

Tomato Leaf Disease Classification Using Deep Learning Techniques

Angulakshmi M*, Ujai BC, Bala Murugan VV, Siddharth V,
Appana Venkata Raju Sai, Ssathyan SR

School of Computer Science Engineering and Information Systems, Vellore Institute of Technology, Vellore, India. *Corresponding Author's Email: angulakshmi.m@vit.ac.in

Abstract

Plant leaf disease identification and evaluation in a timely and accurate manner is crucial for efficient farming operations for crop yield optimization. Employing the most recent advances in technology, specifically the combination of deep learning and the Internet of Things (IoT), this paper offers an efficient approach to identifying plant diseases. We propose a transfer learning-based deep learning classification model which makes use of pre-trained models including Convolutional Neural Networks (CNN), AlexNet, Residual Networks (ResNet), InceptionV3, and Visual Geometry Group-16 (VGG-16). To provide wider accessibility, high-resolution images of tomato plant leaves displaying disease symptoms are gathered from a dataset and saved in cloud storage using Internet of Things devices. Following the image extraction from the cloud, images are preprocessed using data argumentation, normalization, color space conversion, background removal, and noise removal. Different tomato plant disease classes are classified using the pre-trained models CNN, AlexNet, ResNet, InceptionV3, and VGG-16. The deep learning models' accuracy is increased using the transfer learning technique, which also reduces the workout duration. The VGG-16 model outperforms other models in the experiment, recognizing plant illnesses with an astounding accuracy of 93.7% on average, proving the efficacy of the suggested approach. This new approach may revolutionize the diagnosis of diseases affecting tomato plants and promote environmentally friendly agricultural practices.

Keywords: Agricultural IoT, Crop Health, Deep Learning Machine Vision, Disease Prediction, Pre-trained Models.

Introduction

In modern agriculture, the ability to identify diseases in tomato leaves is essential for both early intervention and minimizing crop losses. Visual inspection is a useful method for identifying diseases in tomato plants since they can develop various diseases, such as blights, wilts, and fungal infections, which show up as unique visual signs on their leaves. Deep learning and machine learning techniques have become effective tools for automating the detection process in recent years (1). There are various ways to go about this endeavor. First, machine learning methods such as Support Vector Machines (SVMs) and decision trees can be trained on an image dataset that includes annotated images containing wholesome and damaged tomato leaves (2). Through the extraction of relevant data from the images, such as color, texture, and shape. These models are trained to differentiate between different diseases; on the other hand, CNN is a type of deep learning technique that excels at recognizing

complex patterns in images and may be trained to identify specific disease signs. Their ability to distinguish between healthy and damaged leaves across various disease types has demonstrated impressive success. These techniques use complex algorithms to analyze multiple visual properties, including color, texture, and form, to distinguish between healthy and infected leaves. Datasets including pictures of both healthy and damaged leaves are used to train models. The model developed in this research uses CNN for assistance because working with images has advantages, especially when it comes to image categorization. Models are trained using datasets that contain images of both diseased and healthy leaves. In this paper, the model created takes the help of CNN due to the benefits of working with images, particularly in image categorization to produce customized outcomes. Table 1 (3) displays the proportion of agricultural loss caused by weeds, plant diseases, and pests on each

This is an Open Access article distributed under the terms of the Creative Commons Attribution CC BY license (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted reuse, distribution, and reproduction in any medium, provided the original work is properly cited.

(Received 12th March 2024; Accepted 9th July 2024; Published 30th July 2024)

continent. Africa has the highest percentage of plant disease-related agricultural loss (15.6%), followed by Asia (14.2%) and South America (13.5%), according to the table.

Numerous pathogens, including nematodes, bacteria, viruses, and parasitic plants, are the

cause of plant diseases. All sections of a plant, including the stems, leaves, roots, fruits, and flowers, are vulnerable to infection. Crop yields, quality, and safety can all be impacted by plant diseases. In extreme circumstances, they may result in total crop failure.

Table 1: Agricultural Loss (%) Caused by Pests, Plant Disease and Weeds

Continent	Agricultural loss (%)			
	Pests	Plant Disease	Weeds	Total
Africa	16.7	15.6	16.6	48.9
North America	10.2	9.6	11.4	31.2
South America	14.4	13.5	13.4	41.3
Asia	18.7	14.2	13.4	47.1
Europe	10.2	9.8	8.3	28.2

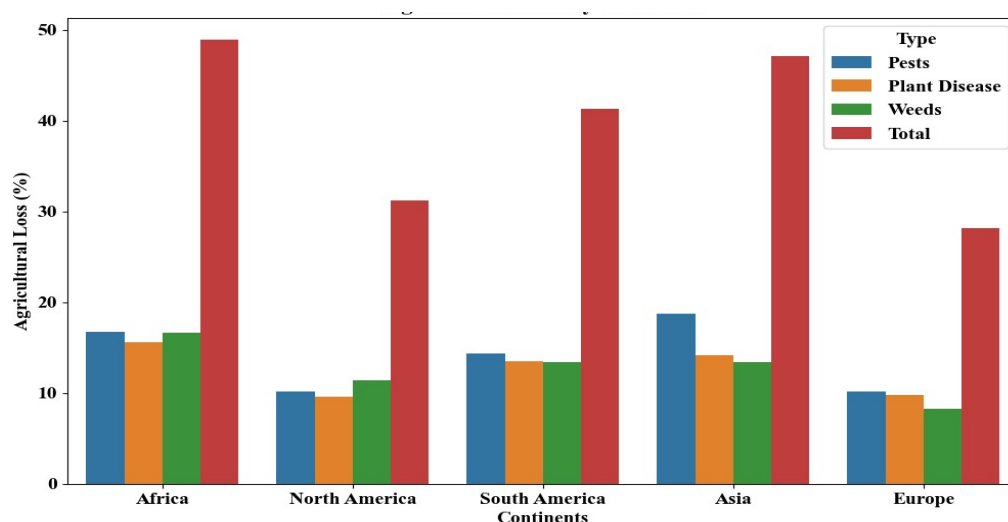


Figure 1: Agricultural Loss (%)

Numerous reasons can lead to plant disease, such as:

- Plant diseases can spread quickly when a single crop is planted over a large area.
- Variations in temperature and precipitation can increase the vulnerability of crops to disease.
- Pests that are resistant to pesticides may emerge as a result of overuse. Plant disease management may become more challenging as a result.

Oerke EC *et al.*, (4) reported the agriculture loss (%) as shown in Figure 1. The graph indicates that, after pests, plant disease is the second-most leading cause of agricultural loss. Plant diseases account for 5% to 20% of agricultural losses in the Continents as shown on the graph. Numerous pathogens, including bacteria, viruses, and fungi, can cause a wide range of plant diseases. They can harm a plant's entire structure, including the roots,

leaves, and fruits. Plant diseases have the power to destroy plants, drastically lower crop yields, and render food crops unmarketable. With its ability to train image-based models on large datasets of both healthy and diseased plant leaves, deep learning can also be applied to large-scale disease monitoring and analysis, which can help reduce overall loss and inform preventative measures (5). A more comprehensive diagram that demonstrates the relationship between plant disease detection and deep learning could incorporate images, accuracy metrics, and a simplified workflow that outlines the steps involved.

In this section, we proposed to predict four tomato leaf diseases using pre-trained models such as CNN, AlexNet, ResNet, Inception V3, and VGG-16. The following list consists of four different diseases of Tomato leaf.

Tomato Late Blight

Though late blight, which is brought on by the

fungus *Phytophthora infestans*, is rare, it can wreak havoc on tomato plants when it's cool and rainy (6). Either young (upper) or old (lower) leaves can be affected by late blight.

Tomato Leaf Mold

In humid greenhouses with inadequate air circulation, the fungus *Fulvia fulva* can produce leaf mold, which can lead to various issues. On lower leaves, this fungal disease manifests as fuzzy plenty of buff-colored spores on the underside and yellow spots on the top surface. As the disease spreads higher on the plant, these leaves begin to drop prematurely.

Tomato Mosaic Virus (ToMV)

ToMV is an RNA-based virus that infects plant species and is stable and extensively dispersed. Typically, it causes uneven ripening, brown wall or internal browning on some fruit kinds, and bending and mosaicking of the leaves (7).

