CHAPTER 1

INTRODUCTION AND OBJECTIVES

1.1 INTRODUCTION

Agriculture is the mother of all civilizations. The focus is on improving productivity without taking into consideration the environmental effects that have appeared in the degeneration of the environment. Plant diseases are very important as this can especially imply both the quality and quantity of plant in the development of agriculture. Generally, the diseases of plants include fungi, bacteria, viruses, moulds, etc. Farmers or specialists typically recognize plants disease and diagnosis them with naked-eyed .

Nevertheless, this approach can be time-consuming, expensive and incorrect, therefore, detection and classification of plants diseases using deep learning technique presents quickly and accurately method. Photographic photos of symptoms of plant infection are utilized for detection of plant disease and for study, teaching and analysis, etc.

Utilizing computer image processing and deep learning technology this is to get a quick and accurate detection; Studies demonstrate that deep learning techniques are effective methods for classification of plant diseases Primary impels have been made to enhance dependability, correctness, and accuracy of image analysis for detecting and classify plant illness .

An automated system aimed at helping diagnose plant diseases by the presence and visible signs of the plant could be of big help to learners in the growing process and also qualified professionals as an infection diagnostic verification system. The researchers used visualization techniques to extract plant disease representation from trained CNN .

There are many researchers have been done every year in the growing part of image processing and computer vision . In this work, we propose a system that detects and classifies plants diseases using a machine learning technique. Our work in deep learning and based on CNN . The data set that was taken from a global data set which is (Plant Village) includes a number of the plants.

Moreover, the rise of affordable mobile devices equipped with cameras has enabled the deployment of these models in portable applications, empowering farmers with tools that can assist in real-time diagnosis directly in the field. When integrated with geographic information systems (GIS), cloud platforms, and expert knowledge bases, such machine learning systems can also contribute to disease outbreak monitoring and decision support for agricultural management.

This study/project aims to develop and evaluate a machine learning-based leaf disease detection system that can classify plant leaf images into specific disease categories. The process involves several stages including image acquisition, preprocessing, model training, validation, and performance evaluation. The end goal is to create a reliable, scalable, and easy-to-use tool that aids in early disease identification, thereby supporting precision agriculture and minimizing crop loss.

1.2 OBJECTIVE

The primary objective of this study is to develop an efficient and accurate machine learning-based system for the detection and classification of plant leaf diseases. By leveraging image processing and deep learning techniques particularly Convolutional Neural Networks (CNNs) the system aims to automatically identify visual symptoms on leaf surfaces such as spots, discoloration, or lesions, which are indicative of various plant diseases. This project seeks to eliminate the limitations of manual disease identification methods, which are often slow, inconsistent, and dependent on expert knowledge. Through training on large datasets of labeled leaf images, the proposed model intends to learn discriminative features that can distinguish between healthy leaves and those affected by specific diseases. Ultimately, the goal is to build a scalable, user-friendly, and real-time diagnostic tool that can assist farmers, agronomists, and agricultural extension workers in making timely decisions, thereby improving crop health management and reducing potential yield losses.

CHAPTER 2

LITERATURE SURVEY

Ranjan, Malvika, and others ”Detection and classification of leaf disease using artificial neural network.” International Journal of Technical Research and Applications 3.3 (2015): 331-333The study of plant leaf disease detection by Malvika Ranjan and colleagues starts with image capturing. Color data, such as HSV features, are retrieved from the segmentation results, and an artificial neural network (ANN) is then trained by selecting feature values that can effectively discriminate between healthy and sick samples. Using a combination of image data processing methods and ann, the current study suggests a method for identifying cotton leaf illnesses early and reliably.

Sladojevic, Srdjan ,and others. ”Deep neural networks-based recognition of plant diseases by leaf image classification.” Computational intelligence and neuroscience 2016 (2016) . Srdjan Sladojevic and colleagues present Deep Convolutional Neural Network Supported Identification of Crop Diseases by Plant Image Classification, a new method for the construction of a crop diseases recognition model based on plant image classification and deep convolutional networks. The methodology employed and the novel technique of training allow for a quick and painless system set up in practice. With the ability to identify crops from their surroundings, the built model can recognize thirteen types of plant illnesses from healthy leaves. All of the necessary processes for applying this diseases recognition model are detailed throughout the study, beginning with the collection of photographs in order to establish a database that is evaluated by agricultural experts. Caffe, a deep learning framework developed by Berkley Vision and Learning Centre, was used to perform the deep CNN training. The experimental results on the developed model achieved precision between 91% and 98%, for separate class tests, on average 96.3%.

Cortes, Emanuel. ”Plant disease classification using convolutional networks and generative adversarial networks.” (2017). CNN and Modeling Adversarial Networks were used to classify plant diseases. Others, like Emanuel Cortes A deep neural network and semi-supervised algorithms were trained to distinguish crop species and disease status of 57 different classes using a publicly available dataset of 86,147 photos of ill and healthy plants. rs-net was the unlabeled data experiment that functioned successfully. With a detection rate of 1e-5, it was able to score more than 80% in the training phase in less than 5 epochs.

Wallelign, Serawork, Mihai Polceanu, and Cedric Buche. ”Soybean plant disease identification using convolutional neural network.” The thirty-first international flairs conference. 2018Plant disease identification and treatment using neural network models, Konstantinos P. Ferentinos and colleagues-built CNN models to conduct crop disease identification and diagnosis using basic leaf pictures of healthy and sick plants. The models were trained using an open collection of 87,848 photos, which included 25 kinds of plants in 58 various classes of [plant, illness] pairs, including non-affected plants . Multiple model architectures were developed, with the top performing one achieving a success rate of 99.53 percent. The model's high success rate makes it a valuable or early detection tool.

Dhakal, Ashwin, and Subarna Shakya. ”Image-based plant disease detection with deep learning.” International Journal of Computer Trends and Technology 61.1 (2018): 26-29. Prasanna Mohanty and colleagues developed a deep convolutional neural network using deep learning to detect 14 different crops and 26 illnesses. On a held-out test set, the training set model obtained an accuracy of 99.35 percent, illustrating the practicality of this strategy. The model still obtains a 31.4 percent accuracy when tested on a collection of photographs acquired from reputable web sources - i.e. images shot under settings distinct from those used for training. While this accuracy is substantially greater than the one based on random selection 2.6%, a larger collection of training data is required to increase overall accuracy.

