar, G. (2021). An effectual plant leaf disease detection using deep learning network with IoT strategies. *Annals of the Romanian Society for Cell Biology*, 8876–8885.
- Roy, A. M., & Bhaduri, J. (2021). A deep learning enabled multi-class plant disease detection model based on computer vision. *AI*, 2(3), 413–428. <https://doi.org/10.3390/ai2030026>
- Salau, A. O., Abeje, B. T., Faisal, A. N., & Asfaw, T. T. (2023). *Faba bean disease detection using deep learning techniques* [Paper presentation]. 2023 International Conference on Cyber Management and Engineering (CyMaEn), Bangkok, Thailand (pp. 344–349). <https://doi.org/10.1109/CyMaEn57228.2023.10051088>
- Saleem, M. H., Khanchi, S., Potgieter, J., & Arif, K. M. (2020). Image-based plant disease identification by deep learning meta-architectures. *Plants*, 9(11), 1451. <https://doi.org/10.3390/plants9111451>
- Sharma, R., Singh, A., Kavita, Jhanjhi, N. Z., Masud, M., Sami Jaha, E., & Verma, S. (2022). Plant disease diagnosis and image classification using deep learning. *Computers, Materials & Continua*, 71(2), 2125–2140. <https://doi.org/10.32604/cmc.2022.020017>
- Sujatha, R., Chatterjee, J. M., Jhanjhi, N. Z., & Brohi, S. N. (2021). Performance of deep learning vs machine learning in plant leaf disease detection. *Microprocessors and Microsystems*, 80, 103615. <https://doi.org/10.1016/j.micpro.2020.103615>
- Tan, L., Lu, J., & Jiang, H. (2021). Tomato leaf diseases classification based on leaf images: a comparison between classical machine learning and deep learning methods. *AgriEngineering*, 3(3), 542–558. <https://doi.org/10.3390/agriengineering3030035>
- Upadhyay, S. K., & Kumar, A. (2022). A novel approach for rice plant diseases classification with deep convolutional neural network. *International Journal of Information Technology*, 14(1), 185–199. <https://doi.org/10.1007/s41870-021-00817-5>
- Ushadevi, G. (2020). A survey on plant disease prediction using machine learning and deep learning techniques. *Inteligencia Artificial*, 23(65), 136–154. <https://doi.org/10.4114/intartif.vol23iss65pp136-154>
- Xian, T., Soo, & R., Ngadiran. (2021). Plant diseases classification using machine learning. *Journal of Physics: Conference Series*. 1962(1). 012024.