A Project Report On

Plant Leaf Disease Detection Using Deep Learning

Submitted by

P.DHARANIPATHI(R170584)

Submitted to

Under the Supervision of

Mr.P.Harinadha
M.Tech(University of Hyderabad**),**
Assistant Professor,
Department of Computer Science and Engineering

As a part of

Partial fulfillment of the degree of Bachelor of Technology in
Computer Science and Engineering

CERTIFICATE OF EXAMINER

This is to certify that the report entitled **Plant Leaf Disease Detection Using Deep Learning** submitted by **P.DHARANIPATHI (R170584)**in partial fulfillment of the requirement for the award of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out by her under my supervision and guidance.

I am ineffably indebted to **Mr.P.Harinadha**, my project internal guide for conscientious guidance and encouragement to accomplish this project.

I am extremely thankful and pay my gratitude to **Mr.N.Satyanandaram, HOD CSE,** for his valuable guidance and support on the completion of this project.

Project Guide,
Mr.P.Harinadha,
CSE,
RGUKT, RK-Valley.

Head of Department,
Mr.N.Satyanandaram,
CSE,
RGUKT,RK-Valley.

RGUKT
R.K.Valley

CERTIFICATION OF PROJECT COMPLETION

This is to certify that the report entitled **Plant Leaf Disease Detection Using Deep Learning** submitted by **P.DHARANIPATHI (R170584)** in partial fulfillment of the requirement for the award of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out by her under my supervision and guidance.

Project Guide,
Mr.P.Harinadha,
CSE,
RGUKT, RK-Valley.

Head of Department,
Mr.N.Satyanandaram,
CSE,
RGUKT,RK-Valley.

Acknowledgement

I would like to express my sincere gratitude to **Mr.P.Harinadha** , my project internal guide for valuable suggestions and keen interest throughout the progress of my course of research.

I would also like to extend my gratitude to **N.Satyanandaram, HOD of CSE**, for providing extraordinary support in this process.

At the outset, I would like to thank **Rajiv Gandhi University of Knowledge Technologies, R.K Valley** for providing all the necessary resources for the successful completion of my course work.

With Sincere Regards,
P.Dharanipathi(R170584).

Declaration

I **P.Dharanipathi** here declare that this report entitled **"Plant Leaf Disease Detection Using Deep Learning"** submitted by me under the guidance and supervision of **Mr.P.Harinadha**, is a bonafide work. I also declare that it has not been submitted previously in part or in full to this University or Institution for the award of any degree.

Date: 3rd May,2023,
Place: RK-Valley.

P.Dharanipathi,
ID No: R170584.

INDEX

Abstract

Plant leaf disease detection is a critical task for the agriculture industry, as plant diseases can cause significant economic and environmental damage. Early detection and accurate diagnosis of plant diseases can help prevent the spread of diseases and reduce crop losses. In recent years, deep learning techniques, particularly convolutional neural networks (CNNs), have shown promising results in image recognition and classification tasks. In this project, we propose a plant leaf disease detection system using deep learning techniques, specifically a CNN architecture.

The proposed system is trained on a dataset of plant leaf images consisting of both healthy and diseased leaves, and can detect various types of plant leaf diseases, including bacterial and fungal infections. Our experimental results show that the proposed system can detect plant leaf diseases with high accuracy, making it a valuable tool for the agriculture industry.

Overall, the proposed plant leaf disease detection system using deep learning techniques is a promising solution to the problem of plant disease detection in agriculture. The proposed system has the potential to reduce the negative impact of plant diseases on the environment and the economy and can help improve crop yields.

Introduction

Plant diseases are a significant threat to agriculture, causing significant losses in crop yield and quality worldwide. Early detection and accurate diagnosis of plant diseases are critical for preventing the spread of diseases and minimizing crop losses. In recent years, deep learning techniques, particularly convolutional neural networks (CNNs), have shown promising results in image recognition and classification tasks.

In this project, we propose a plant leaf disease detection system using deep learning techniques. The proposed system utilizes a CNN architecture to identify plant leaf diseases from images of plant leaves. The goal of this project is to develop a reliable and accurate plant leaf disease detection system that can assist farmers and agricultural researchers in identifying and managing plant diseases.

The proposed plant leaf disease detection system using deep learning techniques is a promising solution to the problem of plant disease detection in agriculture. The proposed system has the potential to improve the accuracy and efficiency of plant disease detection, reduce the negative impact of plant diseases on the environment and the economy, and help improve crop yields.

Dataset colletion

- The Tomato dataset collected from open source website “Kaggle”.
- The Dataset contains 16k(16011) image samples of Tomato crop.
- The Tomato dataset consists of 10 classes corresponding to 9 diseased classes and 1 healthy class.