A. Venkataramanan, D.K.P. Honakeri, P. Agarwal, Plant disease detection and classification using deep neural networks, Int. J. Comput. Sci. Eng. 11 (9) (2019) 40–46. The documentation of the Tomato Plant leaf infection by image processing method is proposed in this paper ,which includes clustering, an open-source algorithm, and Image segmentation. Here CNN algorithm is used for the extraction of hierarchical features, which maps input image pixel concentrations and compares with the training set of images. It utilizes fuzzy logic, hybrid algorithms, and an Artificial Neural Network could also be organized. GLCM (Gray Level Co-occurrence Matrix) has been implemented to categorize and segregate the leaf image depending on various phases. Real-time submission centered on disease identification will be the main factor in the selection of techniques. However, the method consumes more time during the training process.

CHAPTER 3
LEAF DISEASE

3.1 DEFINITION

Leaf disease refers to any pathological condition that affects the structure, color, texture, or function of a plant's leaves due to the presence of harmful organisms such as fungi, bacteria, viruses, or pests. These diseases commonly manifest as visible symptoms on the leaf surface, including spots, blights, wilting, yellowing (chlorosis), curling, or necrosis (dead tissue). Leaf diseases can significantly impair photosynthesis and overall plant health, leading to reduced growth, lower crop yield, and in severe cases, plant death. Timely detection and identification are critical for effective management and prevention of widespread agricultural losses.

- Fungal Leaf Diseases
- Bacterial Leaf Diseases
- Viral Leaf Diseases
- Environmental and Physiological Disorders

3.1.1 Fungal Leaf Diseases

Numerous fungi that like warm, humid conditions are the source of fungal leaf diseases, which are prevalent in plants. Spots, mold, or mildew on the leaves are common indications of these diseases, which can lower photosynthesis, weaken the plant, and in extreme situations, cause leaf drop or plant death. Powdery mildew, downy mildew, early and late blight, and rust are a few examples of fungal leaf diseases, each having a unique set of symptoms and treatment methods.

Figure 3.1 Septoria Leaf Spot of Tomato

3.1.2 Bacterial Leaf Diseases

Plant diseases called bacterial leaf diseases are brought on by bacteria and frequently manifest as wilting, blights, or patches on the leaves. These illnesses can seriously harm and reduce the productivity of a variety of plants, including ornamental and agricultural ones.

Figure 3.2 Bacterial Black Spot in Mango

3.1.3 Viral Leaf Diseases

Viral leaf diseases in plants manifest as various symptoms, including mosaic patterns, leaf curling, vein clearing, and necrosis (tissue death). These diseases are caused by viruses that infect plant cells, disrupting their normal growth and development.

Figure 3.3 Tomato Yellow Leaf Curl Virus

CHAPTER 4

CONVOLUTIONAL NEURAL NETWORK (CNN)

Because Convolutional Neural Networks (CNNs) can automatically extract information from photos and properly identify them, they are frequently employed in plant disease diagnosis. Large datasets of plant leaf photos, like the Plant Village dataset, which includes tagged photos of both healthy and diseased leaves, are used in this context to train CNNs. These images are processed by the CNN using a number of layers, such as pooling layers that lower dimensionality while maintaining crucial features and convolutional layers that identify patterns like edges, textures, and colors. CNN learns more intricate traits that aid with illness differentiation as the image moves through deeper layers**.**

The system can then determine if a leaf is healthy or diseased, as well as the sort of disease it has, thanks to fully linked layers and a SoftMax classifier that output the probability for each disease class. CNNs are particularly useful for image-based tasks like plant disease identification because they do not require manual feature extraction. They are appropriate for large-scale agricultural monitoring applications as well as real-time detection systems because to their excellent accuracy and versatility.

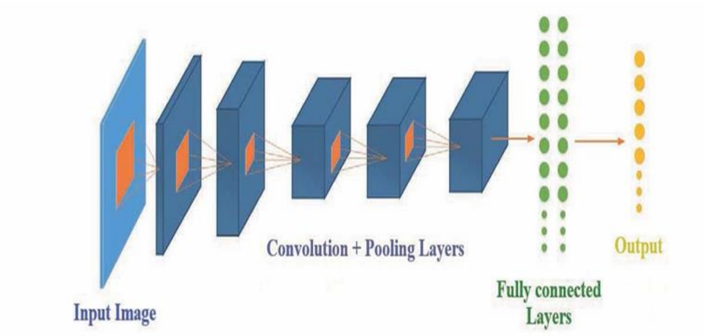

Figure 4.1 . Illustration of Convolutional Neural Network Architecture

4.1 CNN ARCHITECTURE IN LEAF DISEASE DETECTION

Convolutional Neural Networks (CNNs) are a specialized class of deep learning models designed to process data that comes in the form of arrays, such as images. In the context of leaf disease detection, CNNs serve as the core model that learns to recognize disease-specific features from leaf images. The architecture typically includes several critical components: convolutional layers, activation functions, pooling layers, fully connected layers, and an output layer.

- Input Layer
- Convolutional Layers
- Activation Function (ReLU)
- Pooling Layers
- Fully Connected Layers
- Output Layer
- Model Training and Optimization

4.1.1 Input Layer

The CNN architecture begins with an input layer that receives images of diseased or healthy leaves. These images are usually resized to fixed dimensions (e.g., 128×128 or 224×224 pixels) and normalized for consistency. This layer ensures the model receives a uniform data structure, facilitating faster and more stable training.

4.1.2 Convolutional Layers

The convolutional layers are the heart of a CNN. They apply a series of learnable filters (kernels) that slide across the image and extract low-level features such as edges, textures, or color gradients. Deeper layers capture more complex features like disease spots, vein distortions, or lesion shapes, which are critical for distinguishing between different plant diseases. Each filter helps detect a specific pattern, and stacking multiple convolutional layers allows the model to learn hierarchical representations of the image data.

4.1.3 Activation Function (ReLU)

After each convolutional operation, a non-linear activation function such as the Rectified Linear Unit (ReLU) is applied. ReLU replaces negative pixel values with zero, introducing non-linearity into the model. This enables the network to learn more complex and abstract features, improving its ability to capture subtle variations between healthy and diseased leaves.