Tomato Yellow Leaf Curl Virus (TYLCV)

The single genomic component of the monopartite tomato yellow leaf curl virus (TYLCV) is homologous to DNA-A of the bipartite begomoviruses. To infect a system, these single-stranded viruses only need DNA-A (8).

Thus, we present a deep learning-based transfer learning method with an integrated IoT approach to identify tomato plant diseases from leaf images. Using this technique, farmers can upload photos of their plants' leaves via a network connection or gather them directly in the field. The images are then safely saved in cloud storage. To predict diseases, a data analysis component retrieves these images from the cloud. To predict different tomato plant diseases, we use deep learning models in conjunction with transfer learning. The method minimizes the amount of data and the time needed for model testing and training. To predict tomato plant diseases, pre-trained models such as CNN, AlexNet, ResNet, Inception V3, and VGG-16 are modified and improved.

Findings are saved on cloud servers and can be accessed by researchers, farmers, and agricultural specialists for planning mitigation and diagnosis.

Key contributions of this proposed method:

- Applying transfer learning for efficiency: Pre-trained models reduce data requirements and speed up model development.

- Deep learning for accurate diagnosis: This model offers high accuracy in identifying specific plant diseases.
- Enhancing the effectiveness of the model by integrating IOT with the prediction model.

Various deep-learning models and techniques have been discussed in this section. Several researches about deep learning models detecting diseases in plants have been done, and few of them have been discussed here. Tomato is a widely fruit grown in India. Diseases affect their growth and decrease their production in the market. In recent years Artificial Early disease detection in a variety of crops has been made possible in large part by artificial intelligence and machine learning.

Agarwal *et al.*, (9) used convolution neural networks to classify and detect diseases. This CNN model consists of eight layers: two fully connected layers, three max-pooling layers, and three convolutional layers. Nine distinct categories of illnesses have been classified by the authors. In the course of this study, 7000 images in all were examined. The accuracy ranged from 76% to 100% depending on the kind of illness. In this study, an average accuracy of 91.2% was reached. To compare performance, the same researchers also conducted tests.

For VGG-16, the accuracy was 77.2%; for Mobilenet, it was 63.75%; and for Inception, it was 63.4%. LeNet, a CNN variant, was employed by Tiwari *et al.*, (10) to identify and categorize illnesses in tomato leaves. They made use of 18160 pictures from ten distinct classes. To expedite the training process, researchers in this experiment downsized every picture to 60 X 60 resolution. The convolutional, stimulation, pooling, and fully linked layers of the traditional CNN model are all present in LeNet. In addition, they have expanded the original LeNet design with new layers.

This experiment yielded a 94% accuracy rate. Guerrero-Ibañez *et al.*, (11) have used 4 architectures of CNN to detect the leaf diseases. LeNet -2 Convolutional layers, VGG16 -13 convolutional layers, ResNet50, and Extreme Inception (Xception) (36 convolutional layers). The researchers have used 14,903 images which belong to 9 different classes. Adding these many layers to their architecture is a complex task. As they added more convolutional layers, the overall complexity of these models increased (12). Use of Random Forest Classification algorithm to classify

the healthy and diseased images.

The Histogram of an Oriented Gradient (HOG) is used for feature extraction. A feature descriptor employed in computer vision and image processing for the recognition of objects is the Histogram of oriented gradients or HOG. The accuracy was 70.4%. Random Forest model accuracy was slightly better compared to other classification machine learning models. The author also used three-component descriptors – Hu moments, Haralick texture, and Color Histogram. Hu moments extract shape, Haralick texture extract texture, and color histogram represent the color distribution in an image.

In comparison to conventional machine learning models, the use of Deep Learning models for plant disease detection and classification has demonstrated improved accuracy. A comprehensive review of the Deep Learning models used for detecting plant diseases is provided by the authors in their publication (13). All of the deep learning models from 2012 to 2018 have their parameters, salient characteristics, and pros and cons mentioned. A breakdown of all the models' performance measures during these years is given. A method based on the CNN and Learning Vector Quantization (LVQ) algorithms was presented in the paper (14). Learning Vector Quantization refers to a neural network that combines supervised and competitive learning. The average accuracy that the researchers were able to achieve was 86%. There were a total of five distinct classifications utilized. One for healthy leaves and four for leaf diseases. This research paper (15) describes a work in which the Resnet model has been used to classify plant diseases with leaf images using deep-learning approach. Therefore, it highlights the importance of agricultural productivity in India and the global food industry with a special focus on commodification as well as acknowledging disease detection challenges due to infrastructure scarcity. With SGD optimization, the accuracy is as high as 87.5% using ResNet for classifying plant diseases in this work. The authors introduce layer-wise fine-tuning and later propose contributions toward disease detection via this new ground of visual recognition accommodated through the rapid growth in technology. In the paper (16), the study utilized fine-tuned pre-trained CNNs, namely VGG16, ResNet, and Inception, in detecting

diseases of plant leaves. The training set consisted of 5,000 images: 3,500 healthy and 1,500 diseased leaves. The dataset was pre-processed for quality and agreement. The accuracy of the models achieved was at a maximum of 99.35% when tested on Plant Village data, but it went down to 31% on differently conditioned images. The advantages of the approach were high accuracy, rapid classification suitable for mobile applications, and general applicability across species and diseases. However, the model performance degraded in the presence of varied image conditions and could classify only the upright leaf orientation type on a uniform background. The comparisons of various methods discussed above are given in Table 2.

Methodology

We designed an automated system that involves taking images of the tomato leaves using a Raspberry Pi HQ camera connected to an NVIDIA Jetson Nano GPU processor, processing the images using Transfer learning-based deep learning models, and recording the results on the Cloud to create an efficient, reasonably priced, and easily deployable solution. The images are preprocessed using data argumentation, normalization, color space conversion, background removal, and noise removal are performed. Then the images are classified using pre-trained deep learning models such as CNN, AlexNet, ResNet, Inception V3, and VGG net. These models are already pre-trained and used for the classification of tomato leaf diseases. It classifies the leaf as normal, leaf late blight, Tomato mold leaf, and yellow virus leaf. The proposed method is shown in Figure 2. In this section, we proposed a data pre-processing architecture for the Plant disease detection model, we use deep learning models which are particularly good at extracting complex patterns from image data, making them well-suited for plant disease detection.

However, to fully utilize raw images, we require careful preprocessing to unlock their full potential. Here's an In-Depth Look at Some Important Methods.

Image Augmentation

A common problem in deep learning is data scarcity. Augmentation artificially enlarges the dataset with diverse variations of existing images, enhancing model generalizability and preventing over fitting. Techniques like Rotation, flip, scaling,

cropping, color jittering, noise addition, and elastic deformations are common methods (17) applied to leaves. For the proposed method,

argumentation such as Enhanced image, height, width shift, rotation, shearing and, horizontal flips are applied.

Table 2: Comparison of different Methods for Tomato Leaf Detection

Reference	Model	Data	Accuracy	Advantages	Disadvantages
Agarwal <i>et al.</i> , (9)	CNN (8 layers: 2 fully connected, 3 max-pooling, 3 convolutional)	7000 images, 9 categories	76%-100% (average 91.2%)	High average accuracy, effective for multiple disease categories	Limited data, varied accuracy depending on diseases
Tiwari <i>et al.</i> , (10)	LeNet (CNN variant)	18,160 images, 10 classes	94%	High accuracy, efficient training with downsampled images	Limited to specific resolution, may not generalize well
Guerrero-Ibañez <i>et al.</i> , (11)	LeNet, VGG-16, ResNet50, Xception	14,903 images, 9 classes	93.25% (VGG-16)	Extensive use of different architectures, high accuracy	Increasing complexity with more layers, high computational cost
Bhise <i>et al.</i> , (12)	ResNet	Not specified	87.5% (SGD optimization)	High accuracy with SGD optimization, potential for fine-tuning	Infrastructure challenges for disease detection
Saleem <i>et al.</i> , (13)	ResNet50	5,000 images	Validation: 91%, Testing: 89%	High precision for some diseases, improved performance with data augmentation	Difficult to distinguish visually similar diseases, needs larger dataset and field validation

Data Normalization

Standardized input values are frequently assumed by deep learning algorithms. By scaling pixel intensities to a standard range (such as 0–1 or mean–standard deviation), normalization promotes convergence and increases training stability (18). The dataset images are normalized for further processing

Color Space Conversion

Certain features specific to a disease can be emphasized by converting Red Green Blue (RGB) images to other color spaces. HSV highlights saturation and hue, which helps identify illnesses based on color. We used CIELAB a color perception tool that helps identify subtle symptoms (19).