The 10 disease classes are listed below:

['Tomato_Bacterial_spot',
'Tomato_Early_blight',
'Tomato_Late_blight',
'Tomato_Leaf_Mold',
'Tomato_Septoria_leaf_spot',
'Tomato_Spider_mites_Two_spotted_spider_mite',
'Tomato__Target_Spot',
'Tomato__Tomato_YellowLeaf__Curl_Virus',
'Tomato__Tomato_mosaic_virus',
'Tomato_healthy']

Sample images of Tomato dataset

Dataset Specifications

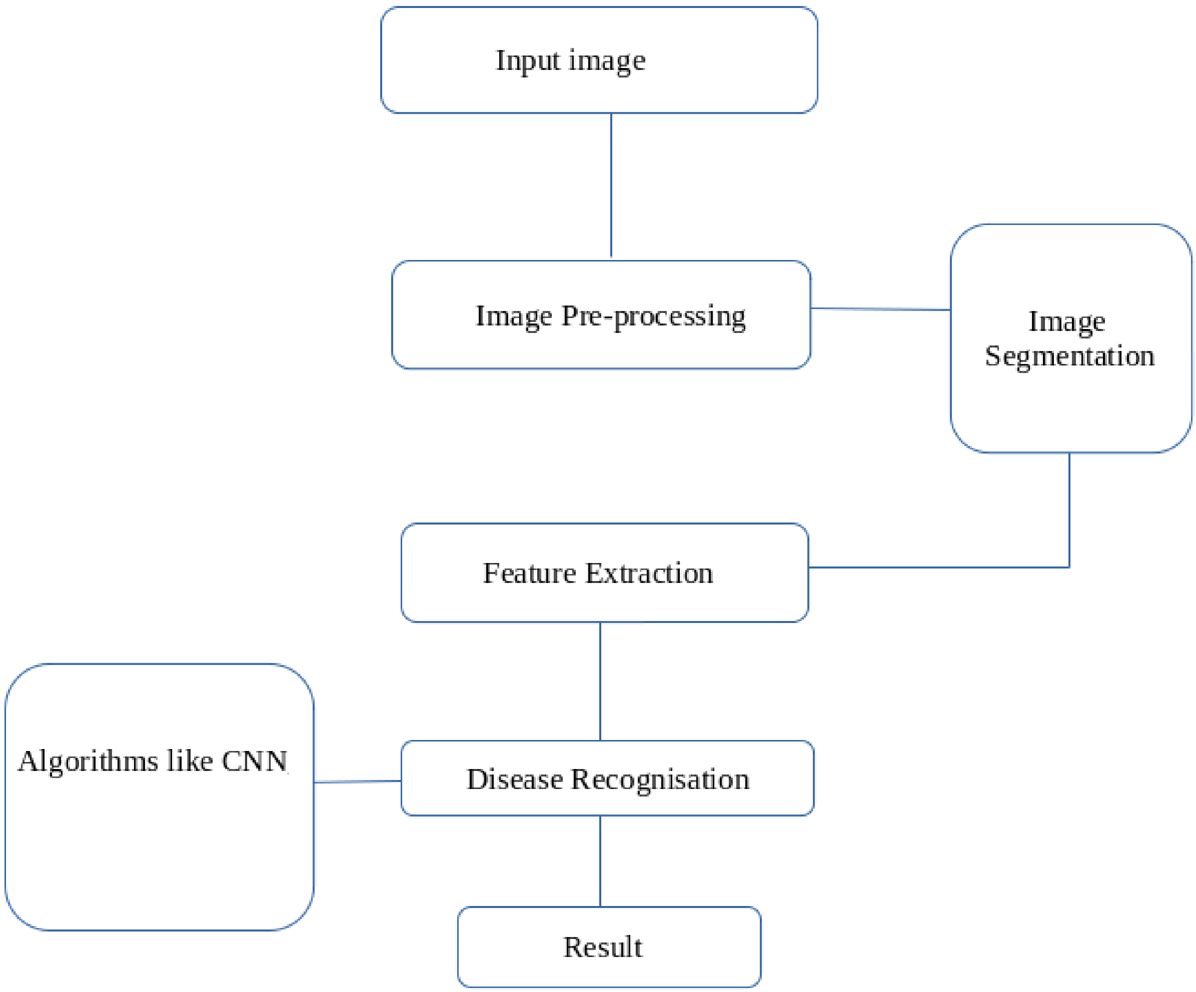
Plant	**Disease Name**	**No.of images**
	Tomato_Bacterial_spot	2,127
	Tomato_Early_blight	1,000
	Tomato_Late_blight	1,909
	Tomato_Leaf_Mold	952
Tomato	Tomato_Septoria_leaf_spot	1,771
	Tomato_Spider_mites_ Two_spotted_spider_mite	1,676
	Tomato__Target_Spot	1,404
	Tomato__Tomato_YellowLeaf__Curl_Virus	3,208
	Tomato__Tomato_mosaic_virus	373
	Tomato_healthy	1,591