4.1.4 Pooling Layers

To reduce the spatial dimensions of the feature maps and computational cost, pooling layers typically Max Pooling are introduced after some convolutional layers. Pooling summarizes the presence of features in patches of the feature map, helping the model become invariant to small translations or distortions in the image. This is especially useful for handling different leaf orientations and lighting conditions in real-world scenarios.

4.1.5 Fully Connected Layers

Once the feature maps are sufficiently reduced, they are flattened and passed into one or more fully connected layers. These layers act as a traditional neural network, where all neurons are connected to every neuron in the next layer. This stage is responsible for combining all extracted features and making final predictions. The fully connected layers serve as a decision-making part of the network based on the learned visual features

4.1.6 Output Layer

The final layer in the CNN is the output layer, typically using a softmax activation function for multi-class classification. It assigns probabilities to each possible disease class (or a healthy class), and the class with the highest probability is selected as the model's prediction. For binary classification (e.g., diseased vs. healthy), a sigmoid activation may be used instead.

4.1.7 Model Training and Optimization

The model is trained using a labeled dataset of leaf images, where the true class (disease type or healthy) is known. Optimization algorithms like Adam are used to minimize the loss function—usually categorical cross-entropy for multi-class problems. During training, the model iteratively adjusts its weights based on the error between predicted and actual classes

CHAPTER 5
METHODOLOGY

Diseases in Plants are a major concern to the farmers these days. Many a times, the farmers are not sure which pesticide or insecticide is needed to treat a particular diseased plant because they are not sure of the type of disease. This results in spraying wrong pesticides, damaging the plants which affect the plant yield. To overcome with this problem, we have come up with a solution of developing a system that easily identifies some common diseases that occur in the plants.

Through a machine learning-based plant disease detection system using Convolutional Neural Networks (CNNs) to enable accurate and automated identification of plant diseases from leaf images. The approach involves collecting a diverse dataset of healthy and diseased plant leaves from sources such as Plant Village and real-world agricultural fields.

Preprocessing techniques, including image resizing, normalization, and data augmentation, will be applied to enhance model performance The model will be optimized using hyperparameter tuning and evaluated using metrics such as accuracy, precision, recall, and F1-score.

This not only beneficial to the farmers in saving the crops, but also in saving money by buying only right kind of pesticides suitable to treat the particular disease. This aims to develop a reliable, scalable, and efficient tool for early disease detection, ultimately aiding farmers and researchers in improving agricultural productivity.

As the system does not involve any heavy machineries and electricity, the system proves to not only be a cost-effective solution, but also an environment-friendly one.

Once the model achieves satisfactory results, it is tested on unseen data to assess its generalization ability. Finally, the trained model is integrated into a user-friendly platform—such as a mobile or web application—allowing users to upload leaf images and receive real-time disease diagnosis. This end-to-end methodology ensures the development of a scalable, accurate, and practical solution for automated leaf disease detection, ultimately supporting precision agriculture and timely crop health management

CHAPTER 6

BLOCK DIAGRAM AND DESCRIPTION OF THE BLOCK DIAGRAM

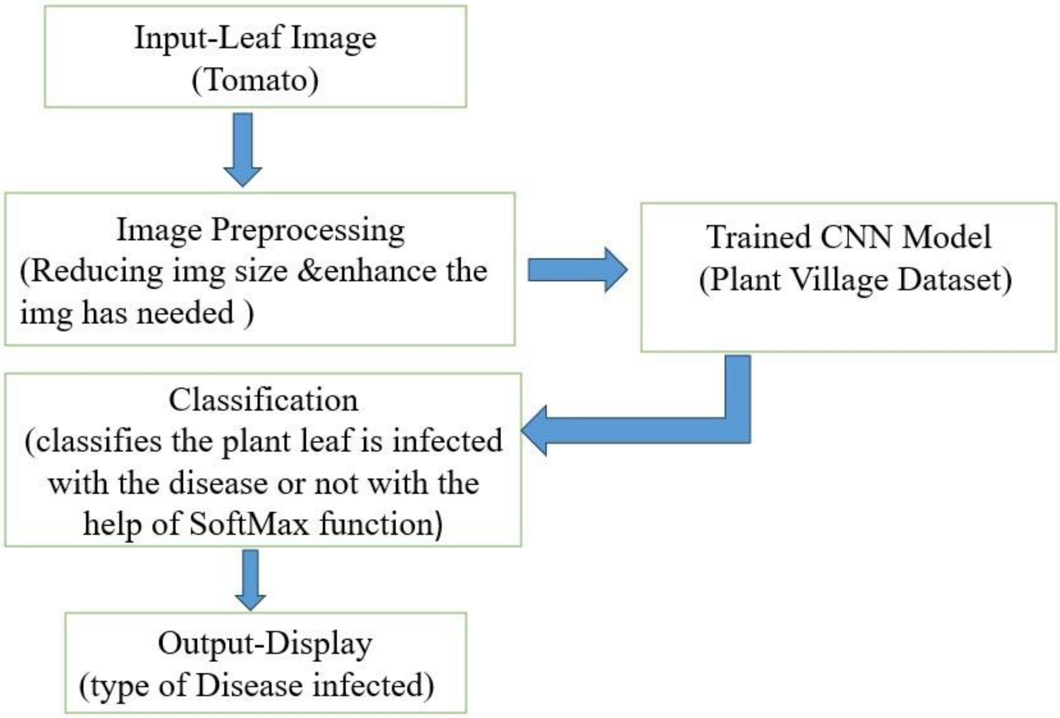

Figure 6.1 Block diagram

6.1 DESCRIPTION OF THE BLOCK DIAGRAM

6.1.1 Input - Leaf Image

The procedure begins with the input of an image of a leaf. This picture was taken with a camera, cell phone, or other imaging equipment. The objective is to determine if the leaf is disease-free or not. This stage is essential since the input image's quality has a big impact on the classification's accuracy.

A clean image for processing depends on a number of factors, including lighting, angle, and resolution.