Segmentation and Background Removal

Plant diseases rarely affect the entire image. Isolating the relevant region (i.e., leaf) through techniques like thresholding, level sets, or deep learning-based segmentation methods improves disease detection accuracy by focusing on the affected area (20). Hence segmentation using the thresholding method is performed on the given dataset images

Noise Reduction and Filtering

Confusion in Deep learning models can arise from noise introduced during acquisition or transmission. Techniques like Gaussian filtering, median filtering, and bilateral filtering smoothen images while preserving edges crucial for disease identification (21). In the proposed method, we applied Gaussian filtering to remove noises.

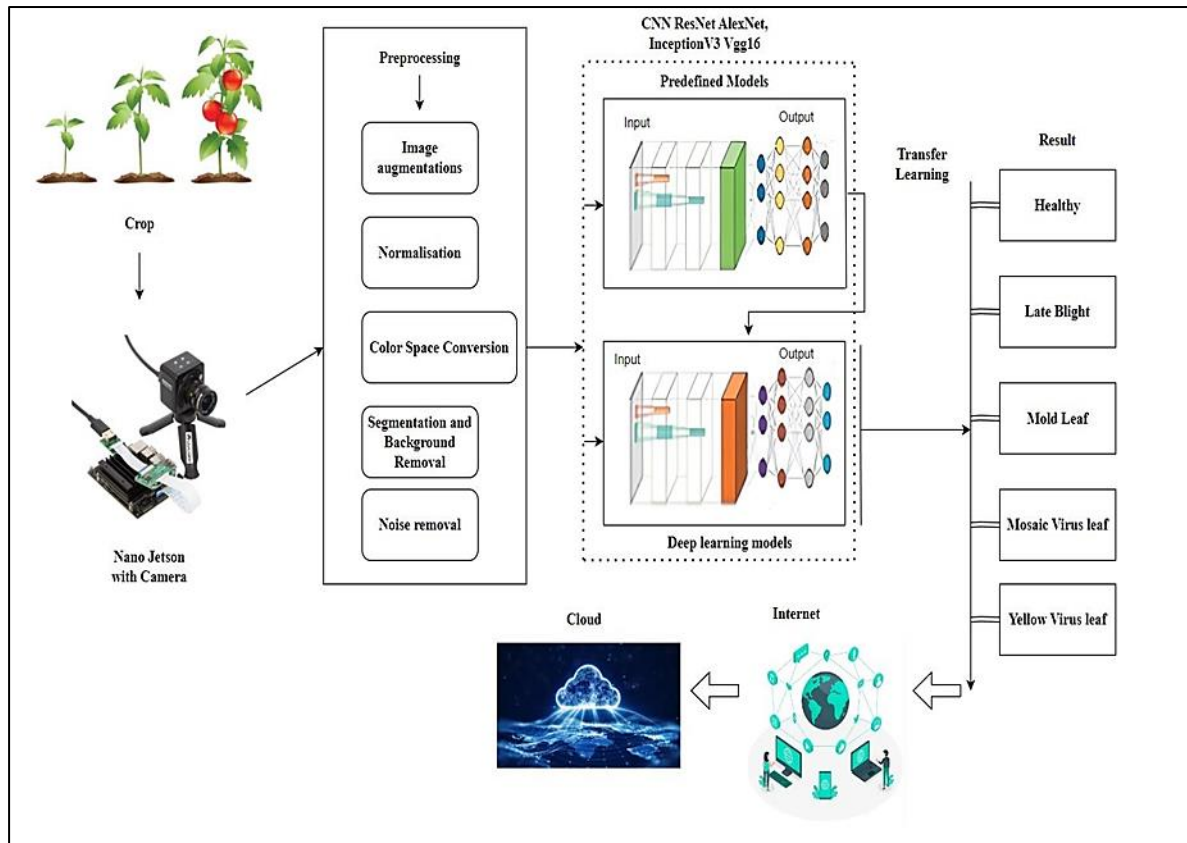


Figure 2: Proposed Method

Transfer Learning

Apply learned models to huge ImageNet, such as VGG16 or ResNet. datasets to leverage pre-learned features and accelerate training, especially with limited data (22). In the proposed method we have applied 5 pre-trained networks and compared their results by carefully selecting and applying these data preprocessing techniques, we can prepare our image data for optimal performance in deep learning-based plant disease detection models. Experimentation and evaluation are key to finding the best approach for our specific needs. In this section, we proposed five transfer learning based Deep learning models.

CNN

Deep learning neural network architectures such as convolutional networks are utilized primarily in the field of recognizing patterns in images (23). A prime instance of an artificial neural network is CNN. Three layers comprise a traditional Artificial Neural Network (ANN): input, hidden, and output. Convolutional, non-linearity, pooling, and fully connected layers make up the layers of a CNN that resemble this. A great deal of tasks requiring image-driven pattern recognition is handled by

CNNs. Benefits CNN can automatically recognize and determine pertinent aspects in images one key component of the strategy is that the networks have been developed to automatically adjust to the spatial arrangement of features the subject learns and pull relevant characteristics from the images dynamically translation invariance ability to recognize translation invariance a technique that aids in confirming the presence of illnesses unrelated to leaf orientation.

Natural Language Processing (NLP), computer vision, and image classification are a few additional applications for CNN (24). Agarwal M and Others (9) have reported the CNN architecture is shown in Figure 3. We must first decide on our model's architecture. Our data is input in 256×256 format with three channels. We have five distinct classes altogether, thus five is the total amount of output classes that we have set. The structure of our model consists of pooling and convolutional layers. Initially, a 3×3 convolutional layers of 32 filters is present. A 2×2 max pooling layer follows this. This minimizes the size by highlighting things at a lower level. This structure is repeated twice to add maximum pooling layers of size 2×2 and convolutional layers containing 64 and 128 filters,

respectively, to deepen our model. A flattening layer is applied to the resultant feature map to convert it into a flat vector. A thick, covert coating of Next, a 128-neuron hidden (dense) layer is introduced. This layer improves generalization and enriches the learned characteristics. Lastly, the output layer uses the softmax activation function to compute the likelihoods between classes using four neurons. To teach our model, we must

ascertain the ideal metrics and function. We employ the Rectified Adam optimization technique in this paper. This approach facilitates the more effective use of gradients by dynamically adjusting the learning rate. Additionally, because categorical cross-entropy is frequently employed in multiclass classification tasks, it is utilized as the loss of function during training.

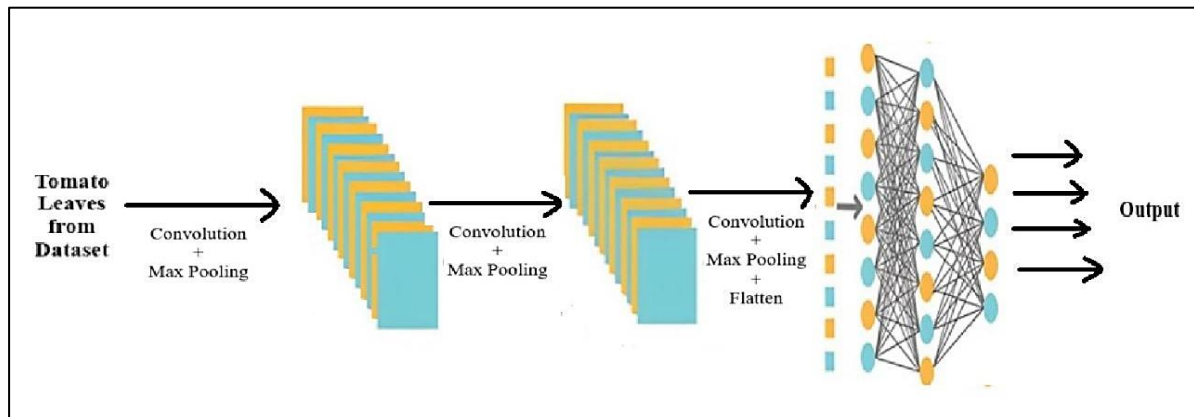


Figure 3: The Architecture of CNN Based on Agarwal *et al.*, (9).