Total : 16,011

Indications of Tomato diseases

Disease Name	Indications
Tomato_Bacterial_spot	Small, water-soaked spots on leaves, stems, and fruit that become raised and scabby as they enlarge.
Tomato_Early_blight	Brownish-black lesions on lower leaves that may have yellow halos around them. Lesions may expand and cause defoliation.
Tomato_Late_blight	Dark, water-soaked lesions on leaves, stems, and fruit that turn brown and cause leaves to curl and twist as they enlarge.
Tomato_Leaf_Mold	Yellow spots on upper leaf surface that turn brown or gray as disease progresses. Lower leaf surface may appear fuzzy.
Tomato_Septoria_leaf_spot	Small, circular spots with gray centers and dark brown margins on leaves that may merge to form irregularly shaped lesions.
Tomato_Spider_mites_ Two_spotted_spider_mite	Yellow or white speckles on leaves that turn brown as infestation progresses. Leaves may become distorted and plant may appear stunted.
Tomato_Target_Spot	Circular, water-soaked lesions on leaves that turn brown and have a concentric ring pattern. Lesions may expand and cause defoliation.
Tomato_YellowLeaf__Curl_Virus	Leaves curl upward and inward and may have yellow mottling or mosaic pattern. Plant may appear stunted and fruit may be deformed.
Tomato_Tomato_mosaic_virus	Mottled or mosaic pattern on leaves that may be yellow or green. Fruit may be small, deformed, or discolored.
Tomato_healthy	Green, normal-looking leaves and fruit without any signs of disease.

FlowChart



Input image
Image Pre-processing
Image
Segmentation
Feature Extraction
Algorithms like CNN
Disease Recognisation
Result

Dataset preprocessing

Image preprocessing techniques used in our project to prepare the input images for deep learning model.

Image Resizing and Rescaling: Resizing images to a standard size, such as 256x256 pixels, is a common preprocessing technique used in deep learning. Resizing images to a consistent size ensures that all images have the same dimensions, which is required for inputting them into the model.

In our project, A '**Sequential'** model is defined to resize the input images to a uniform size of 256x256 pixels, and rescale the pixel values from the range of 0-255 to the range of 0-1. This helps in standardizing the input data, as different images can have varying sizes and pixel intensity ranges.

Data Augmentation: Data augmentation techniques, such as random rotation, flipping, and zooming, can be used to increase the size of the dataset and improve the model's ability to generalize to new images.