6.1.2 Image Preprocessing

To make sure it is appropriate for analysis, the image is pre-processed once it is taken. Several processes, including scaling, noise reduction, contrast enhancement, and color normalization, are part of image preprocessing. In order to speed up processing while preserving important information, the image size is frequently decreased. Techniques like edge detection and segmentation may also be used in this step to draw attention to crucial leaf regions. Improving image quality and standardizing the format for subsequent stages are the main objectives of preprocessing.

6.1.3 Plant Village Dataset

The captioned photos of plant leaves in the Plant Village Dataset include both healthy and sick specimens. Machine learning algorithms for plant disease detection are frequently trained using this dataset. The collection includes hundreds of photos of both healthy leaves and various disease classes. These pictures aid in the CNN model's learning of characteristics and patterns linked to different plant diseases. The model wouldn't have a reference to tell healthy leaves from diseased ones without this dataset.

6.1.4 Trained Model (CNN)

A Convolutional Neural Network (CNN) is applied to train the model using the dataset. CNN is a deep learning system that excels in classifying and recognizing images. It is made up of several layers that extract characteristics from images, including convolutional, pooling, and fully connected layers. The model gains the ability to identify patterns in leaf photos during training, including color shifts, texture differences, and shape deformations that point to the presence of disease. To increase accuracy, the CNN gives various features weights and refines them over several rounds. The model is prepared to classify new photos after it has received enough training.

6.1.5 Classification

At this point, the system uses a classification algorithm to determine whether the plant leaf is healthy or sick. Each disease category is given a probability score by the SoftMax function used in the model. The SoftMax function aids in multi-class categorization by identifying whether a leaf is healthy or falls into one of several illness categories. When deciding on the best course of action, such suggesting pesticides or other treatments, this classification stage is essential.

6.1.6 Output – Display

Lastly, the system shows the categorization result, which includes the precise disease category and whether the leaf is healthy or sick. A computer screen, a smartphone app, or an agricultural monitoring system can all show the output. With this knowledge, the farmer or agricultural specialist can take the appropriate steps to stop the spread of illness, such using the right pesticide, getting rid of diseased plants, or changing the surroundings.

This automated method uses deep learning techniques to greatly increase the efficiency of plant disease diagnosis. The system can correctly identify plant illnesses from leaf photos by using a CNN model that was trained on the Plant Village Dataset. From image preprocessing to classification and result display, the entire process guarantees prompt disease identification, assisting farmers in taking preventative action and enhancing agricultural output. This method minimizes the chance of crop loss from diseases that go undiscovered while also cutting down on the amount of time spent on human examination

CHAPTER 7
HARDWARE DESCRIPTION

7.1 REQUIREMENTS

- Personal Computer

7.1.1 Personal Computer

A PC provides the basic computing environment for all the tasks involved in the project.

- Running the operating system (Windows).
- Executing the Python code.
- Storing the data and code.
- Displaying the results.

The PC should we have

- A decent CPU. (AMD RYZEN 5)
- Sufficient RAM. (16 GB)
- Adequate storage. (512 GB/ SSD)
- And most importantly, a dedicated GPU for efficient training. (AMDA RADEON GRAPHICS)

CHAPTER 8

SOFTWARE DESCRIPTION

8.1 REQUIREMENTS

- Python
- Kaggle
- Google Colab
- Google Drive

8.1.1 Python

Python is the fundamental programming language for deep learning-based plant disease detection because of its ease of use, adaptability, and robust library ecosystem. NumPy is utilized in this environment for numerical tasks like preprocessing picture pixel data into arrays that can be input into neural networks and manipulating matrices. Pandas, which provides user-friendly data structures like Data Frames, is essential for managing and evaluating structured data, including illness labels, metadata, and model performance indicators. Seaborn and Matplotlib are essential tools for visualizing results: While Seaborn offers more sophisticated, visually appealing statistical displays like heatmaps and class distribution charts, Matplotlib is frequently used to plot loss and accuracy curves over training epochs. When combined, these modules make Python the perfect language for creating and evaluating plant disease detection models by streamlining the process from data preprocessing and model evaluation to result interpretation.

Figure 8.1 (a) Python

8.1.2 Kaggle

By offering a collaborative platform where researchers, data scientists, and developers can access high-quality datasets, share code, and compete, Kaggle plays a vital role in the detection of plant diseases. The well-known PlantVillage dataset, which includes thousands of tagged leaf photos of both healthy and diseased plants, is one of the datasets available on Kaggle for plant disease diagnosis. For image classification tasks, users can use these datasets to train deep learning or machine learning models, especially convolutional neural networks (CNNs). Without installing anything locally, users can write and run Python code using libraries like numpy pandas seaborn matplotlib right in the browser. Python is the perfect language for creating and evaluating plant disease detection models because of these packages, which together simplify the process from data preprocessing and model evaluation to result interpretation.

8.1.3 Google Colab

Often used in plant disease detection projects, Google Colab is a free cloud-based platform that offers a ready-to-use environment for creating and executing deep learning models. It supports Python and popular libraries that are necessary for creating and training plant disease detection models using image data. Users can simply upload datasets (like Plant Village) or access them directly from Google Drive, preprocess the images, and train convolutional neural networks (CNNs) without the need for expensive local hardware because it has built-in access to free GPUs and TPUs, which greatly speeds up training, especially for large datasets. Additionally, Colab makes it simple to visualize results, plot accuracy/loss graphs, and even displays classified images to analyze model performance. Its collaborative features also make it ideal for sharing notebooks with teammates or publishing research work efficiently.