ResNet

In the proposed method, ResNet employs skip connections, which function as informational "magic highways," to counteract vanishing gradients. These shortcuts guarantee that important visual cues about healthy and diseased plant tissues reach the diagnosis station (final layers), unlike regular roads (convolutional layers) where details can be lost. Consider the analysis of leaf images. While regular models may find it difficult to retain subtle color changes that are layered deeply, ResNet's shortcuts allow it to learn

intricate disease patterns while retaining essential fundamental information. Because of this, ResNet is an effective method for detecting plant diseases early and accurately, which could save crops and livelihoods. To achieve this, we have limited the batch size to 32, and the output classes to 5 (the total number of qualified individuals). We gave ourselves some leeway in determining the total amount of input channels, which will be determined by the various experiments we carried out. Padshetty *et al.*, (25) used ResNet Architecture as shown in Figure 4. The proposed method used the same architecture.

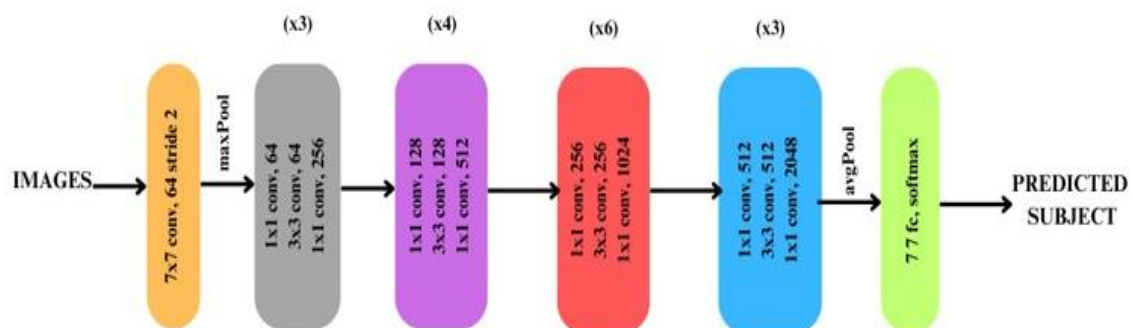


Figure 4: The architecture of ResNet based on Padshetty *et al.*, (25)

Benefits: It was one of the models that first demonstrated the competitiveness of deep

learning in image classification. It is deep enough to capture the deeper patterns involved in leaf

diseases. It utilizes the ReLU activation function, enabling quicker training and solving the vanishing gradient problem. It is included to maximize the use of GPU processing for faster training.

Alexnet

The science of computer vision was greatly enhanced by the groundbreaking CNN architecture known as AlexNet, which was first introduced in 2012. The network, which was created by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, was essential to deep learning's success in image recognition applications first, second, third, and fourth layers, which are then followed by the pooling layer and the three fully-connected layers in the fifth layer. During the back-propagation optimization process for the AlexNet architecture, convolutional kernels are retrieved by using the Stochastic Gradient Descent (SGD) technique to optimize the entire cost function. In general, the pooling layers work on the convolved feature corresponds to aggregate the data within the specified neighborhood window using a max pooling operation or an average pooling operation. The convolutional layers behave on the feature maps that are supplied with the slipping convolutional kernels in order to produce the convolved feature maps. Some of the useful techniques, such as the dropout regularization method and the Rectified Linear Units (ReLU) the non-linear layer, are responsible for AlexNet's

success. A rectifier with a half-wave function called the ReLU can greatly speed up training and avoid overfitting. The dropout technique, which is typically used in the fully connected layers of the AlexNet architecture, can be thought of as a type of normalization by stochastically lowering the number of input cells or hidden neurons to zero in order to lessen the co-adaptations of the neurons (26). One unique feature of AlexNet was that its activation functions were ReLU. ReLU made it possible to solve the vanishing gradient issue, which sped up training convergence. By normalizing the responses of nearby neurons, Local Response Normalization (LRN), which was incorporated into the first two convolutional layers, improved the network's capacity for generalization. Figure 5 shows architecture of Alexnet as used by Chen *et al.*, (27), applied to the proposed method.

Benefits: ResNet utilizes residual connections, thereby making the training of much deeper networks possible by solving the vanishing gradient problem. Capable of training much deeper networks such as 50, 101, or 152 layers, which are potentially good at grasping very intricate datasets. They aid in the training of deeper networks by adding short-circuit connections. This implies that shortcut paths to some layers are available so that the degradation problem can be alleviated. Performance: The network exhibits high performance in most cases across a range of image classification benchmarks.

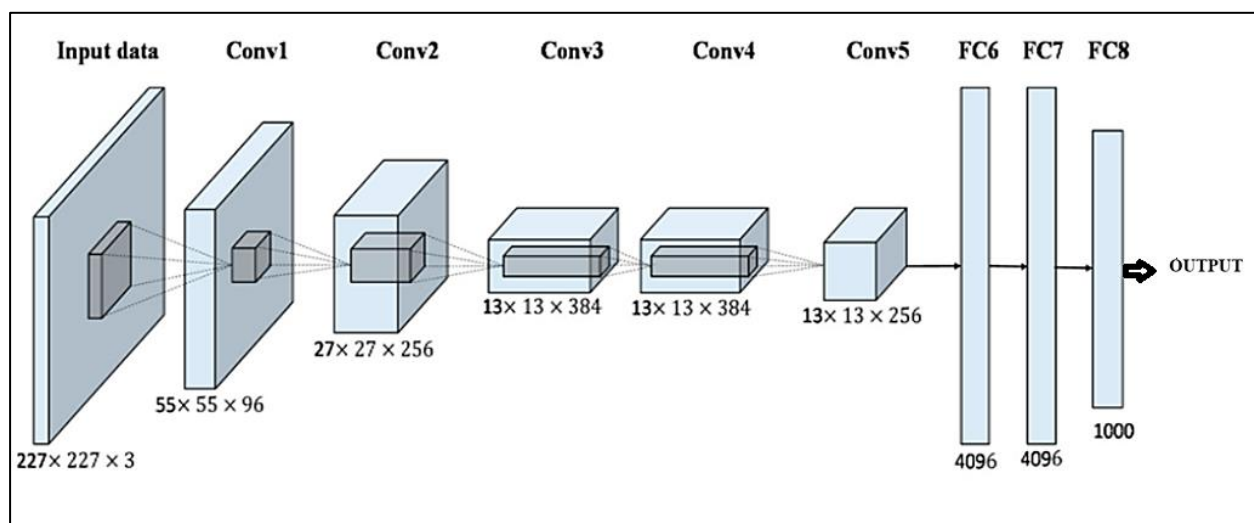


Figure 5: Architecture of AlexNet Based on Chen *et al.*, (27)

VGG-16

In the proposed methodology, VGG-16, known for

its deep architecture, utilizes a sequential arrangement of convolutional layers followed by

max-pooling layers, culminating in fully connected layers. This structured design enables the model to capture intricate features of plant images, essential for accurate disease detection. Unlike some traditional models, VGG-16's approach enhances feature learning, ensuring that subtle color changes indicative of plant diseases is effectively retained during the training process. With its well-defined architecture, VGG-16 stands as a robust tool for early and accurate plant disease detection, contributing significantly to crop preservation and agricultural sustainability. Figure 6 shows the architecture of VGG16 as proposed by Alatawi *et al.*, (28). The same type is used in the proposed method. An RGB image with a preset dimension of 224 by 224 is the input for the Cov1 layer. After passing through an array of convolution (conv.) layers, the image is processed through filters having an extremely tiny receptive field 3×3 , which is sufficient to capture left/right, up/down, and center concepts. As a linear transformation on the source channels, each of the configurations additionally makes use of 1×1 convolution filters. For three-by-three convolution layers, the amount of padding corresponds to one pixel, and the convolution stride stays constant at one pixel. The convolution layer's input's spatial padding is made to preserve the depth of field even after the convolution. Five classes are finally classified.