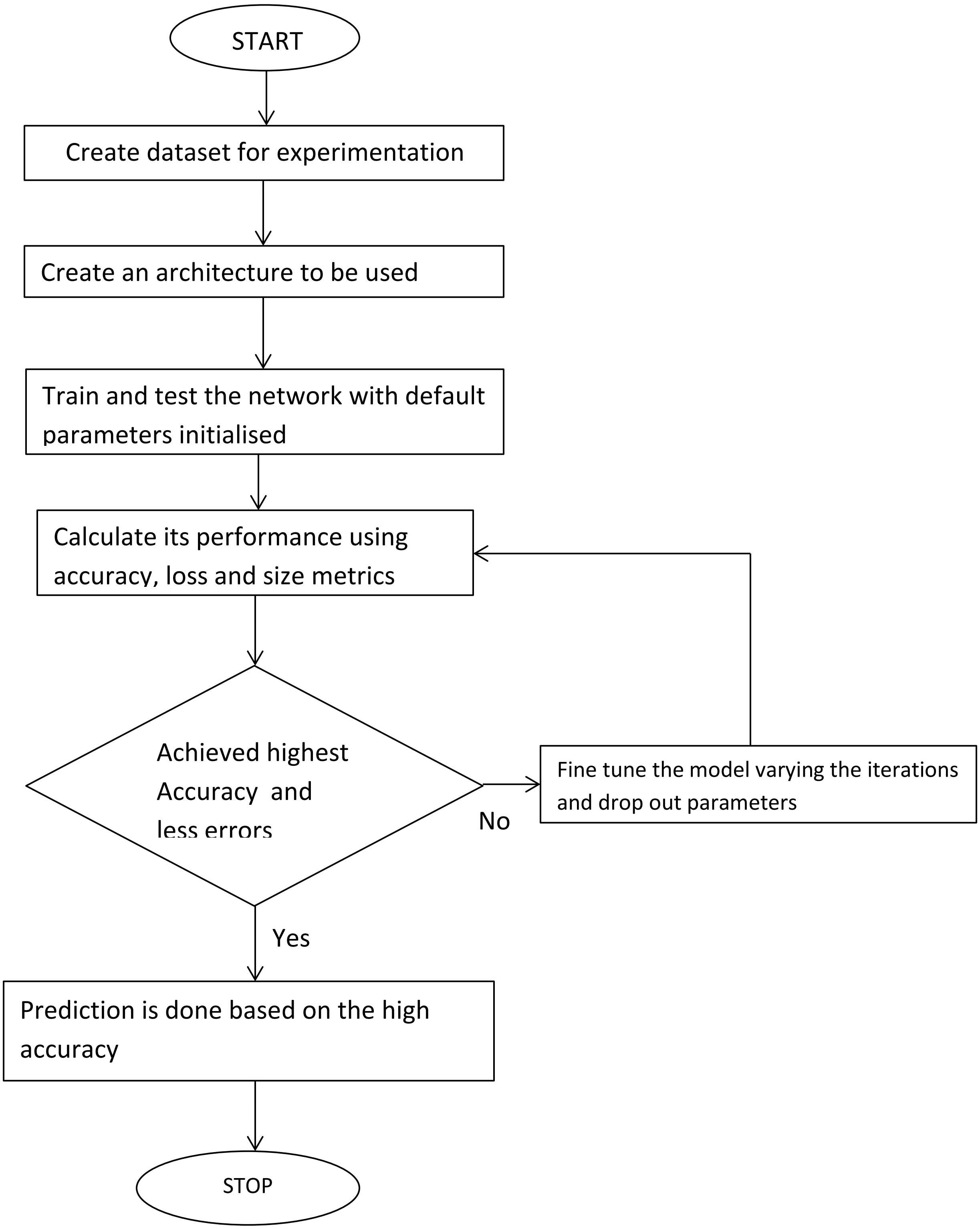
In our project, A **'Sequential'** model is defined to apply random horizontal and vertical flips, and random rotations up to a maximum of 0.2 radians to the input images during training. This helps in generating more training data by creating variations of the original images, which can improve the performance of the deep learning model by reducing overfitting.

Caching and Prefetching: Caching involves storing data in memory, which can improve the speed of accessing the data during training or inference. When a dataset is cached, each element is loaded into memory the first time it is accessed and then retrieved from memory for all subsequent accesses.

Prefetching is the technique of loading the next batch of data in advance, while the current batch is being processed. This can help to reduce the time that the model spends waiting for data to be loaded during training, as the data is already available in memory when it is needed.

In our project, The training, validation, and test datasets are cached and prefetched using the **'cache()'** and **'prefetch()'** methods of the **'tf.data.Dataset'** class. This helps in improving the performance of the data input pipeline, as the images are read from memory instead of being loaded from disk for every epoch during training.

Data flow diagram of project

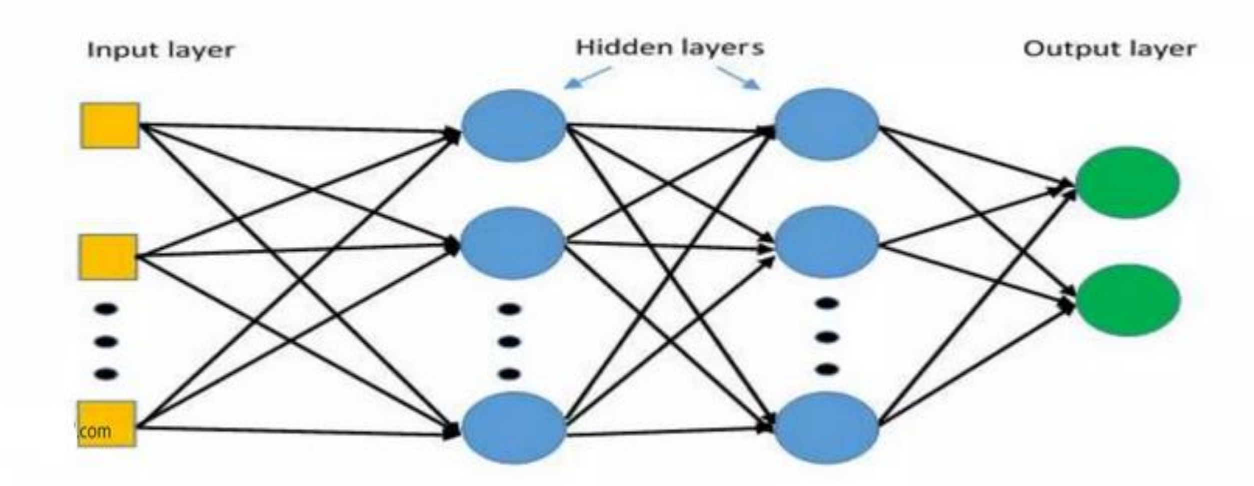


Defining deep learning model

Deep learning is a subfeild of machine learning concerned with algorithms inspired by the structure and function of the human brain called Artificial Neural Networks(ANN).