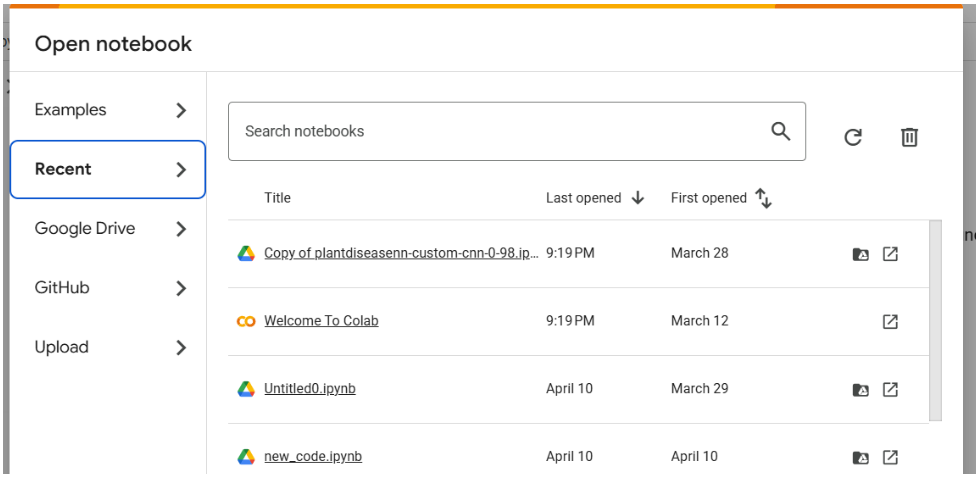

Figure 8.1 (b) Google Colab

8.1.4 Google Drive

Google Drive is commonly used in plant disease detection projects as a reliable cloud storage solution for managing datasets, models, and project files. When working on platforms like Google Colab, users often mount Google Drive to access large image datasets (such as leaf images from Plant Village) directly from the cloud, eliminating the need to upload files repeatedly. This integration enables for easy reading, writing, and storing of data, including preprocessed images, trained model checkpoints, and result logs. Researchers and developers can also use Google Drive to store and manage several versions of trained models, making it easy to compare performance or resume training. Additionally, it provides as a handy method to communicate datasets and code notebooks with collaborators, enabling successful teamwork in plant disease detection project.

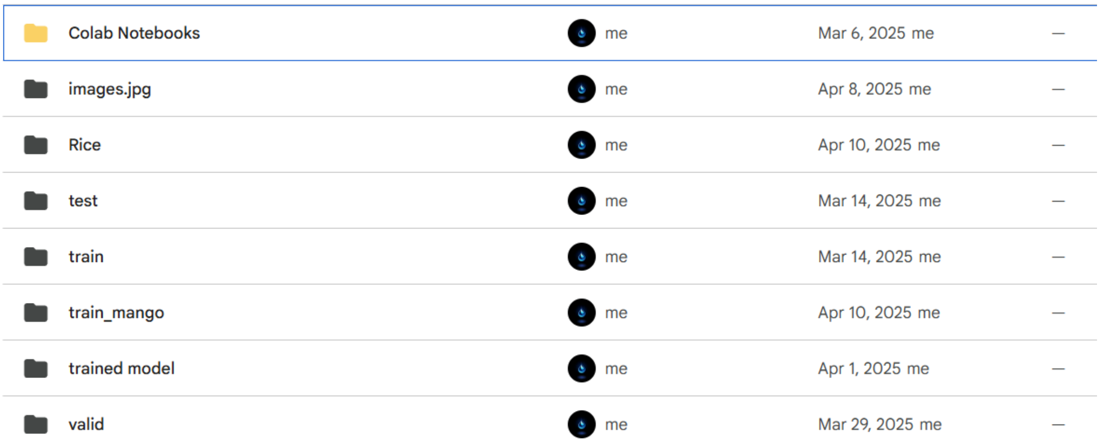

Figure 8.1 (c) Google Drive With Dataset

8.2 LIBRARY PACKAGES

- NumPy
- Pandas
- Matplotlib
- PyTorch
- Seaborn

8.2.1 Numpy

NumPy is utilized to handle numerical calculations in plant disease detection, particularly when preprocessing images and preparing model input. Pixel values are frequently transformed into Numpy arrays for manipulation, normalization, and reshaping when working with picture data. Numpy arrays are crucial for effective computation and matrix operations in deep learning workflows since they are also utilized to store batch data during training and inference.

Figure 8.2 (a) Numpy

8.2.2 Pandas

The main purpose of pandas is to manage structured data, including performance metrics, training logs, image labels, and classification results. Pandas DataFrames facilitate the loading of CSV files with disease categories, data distribution analysis, sample filtering, and dataset preparation for model training and assessment in plant disease detection. For accuracy analysis, it also aids in combining prediction results with ground truth.

Figure 8.2 (b) Pandas

8.2.3 Matplotlib

Matplotlib is used to create custom plots, including comparison charts, sample image visualizations, and accuracy and loss graphs over training epochs. It aids in the visual interpretation of results, such as highlighting sick spots on leaf photos, and tracks model performance throughout training in plant disease identification. When used in conjunction with Seaborn, it provides precise control over visualization.

Figure 8.2 (c) Matplotlib

8.2.4 Pytorch

Convolutional neural networks (CNNs) are constructed, trained, and assessed for plant disease detection using PyTorch, the fundamental deep learning package. It offers resources for defining training loops, optimizers, loss functions, and model design. PyTorch is effective at processing big image collections like PlantVillage since it allows GPU acceleration. In order to prepare and enhance leaf image datasets for training reliable models, it also provides data loaders and transformation tools.

Figure 8.2 (d) PyTroch

8.2.5 Seaborn

High-level, aesthetically pleasing statistical plots that aid in evaluating the effectiveness of plant disease detection models are produced using Seaborn. For instance, it can provide distribution plots to analyze prediction confidence, bar charts to compare accuracy by class, and heatmaps to illustrate confusion matrices. Understanding how successfully the model differentiates between various plant diseases depends on these representations.

Figure 8.2 (e) Seaborn

CHAPTER 9

RESULTS AND DISCUSSION

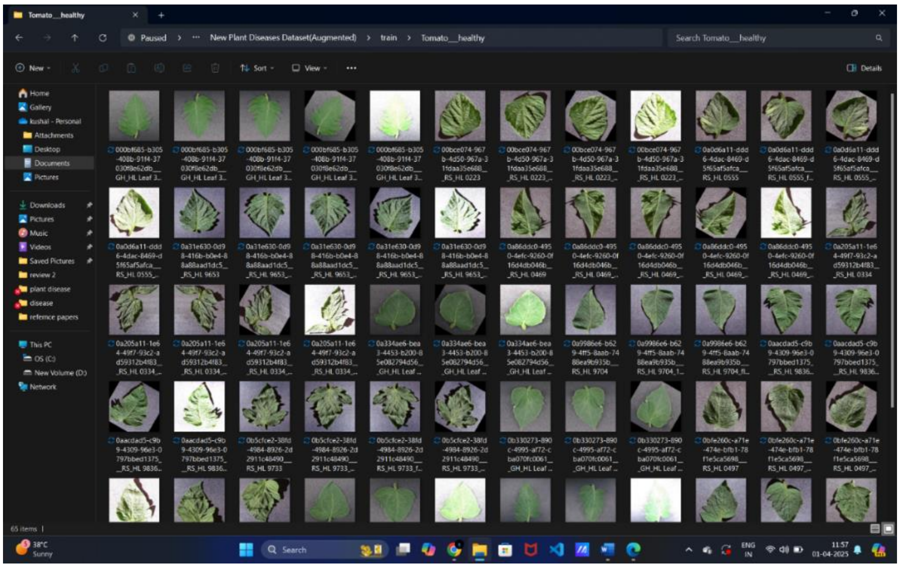

Figure 9.1(a) Data Collection

The Fig 9.1(a) shows images are given in the folder with various types of leaf samples for training and testing.