Benefits: VGG 16 is Very simple and uniform architecture where only 3×3 convolution layers are stacked up one after another. High and enough to realize complicated characteristics (16 layers). Makes it easier for understanding and applying with the same use of small convolution filters.

Inception-V3

In the proposed method, Inception V3, developed by Google, stands out for its intricate design, incorporating inception modules that capture multi-scale features within images. To minimize the size of the data, the Inception module provided by the Inception Neural Network employs maximum pooling and filters of varying sizes (29). This architecture has proven to be highly effective in the domain of plant disease detection, showcasing remarkable capabilities in the early and accurate identification of various diseases. Better optimization methods, lower computing costs, a more complex neural network, and increased efficiency are all features of Inception V3. The utilization of Inception advanced feature extraction capabilities enhances its ability to discern subtle visual cues, contributing to precise disease diagnosis. Its robustness lies in the diverse set of features captured by the inception modules, facilitating the extraction of both global and local information from leaf images. Studies use the versatility and robustness of Inception V3, positioning it as a pivotal tool in advancing the field of plant disease detection and precision agriculture. The author Saritha *et al.*, (29) proposed an Inception Net as shown in Figure 7. We have used the same architecture in the proposed method.

Benefits: InceptionV3 optimally facilitate the capturing of multi-scale features by the use of parallel convolutional layers that are processing different filter sizes. InceptionV3 can achieve high performance with comparatively fewer parameters when compared to other networks of similar depth. Auxiliary classifiers are incorporated in the model to counter the vanishing gradient problem and improve convergence.

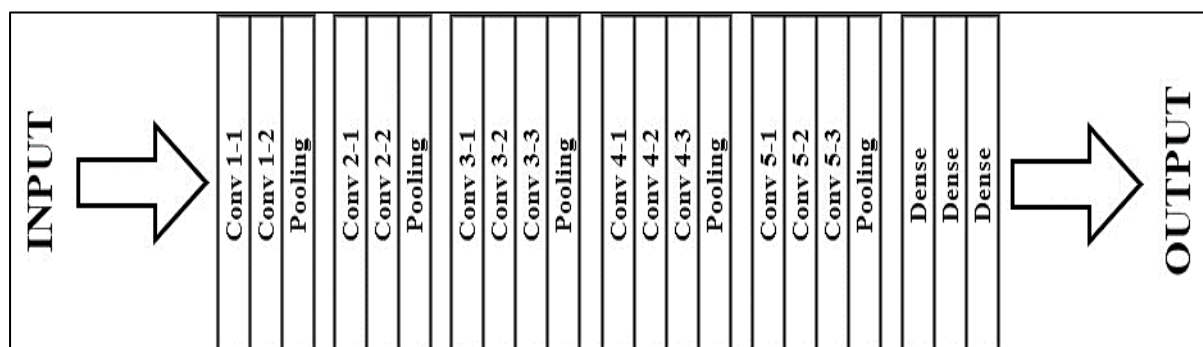


Figure 6: Architecture of VGG-16 Based on Alatawi *et al.*, (28)

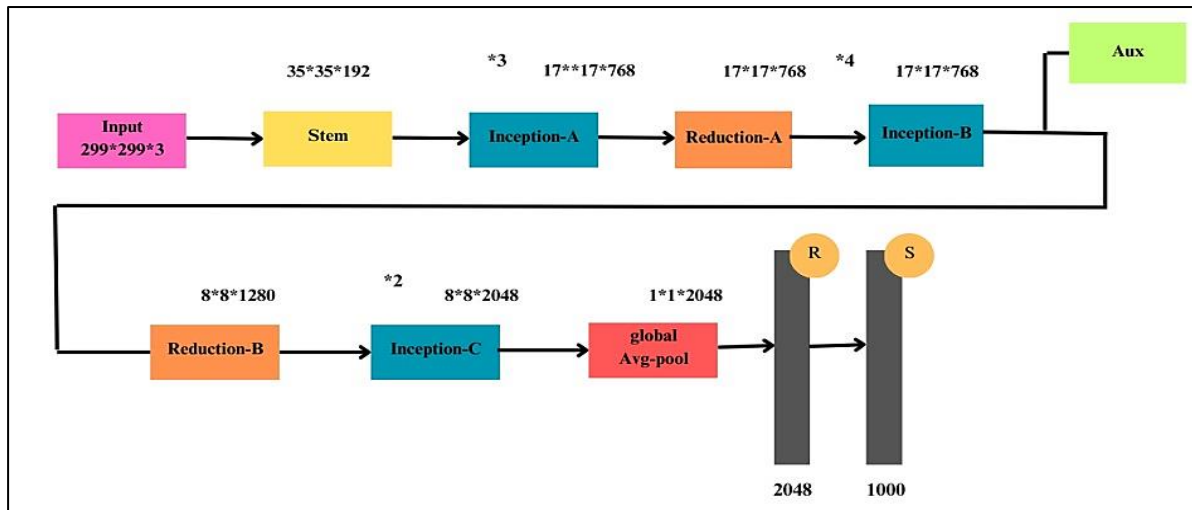


Figure 7: Architecture of Inception V3 Based on Saritha *et al.*, (30)

Comparative Summary

CNN is the basic block to perform classification of images in just the right order of complexity as well as efficiency. AlexNet may be used for relatively simpler problems and where we have the minimum available computational resources. While, ResNet will be suitable for very complex and deep networks in which it can be trained over the very deep models without degradation. It is also efficient in capturing multi-scale features using fewer parameters. VGG-16 is inherently a simple and a deeper one but containing more parameters than InceptionV3 (30).

Criteria for Selection of models

Image Classification Task: These models have shown strong capabilities of correctly classifying an object and a scene that appears in an image, which would also be robust in this task of recognizing plant diseases from leaf images.

Transfer Learning Capability: All these models have, so to say, the qualities of being pre-trained on large datasets, for instance, on ImageNet. Its training features that can also be transferred and fine-tuned to learn specifically from our small dataset regarding tomato plant disease images.

Depth and complexity: Architectures such as ResNet and InceptionV3 are very deep by design, with novelties such as residual connections in the case of ResNet and inception modules for the InceptionV3 architecture; thus, they can encapsulate detail and fine patterns extracted from the images. They are going to be helpful in extracting subtle differences between healthy and diseased leaf patterns of a plant.

Computational Efficiency: While deeper models

such as ResNet and InceptionV3 outperform models such as VGG-16 and AlexNet have an architecture that is relatively simple yet shows effectiveness. This trade-off in complexity must ensure that performance is developed within the bounds of computational efficiency, especially during model deployment on resource-constrained devices or in IoT environments.

Community and Tool Support: These models include a rich community, meaning well-documented implementations and pre-trained weights exist. This in turn makes it easier to bring the models from paper to real-world implementations.

Experimentation and Benchmarking: Most prior works and benchmarks have used these models in the past as baselines or reference points, making the comparison and validation of result much easier.

Therefore, these five models are selected to use in plant leaf disease identification: CNN, AlexNet, ResNet, InceptionV3, and VGG-16. The reason is that they have been proven best in problems of image classification, via transfer learning; for handling complicated features in the image, they are less resource-intensive in terms of computation; at last, they are very well supported by the community together with benchmarking in similar applications. These models together contribute to an accurate and effective framework for identification and diagnosis of diseases in tomato plants using deep learning techniques.

Results

We utilized the images from camera obtained using IOT model which comprises images of various

tomato plant diseases and healthy leaves. The dataset consists of three classes: Diseased Tomato Plants (Tomato late blight, Tomato mold leaf, Tomato mosaic virus and Tomato yellow leaf curl virus), Healthy Tomato Plants (Tomato leaf). 10,182 images total from camera shows tomato leaves in four different disease- and healthy-class categories. This dataset includes samples of leaves with varying degrees of disease infection as well as a broad range of diseases. It is clear that there is an imbalance in the dataset from Table 3's distribution of sample numbers across classes, where there are noticeably different sample counts for each class. The maximum number of samples associated with the Yellow Leaf Curl Virus disease is 5357, while the minimum number of samples associated with the Mosaic Virus disease is 373.