The ANN architecture is consisted of 3 layers, input layer, hidden layer, and output layer.

Architecture of ANN



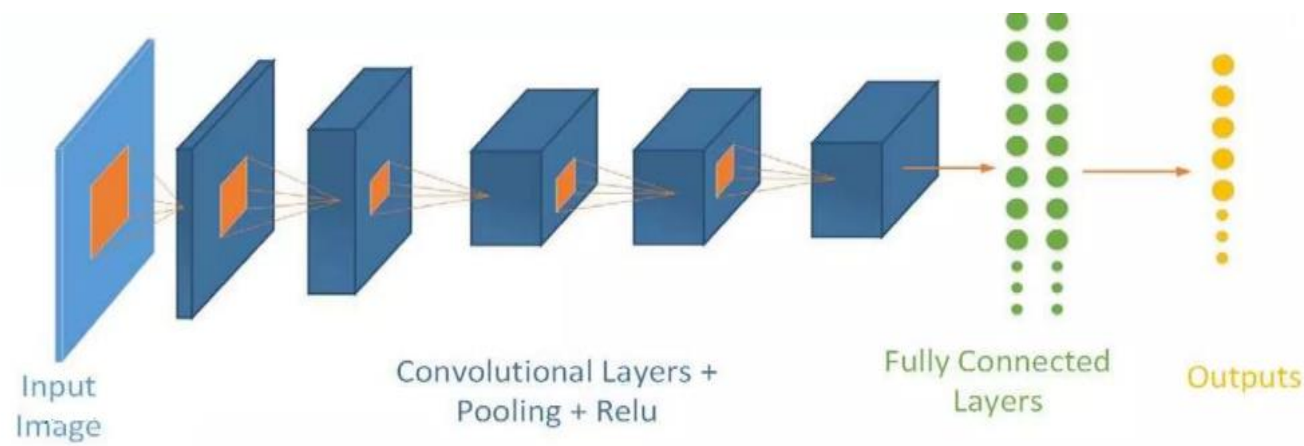
Therefore, a convolutional neural network(CNN) is created and developed to perform plant disease detection and classification using leaf images of healthy and diseased of tomato crop. Recent developments in deep neural networks have allowed researchers to drastically improve the accuracy of object detection and recognition systems. Deep Learning (DL) is the fastest growing and a broader part of the machine learning family. Deep learning uses convolutional neural networks for image classification as it gives the most accurate results in solving real-world problems.

Proposed model

In this project, we proposed and implement **Convolutional Neural Network(CNN)** model or achitecture for classification of diseases of tomato crop.

- CNN's are feedforward neural networks wherein data moves from the input layer to the output layer.
- CNN based classifiers can be directly trained using raw images without the intervention of humans in feature extraction.
- CNN's architecture consists of input, hidden, and fully-connected (output) layers.
- The hidden layers are convolutional, ReLU(Rectified Linear Unit), and pooling layers which are stacked to form a single network.
- CNN can be used to solve classification, clustering, regression, pattern recognition, dimension reduction, structured prediction, machine translation, anomaly detection and computer vision problems.