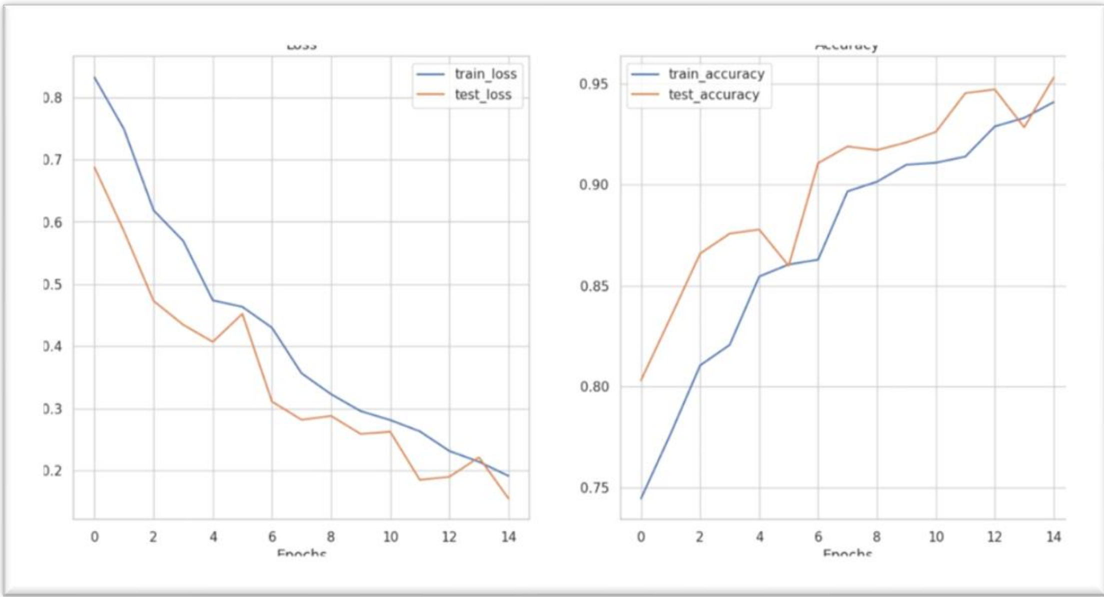

Figure 9.1 (b): Graph

The Fig 9.1 (b) graph shows the images trained for 15 epoch. It shows the 1.Epochs Vs Loss And 2. Epochs Vs Accuracy

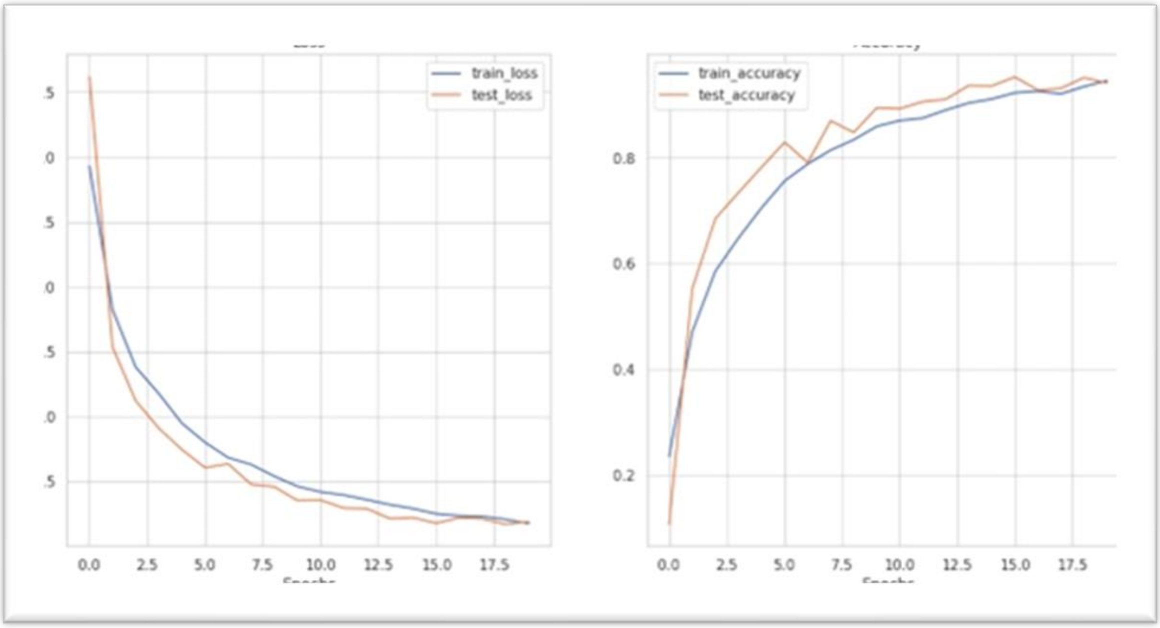

Figure 9.1(c): Graph

The above Fig 9.1(c) graph shows the images trained for 20 epoch. It shows the 1.Epochs Vs Loss And 2. Epochs Vs Accuracy