Table 3: Distribution of Samples in the Dataset

Class Label	Sample Count
Late Blight	1909
Leaf Mold	952
Yellow Leaf Curl Virus	5357
Tomato Mosaic Virus	373
Healthy	1591
Total	10182

All images were standardized to a resolution of 256x256 pixels. During the training phase, 90% of the dataset was used for training the deep learning model, while the remaining 10% was reserved for testing. Additionally, validation was performed on the training dataset using a 10% validation split. Cross-validation with k=5 using stratified k-fold was employed to ensure robust model evaluation. Figure 8 shows different types of tomato leaves with disease.

The following indicators were used to assess the effectiveness of the suggested plant disease detection approach.

Accuracy

This metric, which is computed as the ratio of correctly classified observations to total observations is used to evaluate how accurate the model's predictions are overall. The Equation [1] gives it formula to calculate it

$$Accuracy = \frac{TruePos + TrueNeg}{TruePos + TrueNeg + FalsePos + FalseNeg} \quad [1]$$

Whereas True Pos, for instance, a true positive in the context of plant disease detection happens when the model accurately classifies a diseased plant leaf as such.

True Neg, An illustration of a true negative in plant disease detection is when the model accurately classifies a leaf of a healthy plant as healthy.

False Pos, An illustration of a false positive in plant disease detection is when a healthy plant leaf is mistakenly classified as diseased by the model (Type I error).

False Neg, An illustration of a false negative in plant disease detection is when a Type II error causes the model to mistakenly classify a diseased plant leaf as healthy.

Specificity

The specificity (which is correct Negative Rate): Specificity gauges how well the algorithm can distinguish between all real, healthy plant cases and identify healthy plants. The Equation [2] gives the formula to calculate it.

$$Specificity = \frac{TrueNeg}{TrueNeg + FalsePos} \quad [2]$$

Precision

Precision can be defined as the proportion of all favorably forecasted leaves by the system to the accurately predicted disease-affected leaves. The Equation [3] gives the formula to calculate it.

$$Precision = \frac{TruePos}{TruePos + FalsePos} \quad [3]$$

Sensitivity or Recall

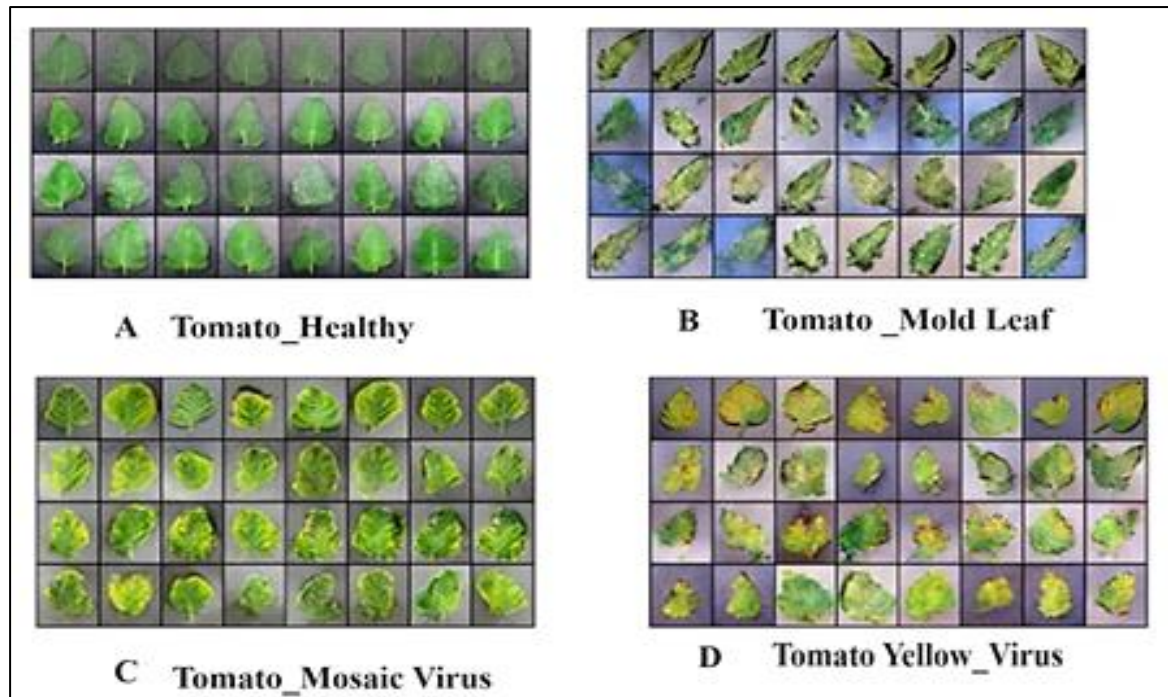
Sensitivity is defined as the percent of correctly forecast leaves with illness compared to the overall instances that were positive of the test case The Equation [4] gives the formula to calculate it.

$$Sensitivity = \frac{TruePos}{TruePos + FalseNeg} \quad [4]$$

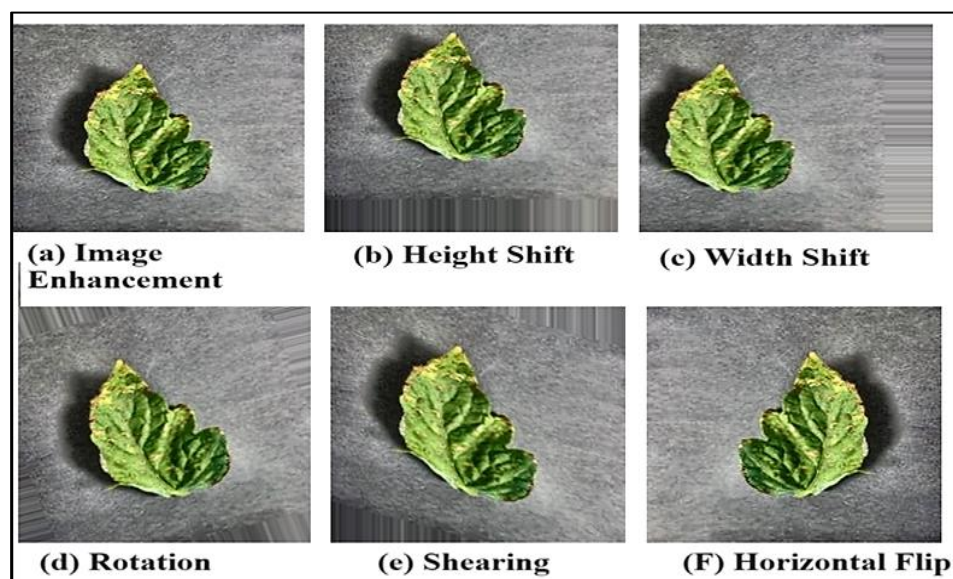
F1-Score

Recall and accuracy can be balanced by one measure dubbed the F1-scoring system, which is the harmonic mean of the two measurements. The Equation [5] gives the formula to calculate it.

$$F1\ score = 2 * \frac{Precision * recall}{Precision + recall} \quad [5]$$

**Figure 8:** Different Type of Tomato Leaf Disease**Table 4:** The Result of the Proposed Method without Argumentation

Method	Accuracy			
	Tomato blight	late mold	Tomato leaf mold	Tomato Mosaic Virus
CNN	0.754	0.762	0.741	0.751
AlexNet	0.768	0.789	0.763	0.773
ResNet	0.787	0.807	0.785	0.790
Inception V3	0.805	0.823	0.802	0.804
VGG-16	0.819	0.838	0.819	0.818

**Figure 9 (a-f):** Image Argumentations

By evaluating these metrics, we can assess the effectiveness and performance of the deep learning model in accurately detecting plant diseases and

distinguishing them from healthy plants and anomalies.

Results without Image Augmentation

The proposed method used metric accuracy for analysis of the experimental results without image argumentation techniques. Following Table 4 shows the result of the proposed method without argumentation.