Architecture of CNN



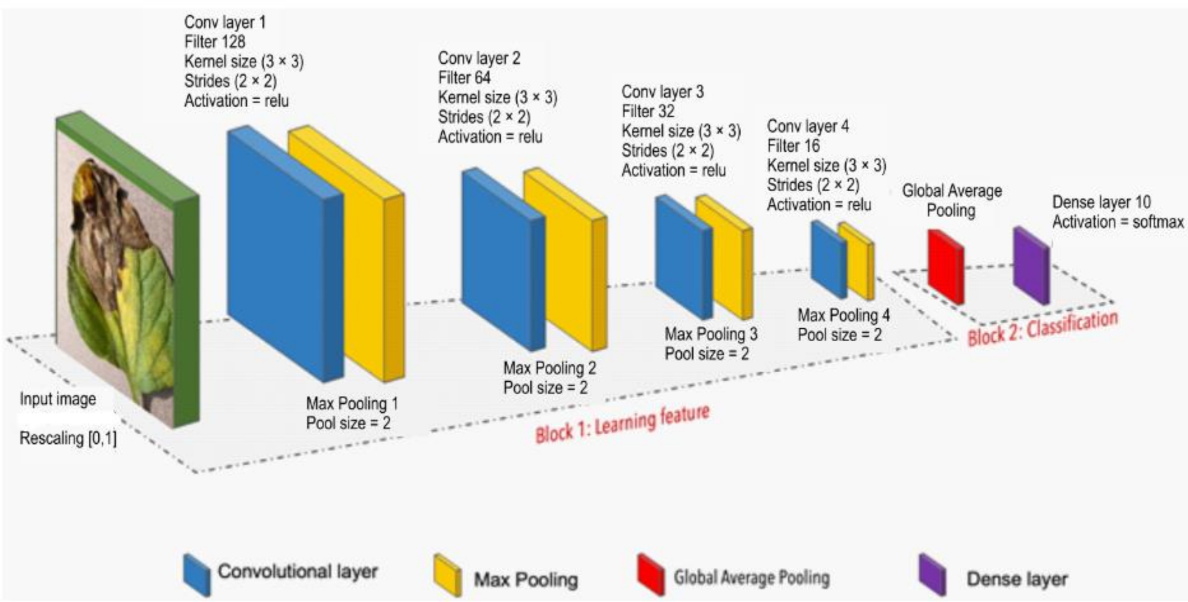
Model creation

We proposed CNN architecture for disease detection in tomato. The network has 256x256 color images as input, which are normalized to (0, 1) values.

The proposed convolutional network has four convolutional layers that use filters whose values were 16, 32, 64, and 128, respectively. These values were assigned in that order since the layers closer to the beginning of the model learn convolutional filters less effectively than the layers closer to the result.

In addition, the kernel size, which represents the width and height of the 2D convolution window, was set to a value of 3 × 3. This value was the recommended value for the number of filters to be used. Finally, Rectified Linear Unit (ReLU) was used as the activation model for each convolved node. After applying the convolutional layer, the maximum clustering layer was applied to down-sample the acquired feature map and condense the most relevant features into patches. This process is repeated for each of the convolutional layers defined in the architecture. The result of the last MaxPooling layer is passed to a MaxAveragePooling layer to be converted to a column vector and connected to the dense layer of 10 output nodes (which represent the 10 categories) used as softmax activation. Each node represents the probability of each category for the evaluated image. Table 1 shows the information of the layer structure of the proposed model.

Representation of proposed algorithm for tomato disease detection



Model creation

For the training process, we use **Adam** as the optimization algorithm. Adam updates network weights iterative based on training data. The loss function was **SparseCategoricalCrossentropy**, one of the most used loss functions for multi-class classification models.

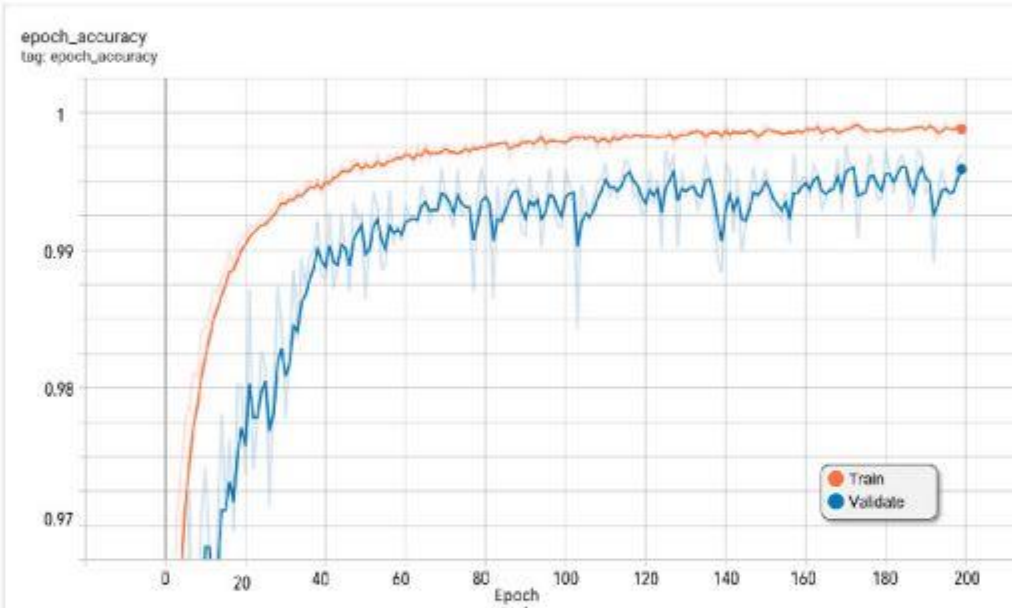
Information of layers structure of the proposed model