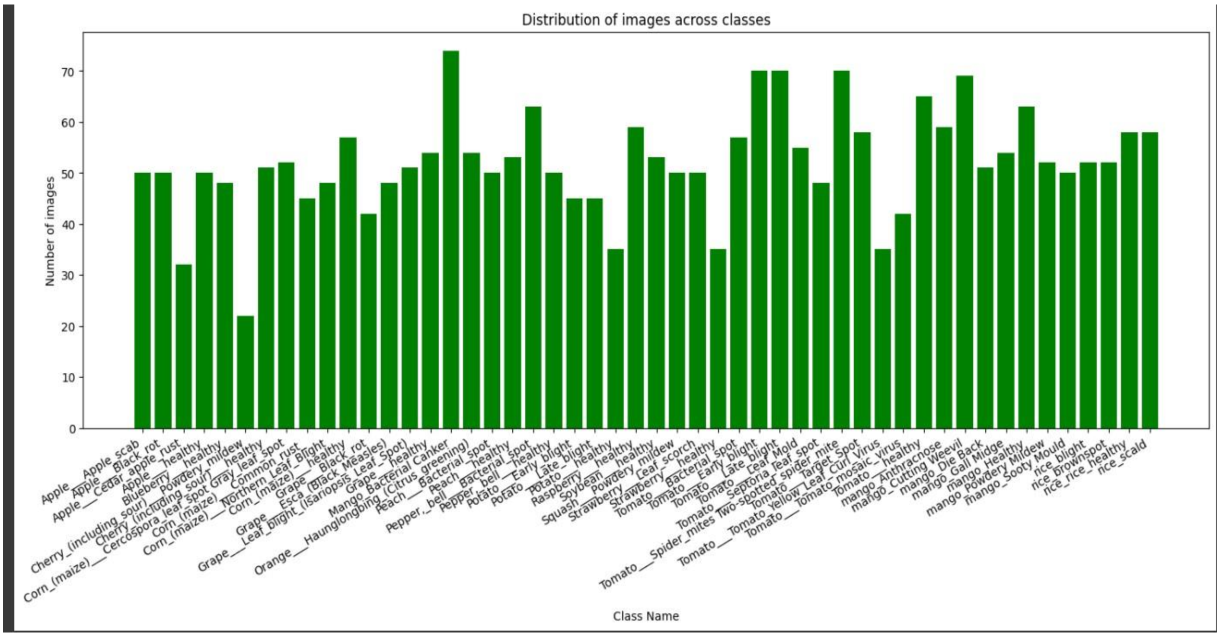

Figure 9.1 (d) Bar Graph

The above Fig 9.1(d) Bar Graph represents the number of images Vs class name

```
Total images found: 48
```

Figure 9.1 (e) Images in Test Folder

The Fig 9.1(e) shows the number of images with various types of leaf given for testing. It consists of healthy and diseased leaves images.

```
==========================================================================================
Layer (type:depth-idx)                   Output Shape              Param #
==========================================================================================
PlantDiseaseNN                           [1, 56]                   --
├─Sequential: 1-1                        [1, 32, 128, 128]         --
│    └─Conv2d: 2-1                       [1, 32, 256, 256]         896
│    └─BatchNorm2d: 2-2                  [1, 32, 256, 256]         64
│    └─ReLU: 2-3                         [1, 32, 256, 256]         --
│    └─MaxPool2d: 2-4                    [1, 32, 128, 128]         --
├─Sequential: 1-2                        [1, 64, 64, 64]           --
│    └─Conv2d: 2-5                       [1, 64, 128, 128]         18,496
│    └─BatchNorm2d: 2-6                  [1, 64, 128, 128]         128
│    └─ReLU: 2-7                         [1, 64, 128, 128]         --
│    └─MaxPool2d: 2-8                    [1, 64, 64, 64]           --
├─Sequential: 1-3                        [1, 128, 32, 32]          --
│    └─Conv2d: 2-9                       [1, 128, 64, 64]          73,856
│    └─BatchNorm2d: 2-10                 [1, 128, 64, 64]          256
│    └─ReLU: 2-11                        [1, 128, 64, 64]          --
│    └─MaxPool2d: 2-12                   [1, 128, 32, 32]          --
├─Sequential: 1-4                        [1, 256, 16, 16]          --
│    └─Conv2d: 2-13                      [1, 256, 32, 32]          295,168
│    └─BatchNorm2d: 2-14                 [1, 256, 32, 32]          512
│    └─ReLU: 2-15                        [1, 256, 32, 32]          --
│    └─MaxPool2d: 2-16                   [1, 256, 16, 16]          --
├─Sequential: 1-5                        [1, 256, 8, 8]            --
│    └─Conv2d: 2-17                      [1, 256, 16, 16]          590,080
│    └─BatchNorm2d: 2-18                 [1, 256, 16, 16]          512
│    └─ReLU: 2-19                        [1, 256, 16, 16]          --
│    └─MaxPool2d: 2-20                   [1, 256, 8, 8]            --
├─Sequential: 1-6                        [1, 56]                   --
│    └─Flatten: 2-21                     [1, 16384]                --
│    └─Linear: 2-22                      [1, 1024]                 16,778,240
│    └─ReLU: 2-23                        [1, 1024]                 --
│    └─Dropout: 2-24                     [1, 1024]                 --
│    └─Linear: 2-25                      [1, 56]                   57,400
==========================================================================================
Total params: 17,815,608
Trainable params: 17,815,608
Non-trainable params: 0
Total mult-adds (Units.GIGABYTES): 1.13
==========================================================================================
Input size (MB): 0.79
Forward/backward pass size (MB): 63.97
Params size (MB): 71.26
Estimated Total Size (MB): 136.02
==========================================================================================
```

Figure 9.1(f) Model Summary

The Fig 9.1(f) shows a detailed summary of a custom convolutional neural network (CNN) model named leafDisease, used likely for leaf disease classification. This summary is generated using PyTorch and tools like torchsummary or torchinfo. The Model Structures has the following layers.