Argumentation

Augmentation of images analysis was needed in this experiment to ascertain how the volume of data affected the accuracy of the model that was developed. The model's performance was compared to and without augmentation in order to undertake experiments. Tests were carried out using constant hyper parameters. The number of nodes needed for the layer that is hidden is indicated by the hidden units. This unit is in the middle of the number of output and input nodes. Adam is an optimization method that is a development of the recently popularized stochastic gradient descent in deep learning. The CNN algorithm, a transfer-learning method, was used in this investigation. In order to minimize the length of the training process, a low epoch value was chosen. 30,000 images were obtained with image augmentation, compared to around 10,1820 without image augmentation. Models lacking image augmentation scored lower and were consequently less adept at identifying visual patterns since there was less variance in the training data. Each pixel in the image is translated by a constant factor in both the horizontal and vertical directions to achieve height and width shifting. In our instance, a random selection from the range [0, 0.2] was made for the constant factor. The RGB values of the closest pixels are filled in the empty spaces while shifting, discarding the pixels that cross the boundary. The results of performing a width shift and a height shift are shown in Figure 9(b) and 9(c), respectively.

The image's centre pixel is taken into consideration when rotating an image. The rotation angle in our instance was selected at random from a range of [-20, 20] degrees. Rotation has an effect, as seen in Figure 9(d). Shearing is done by using a shearing factor to move each pixel in a fixed direction by a quantity proportional to its

distance from the bottom-most pixels in the image. Random selections were made from the possible values [0, 0.2] for the shearing factor. Figure 9(e) illustrates the effect of shearing. To turn an image horizontally, the pixels must be mirrored in relation to the centroid parallel to the x-axis. Figure 9(f) shows the outcome of flipping something horizontally. Table 5 shows the hyperparameters of Image Argumentation taken for the proposed methods. Table 6 shows the results of healthy tomato leaf classification and its comparison with other models. It can be inferred VGG-16 provide better accuracy compare to other models. Table 7 shows the results of Tomato late blight disease leaf classification and its comparison with other models. It can be inferred VGG-16 provide better accuracy compare to other models. Table 8 shows the results of Tomato mold leaf disease classification and its comparison with other models. Table 9 shows the results of Tomato mosaic virus disease classification and its comparison with other models. Table 10 shows the results of Tomato yellow leaf curl virus classification and its comparison with other models. From the experimental analysis, it found the VGG-16 produce better results compared to other methods. It is better because VGG model has high ability to extract higher level features from images. This leads to higher accuracy. As the number of levels with fewer kernels rose, non-linearity also grew, which is a positive trend in deep learning. Nevertheless, the computationally demanding and vast number of parameters associated with VGG-16 present a drawback. We have compared the VGG16 which provides better accuracy for healthy leave detection with 3 other methods and the result shown in the Table 11.

Table 5: Hyper-parameters of Image Argumentation

Hyper-parameter	
Hidden units	128
Optimizer	Adams
Epoch	40
Batch Size	32

Table 6: Comparison of Different Models for Healthy Tomato Leaf Classification

Model	Accuracy	Sensitivity	Specificity	Precision	F1 Score
CNN	0.865	0.822	0.893	0.885	0.852
AlexNet	0.876	0.841	0.912	0.905	0.872
ResNet	0.898	0.876	0.921	0.917	0.896
Inception V3	0.915	0.892	0.928	0.925	0.908
VGG-16	0.937	0.907	0.947	0.945	0.926

Table 7: Comparison of Different Models for Plant Disease-Tomato Late Blight

Model	Accuracy	Sensitivity	Specificity	Precision	F1 Score
CNN	0.872	0.909	0.888	0.891	0.899
AlexNet	0.899	0.928	0.905	0.907	0.917
ResNet	0.915	0.934	0.917	0.919	0.927
Inception V3	0.927	0.945	0.931	0.932	0.939
VGG-16	0.942	0.956	0.949	0.949	0.951

Table 8: Comparison of Different Models for Plant Disease –Tomato Mold Leaf

Model	Accuracy	Sensitivity (Recall)	Specificity	Precision	F1 Score
CNN	0.856	0.818	0.879	0.872	0.842
AlexNet	0.879	0.832	0.918	0.910	0.868
ResNet	0.904	0.892	0.926	0.924	0.906
Inception V3	0.921	0.908	0.934	0.932	0.919
VGG-16	0.935	0.921	0.946	0.944	0.932

Table 9: Comparison of Different Models for Plant Disease –Tomato Mosaic Virus

Model	Accuracy	Sensitivity	Specificity	Precision	F1 Score
CNN	0.879	0.906	0.890	0.891	0.896
AlexNet	0.900	0.926	0.908	0.910	0.917
ResNet	0.916	0.938	0.919	0.921	0.929
Inception V3	0.928	0.947	0.933	0.933	0.939
VGG-16	0.942	0.958	0.948	0.947	0.948

Table 10: Comparison of Different Models for Plant Disease –Tomato Yellow Leaf Curl Virus

Model	Accuracy	Sensitivity	Specificity	Precision	F1 Score
CNN	0.863	0.817	0.887	0.878	0.843
AlexNet	0.883	0.830	0.920	0.912	0.869
ResNet	0.907	0.897	0.928	0.926	0.911
Inception V3	0.922	0.921	0.935	0.934	0.923
VGG-16	0.966	0.923	0.942	0.941	0.932

Table 11: Comparison of Proposed Method with Existing Works

Reference Paper	Proposed Methods	Accuracy
(12)	ResNet	87.5%
(9)	CNN (8 layers: 2 fully connected, 3 max-pooling, 3 convolutional)	76%-100% (average 91.2%)
(10)	LeNet (CNN variant)	92%
proposed	VGG 16	93.7%

Discussion

In addition, the original dataset has a class imbalance, as was previously mentioned. Diverse approaches have been taken on this problem in the body of current literature. Under sampling and/or oversampling specific classes has been the most popular approach to address the problem. It has its own shortcomings even though it somewhat balances the dataset. It is possible that some of the challenging images for particular classes—which might include important information the model needs to learn—will be eliminated by under sampling. Eventually, this reduces the model's generalizability. By employing diverse data augmentation methods, oversampling generates multiple marginally distinct renditions of the source photos. If augmentation is carried out prior to the dataset being split into train, validation, and test sets, it could, nevertheless, bring minute deviations from the training set into the test set. An overestimation of the system's accuracy will result from the model's propensity to correctly categorize the other variants in the test set as it obtains experience classifying one version of the image during training. We decided to carry out data augmentation while runtime since each choice offers benefits and drawbacks. Accuracy, sensitivity, and specificity were mostly higher across all tasks for models employing augmentation in comparison to those not employing augmentation. Across all tasks, models with and without augmentation were more accurate, sensitive, and specific. In comparison to using or not using augmentation, other models were consistently outperformed by VGG-16 model. However, the difference was more pronounced with augmentation since VGG-16 has shown that it could better capture complex features using additional data. Data augmentation will immensely help in enhancing the model efficacy, this is mostly true where the dataset is limited. Augmentation has an impact of increasing the variability and richness of training data; hence better generalization and robustness in model predictions take place.

Applications and Challenges

In practical terms, our fresh deep computing model for tomatoes leaf illness detection presents both advantages and difficulties in real-world agricultural environments. If the system has been put into location it will be able to spot such

circumstances in their early phases allowing for quick action to be taken to stop the spread and lessen the impact on crop harvests. This will enhance labor and time savings while adhering to precision agriculture practices by providing farmers with practical guidance on enhancing yields through the use of diverse computer devices. To reduce yield loss and choose the optimal agricultural techniques, disease knowledge should be kept on a cloud-based digital platform. This is especially useful in areas where plant diseases result in significant food losses.

Nevertheless, for the computerized system to succeed, there are some problems associated with it that should be resolved. A significant amount of training data is crucial for model performance; however, getting a varied dataset that covers different conditions and diseases is problematic. Connectivity to the internet, computational resources, and IoT devices might be required for successful implementation of technical infrastructures which may pose hurdles, particularly in small-scale or resource-constrained farms. The initial costs as well as ongoing expenses related to this technology could also be too high for certain farmers who may not adopt this technology because they do not trust digital tools or are unfamiliar with them.