Layers	Parameters
Conv2D	32 filters, kernel size of (3,3), 'relu' activation function
MaxPooling2D	Pool size of (2,2)
Conv2D	64 filters, kernel size of (3,3), 'relu' activation function
MaxPooling2D	Pool size of (2,2)
Conv2D	64 filters, kernel size of (3,3), 'relu' activation function
MaxPooling2D	Pool size of (2,2)
Conv2D	64 filters, kernel size of (3,3), 'relu' activation function
MaxPooling2D	Pool size of (2,2)
Conv2D	64 filters, kernel size of (3,3), 'relu' activation function
MaxPooling2D	Pool size of (2,2)
Conv2D	64 filters, kernel size of (3,3), 'relu' activation function
MaxPooling2D	Pool size of (2,2)
Dense	64 units, 'relu' activation function
Dense	10 units, 'softmax' activation function

Training Parameters for the Proposed Model

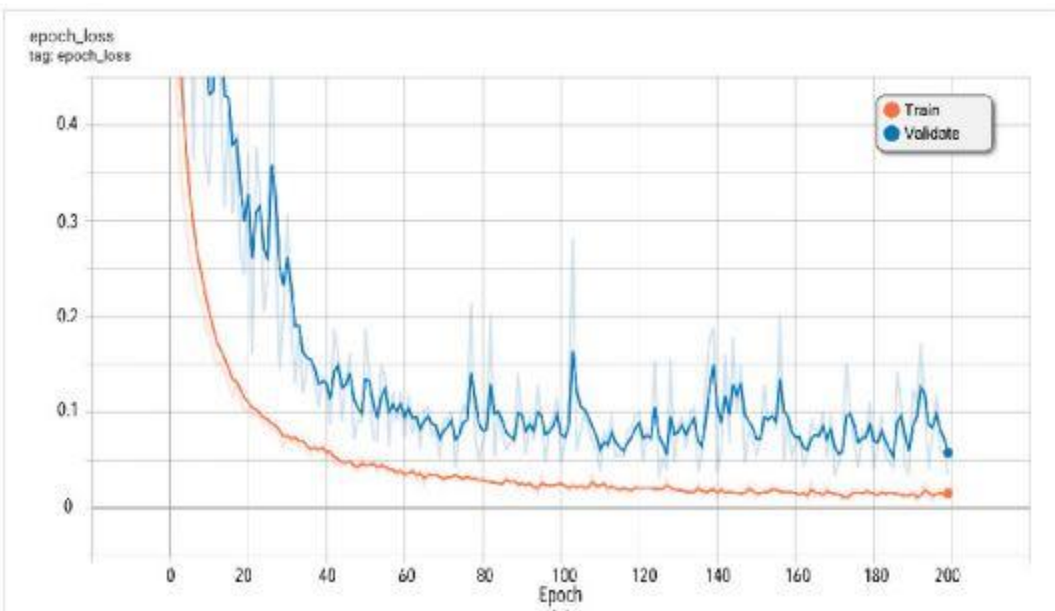
Parameter	Value
Optimization algorithm	Adam
Batch size	32
Number of epochs	200
Steps per epoch	400
Activation function for conv layer	ReLu

Training and Validation Accuracy of proposed model



(a)