- Convolutional layer
- Batch Normalization
- Activation Layers (ReLU)
- Pooling Layers
- Flatten Layer
- Fully Connected (Linear) Layers

Model Parameters

- Total trainable parameters: 17,815,608
- No non-trainable parameters (i.e., no frozen layers)
- Most parameters are in the fully connected layer Linear: 2-22 with over 16 million params

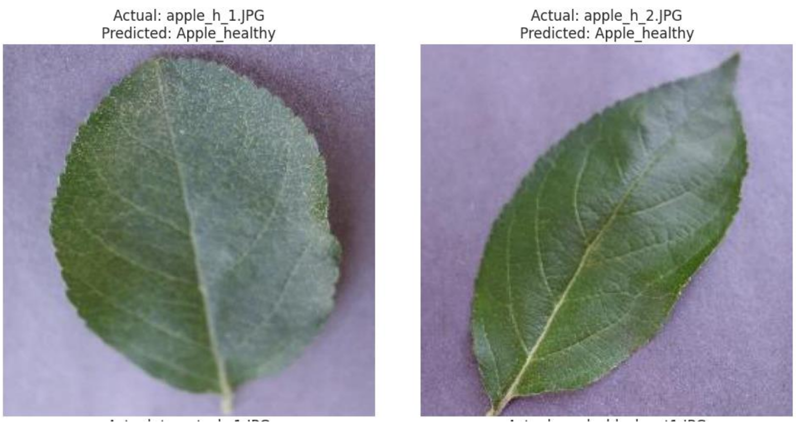

Figure 9.1(g) Output

The Fig 9.1(g) image above represents the output of the model.

CHAPTER 10

APPLICATIONS AND ADVANTAGES

10.1 APPLICATIONS

- **Precision Agriculture**

  Improves crop yield and productivity. And reduces excessive pesticide use, minimizing environmental impact.

- **Integration with IoT-Based Smart Farming System**

  CNN models can be integrated with Internet of Things (IoT) devices such as smart cameras and sensors that continuously monitor plant health. These systems automatically alert farmers when early disease symptoms are detected. Enables real-time disease monitoring and reduces the need for frequent manual inspections.

- **Agricultural Research and Development**

  CNN models assist scientists in studying the patterns and evolution of plant diseases. This data helps in developing disease-resistant crop varieties through genetic modification and selective breeding. Supports agricultural innovation and research. And helps in developing AI-driven farming policies

10.2 ADVANTAGES

- Automatic Feature Extraction: No need for manual feature extraction
- High Accuracy: Higher Accuracy than traditional ML model
- Scalability: Classify multiple plant diseases efficiently

CHAPTER 11

CONCLUSION AND FUTURE IMPROVEMENT

11.1 CONCLUSION

Considering the significance of plants and agriculture both domestically and globally. Due to the numerous plant diseases that currently exist, This study suggested a strong approach based on computer facilities and Deep Learning Techniques to accurately and quickly detect and classify these diseases. The CNN algorithm was used in this work to produce the results. and thus, made it possible to quickly and accurately identify the disease type as well as the kind of plant with the infection by looking at its leaves

11.2 FUTURE SCOPE & IMPROVEMENTS

- **Data Augmentation & Synthetic Data**:

  Generating synthetic images using GANs (Generative Adversarial Networks) can improve model generalization.

- **Hybrid Models**:

  Combining CNNs with transformers (Vision Transformers - ViTs) for better feature extraction.

- **Edge AI & IoT**:

  Deploying disease detection models on IoT devices, drones, and mobile applications for real-time monitoring. And Multi-Spectral & Hyperspectral Imaging: Instead of RGB images, using multi-spectral images for higher accuracy.

REFERENCES

1. Bashir, S., & Sharma, N. (2012). *Remote area plant disease detection using image processing*. IOSR Journal of Electronics and Communication Engineering, 1(6), 31-34.

2. Bashish, D., M. Braik, and S. Bani-Ahmad. (2011). *Detection and classification of leaf diseases using K-means-based segmentation and neural networks-based classification*. Information Technology Journal, 10(2): 267-275.

3. Baranwal, S., S. Khandelwal, A. Arora. (2019). *Deep learning convolutional neural network for apple leaves disease detection*, SSRN Electron. J.

4. Cortes, Emanuel. (2017). *Plant disease classification using convolutional networks and generative adversarial networks*.

5. Fuentes, Alvaro, and others. (2017). *A robust deep-learning based detector for real-time tomato plant diseases and pests' recognition*. Sensors, 17(9): 2022.

6. Ishak, Syafiqah, and others. (2015). *Leaf disease classification using artificial neural network*. Jurnal Teknologi, 77.17.

7. Jasim, M. A., & AL-Tuwaijari, J. M. (2020, April). *Plant leaf diseases detection and classification using image processing and deep learning techniques*. In 2020 International Conference on Computer Science and Software Engineering (CSASE), pp. 259-265.

8. Kaur, L., & Laxmi, V. (2016). *Detection of Unhealthy Region of plant leaves using Neural Network*. Disease Management, 1(05), 34-42.

9. Kumar R, Chug A, Singh AP, Singh D. (2022). *A systematic analysis of machine learning and deep learning-based approaches for plant leaf disease classification: a Review*. J Sensors.

10. Liu, Peide, Tahir Mahmood, and Qaisar Khan. (2017). *Multi attribute decision making based on prioritized aggregation operator under hesitant intuitionistic fuzzy linguistic environment*. Symmetry, 9(11): 270.

11. Mohanty, Sharada P., David P. Hughes, and Marcel Salathé. *Using deep learning for image-based plant disease detection*. Frontiers in Plant Science, 7.

12. Muthukannan, K., P. Latha, R. Pon Selvi and P. Nisha. (2015). *Diseased plant leaves using Neural Network Algorithms*. ARPN Journal of Engineering and Applied Sciences, VOL. 10, NO. 4, MARCH 2015, ISSN 1819-6608.

13. Ranjan, Malvika, and others. (2015). *Detection and classification of leaf disease using artificial neural network*. International Journal of Technical Research and Applications, 3.3: 331-333.

14. Sabah Bashir, Navdeep Sharma. (2012). *Remote Area Plant Disease Detection Using Image Processing*. IOSR Journal of Electronics and Communication Engineering. ISSN : 2278-2834 Volume 2, Issue 6, PP 31-34.

15. Saleem, MH, Potgieter J, Arif KM. (2020). *Plant disease classification: a comparative evaluation of convolutional neural networks and deep learning optimizers*. Plants.

16. Sladojevic, Srdjan, and others. (2016). *Deep neural networks-based recognition of plant diseases by leaf image classification*. Computational Intelligence and Neuroscience.

17. Wallelign, Serawork, Mihai Polceanu, and Cedric Buche. (2018). *Soybean plant disease identification using convolutional neural network*. The Thirty-First International FLAIRS Conference.