Consequently, the model will need to be adjusted and retrained as per various types of tomatoes, growing conditions and new strains of diseases. There is also the issue of regulation and ethics on data privacy that needs to be discussed in terms of technology use in farming. This means that this model should be able to scale up and across different kinds of farming environments through careful consideration for its fit with existing agricultural practices and workflows. However, there are some unsolved problems such as technological collaboration between technologists, agronomists, farmers, investment in infrastructure, training among others all which are vital towards overcoming herculean task barriers or surmounting challenges posed by modern technologies available in agriculture

Future Works

In subsequent research, we plan to create an online tool that will help farmers diagnose illnesses and choose appropriate treatments from a distance, according to the effectiveness findings. In our application, we would employ the VGG16 model

since it offers the most accuracy compared to the other models. Before executing the model to ascertain whether the tomato plants are healthy or unhealthy, we will develop a web application that runs on local host and gathers pictures of plants from clients whether they are diseased or not.

Conclusion

In this work, we combined IoT integration with deep learning models based on transfer learning to forecast several plant diseases. The deep learning models used include VGG-16, Inception V3, AlexNet, ResNet, and CNN. Utilizing pre-trained data from these models has significantly enhanced the accuracy and efficiency of the system while reducing the prediction time for plant disease detection using transfer learning. Integration of Medical IoT with deep learning has facilitated quick and accurate prediction of tomato plant diseases. Among all the models, VGG-16 demonstrated the highest accuracy of 93.7% for Healthy Tomato leaf, 94.2% for Tomato late blight, 93.4% for Tomato mold leaf, 94.2% for Tomato mosaic virus and 96.6% for Tomato yellow leaf curl virus

Abbreviations

IOT: Internet of Things
 CNN: Convolutional Neural Networks
 ResNet: Residual Network
 VGG-16: Visual Geometry Group-16
 SVMs: Support Vector Machines
 ToMV: Tomato Mosaic Virus
 TYLC: Tomato Yellow Leaf Curl Virus
 Xception: Extreme Inception
 HOG: Histogram of an Oriented Gradient
 LVQ: Learning Vector Quantization
 RGB: Red Green Blue
 ANN: Artificial Neural Network
 NLP: Natural Language Processing
 SGD: Stochastic Gradient Descent
 ReLU: Rectified Linear Units
 LRN: Local Response Normalization

Acknowledgement

Nil.

Author Contributions

All Authors contributed entire manuscript in writing, reviewing, implementing, conceptualization and analysis.

Conflict of Interest

The authors declare no conflict of interest.

Ethics Approval

Not applicable.

Funding

No funds received.

References

1. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. *Communications of the ACM*. 2017;60(6):84-90.
2. Tsaftaris SA, Minervini M, Scharr H. Machine learning for plant phenotyping needs image processing. *Trends in Plant Science*. 2016;21(12):989-91.
3. Zhao S, Peng Y, Liu J, Wu S. Tomato leaf disease diagnosis based on improved convolution neural network by attention module. *Agriculture*. 2021;11(7):651.
4. Oerke EC, Dehne HW, Schönbeck F, Weber A. Crop production and crop protection: estimated losses in major food and cash crops. *Elsevier*; 2012; 51 (4):493-495.
5. Panchal AV, Patel SC, Bagyalakshmi K, Kumar P, Khan IR, Soni M. Image-based plant diseases detection using deep learning. *Materials Today: Proceedings*. 2023; 80:3500-6.
6. Masarirambi MT, Mhazo NO, Oseni TO, Shongwe VD. Common physiological disorders of tomato (*Lycopersicon esculentum*) fruit found in Swaziland. *J Agric Soc Sci* 2009;5:123-127.
7. Imran M, Khan MA, Fiaz M, Azeem M, Mustafa M. Influence of environmental conditions on tomato mosaic virus disease development under natural condition. *Pakistan Journal of Phytopathology*. 2013;25(2):117-22.
8. Choi GW, Kim B, Ju H, Cho S, Seo E, Kim J, Park J, Hammond J, Lim HS. Dual infections of Tomato mosaic virus (ToMV) and Tomato yellow leaf curl virus (TYLCV), or Tomato mosaic virus (ToMV) and Tomato chlorosis virus (ToCV), detected in tomato fields located in Chungcheongnam-do in 2017. *Korean Journal of Agricultural Science*. 2018;45(1):38-42.
9. Agarwal M, Singh A, Arjaria S, Sinha A, Gupta S. ToLeD: Tomato leaf disease detection using convolution neural network. *Procedia Computer Science*. 2020; 167:293-301.
10. Tiwari V, Joshi RC, Dutta MK. Dense convolutional neural networks based multiclass plant disease detection and classification using leaf images. *Ecological Informatics*. 2021; 63:101289.
11. Guerrero-Ibañez A, Reyes-Muñoz A. Monitoring tomato leaf disease through convolutional neural networks. *Electronics*. 2023; 12(1):229.
12. Bhise N, Kathet S, Jaiswar S, Adgaonkar A. Plant disease detection using machine learning. *International Research Journal of Engineering and Technology*. 2020; 7(7):2924-9.
13. Saleem MH, Potgieter J, Arif KM. Plant disease detection and classification by deep learning. *Plants*. 2019;8(11):468.
14. Oo YM, Htun NC. Plant leaf disease detection and classification using image processing. *International*

- Journal of Research and Engineering. 2018; 5(9):516-2.
15. Chittabarni S, Deepak G, Umesh G, Barenya BH. Leaf disease detection using machine learning and deep learning: Review and challenges, Applied Soft Computing. 2023; 1:145-51.
 16. Nikhil S. Band. Machine Learning Models for Plant Disease Detection and Classification. International Journal of Intelligent Systems and Applications in Engineering, 2024;12(3): 4115–4121.
 17. Nigam S, Jain R. Plant disease identification using Deep Learning: A review. Indian Journal of Agricultural Sciences. 2020; 90 (2): 249–57.
 18. Litjens G, Kooi T, Bejnordi BE, Setio AA, Ciompi F, Ghafoorian M, Van Der Laak JA, Van Ginneken B, Sánchez CI. A survey on deep learning in medical image analysis. Medical Image Analysis. 2017; 42:60-88.
 19. Mukhopadhyay S, Paul M, Pal R, De D. Tea leaf disease detection using multi-objective image segmentation. Multimedia Tools and Applications. 2021; 80:753-71.
 20. Shah D, Trivedi V, Sheth V, Shah A, Chauhan U. ResTS: Residual deep interpretable architecture for plant disease detection. Inf. Process. Agric. 2022; 9:212–223.
 21. Jung M, Song JS, Shin AY, Choi B, Go S, Kwon SY, Park J, Park SG, Kim YM. Construction of deep learning-based disease detection model in plants. Scientific Reports. 2023; 13(1):7331.
 22. Jackulin C, Murugavalli S. A comprehensive review on detection of plant disease using machine learning and deep learning approaches. Measurement: Sensors. 2022; 24:100441.
 23. Yamashita R, Nishio M, Do RK, Togashi K. Convolutional neural networks: an overview and application in radiology. Insights Into Imaging. 2018; 9:611-29.
 24. Dhingra G, Kumar V, Joshi HD. Study of digital image processing techniques for leaf disease detection and classification. Multimedia Tools and Applications. 2018; 77:19951-20000.
 25. Padshetty S, Ambika. Leaky ReLU-ResNet for Plant Leaf Disease Detection: A Deep Learning Approach. Engineering Proceedings. 2023; 59(1):39.
 26. Chouhan SS, Singh UP, Jain S. Automated plant leaf disease detection and classification using fuzzy based function network. Wireless Personal Communications. 2021; 121(3):1757-79.
 27. Chen HC, Widodo AM, Wisnujati A, Rahaman M, Lin JC, Chen L, Weng CE. AlexNet convolutional neural network for disease detection and classification of tomato leaf. Electronics. 2022; 11(6):951.
 28. Alatawi AA, Alomani SM, Alhawiti NI, Ayaz M. Plant disease detection using AI based vgg-16 model. International Journal of Advanced Computer Science and Applications. 2022; 13(4):718-727.
 29. Lambat RK, Kothari R, Mane MK. Plant disease detection using inceptionv3. International Research Journal of Engineering and Technology (IRJET). 2022; 9(06):2295-2300.
 30. Saritha S, Srinivas VS, Anuhya D, Pavithra G. Performance Analysis of Detection of Disease on Leaf Images with Inception V3 and Mobilenet Deep Learning Techniques. Journal of Pharmaceutical Negative Results. 2022;13(1):111-117.