Training and validation loss of proposed model

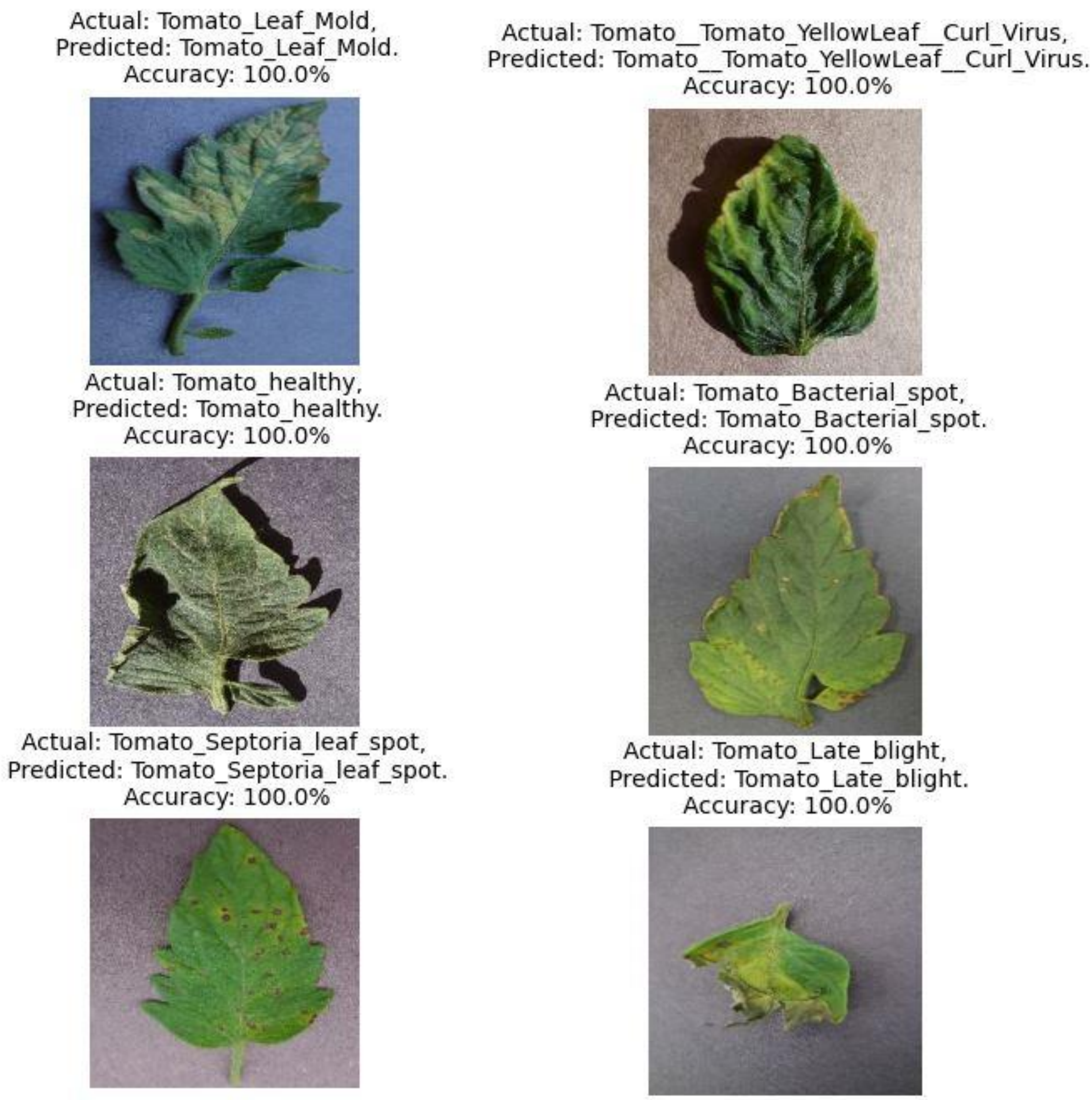


(b)

Results

The architecture and weights obtained from the proposed model were saved as a hierarchical data file to be used during the prediction process. The prediction process uses a dataset with a total of 1632 images. The matplotlib library was used to visualize the prediction result. For each prediction, the image, the true result, and the result of the prediction made with the proposed model were displayed, together with the percentage of accuracy.

Sample predicted images using proposed model



Adantages & Disadvantages

Advantages

- **Accuracy:**
  Deep learning models have the ability to learn from large datasets, resulting in highly accurate predictions for plant disease detection.

- **Efficiency:**
  Deep learning models can process large amounts of data quickly, making them an efficient tool for plant disease detection.

- **Automation:**
  Deep learning models can be trained to automatically detect plant diseases, reducing the need for manual inspection and increasing the speed of diagnosis.

- **Scalability:**
  Deep learning models can be trained on large datasets and deployed on multiple machines, making it easy to scale up for larger plantations.

Disadvantages

- **Data requirements:**
  Deep learning models require a large amount of data to be trained effectively. If sufficient data is not available, the accuracy of the model may be affected.

- **Overfitting**:
  Deep learning models are prone to overfitting, which means that the model can become too specialized to the training dataset and not generalize well to new data.

- **Interpretability**:
  Deep learning models are often considered "black boxes," as it can be difficult to understand how the model arrived at a particular decision.

- **Hardware requirements:**
  Training deep learning models can be computationally intensive and may require specialized hardware, such as GPUs or TPUs. This can make the technology less accessible for small-scale operations.

Conclusion

In conclusion, plant leaf disease detection using deep learning has many advantages, such as high accuracy, efficiency, automation, scalability, adaptability, early detection, and cost-effectiveness. However, it also has some disadvantages, such as data requirements, overfitting, interpretability, hardware requirements, annotation requirements, lack of diversity in data, and environmental factors. Despite these limitations, the benefits of deep learning models for plant disease detection make them a valuable tool in modern agriculture. By leveraging the power of deep learning, farmers and researchers can identify and manage plant diseases more effectively, leading to increased crop yields, reduced crop losses, and improved food security.