A

Project Report

on

Comparative analysis of VGG16, Inception V4, AlexNet, and ResNet 50 for Plant Disease Identification

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2023-24

in

Computer Science and Engineering

By

Utkarsh Arora (2000290100177)

Samaira Singh (2000290100130)

Vaibhav Singh (2000290100181)

Under the supervision of

Mr. Upendra Mishra

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

May, 2024

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature

Name: Utkarsh Arora

Roll No.: 2000290100177

Date:

Signature

Name: Samaira Singh

Roll No.: 2000290100130

Date:

Signature

Name: Vaibhav Singh

Roll No.: 2000290100181

Date:

CERTIFICATE

This is to certify that Project Report entitled "Plant Disease Identification using Deep Learning" which is submitted by Utkarsh Arora, Samaira Singh, Vaibhav Singh in partial fulfillment of the requirement for the award of degree B. Tech. in Department of Computer Science & Engineering of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

.

Mr. Upendra Mishra	**Dr. Vineet Sharma**
(Asstt. Professor)	**(Head of Department)**

Date:

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to superisor name, Department of Computer Science & Engineering, KIET, Ghaziabad, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Vineet Sharma, Head of the Department of Computer Science & Engineering, KIET, Ghaziabad, for his full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members, especially faculty/industry person/any person, of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Date:
Signature:
Name : Utkarsh Arora
Roll No.: 2000290100177

Signature:
Name : Samaira Singh
Roll No.: 2000290100130

Signature:
Name : Vaibhav Singh
Roll No.: 2000290100181

ABSTRACT

The increasing incidence of leaf diseases in agricultural crops has necessitated the development of efficient and automated detection methods to safeguard crop health and maximize yield. In the field of artificial intelligence, deep learning has emerged as a major computing paradigm with immense potential for solving computer vision problems. Convolutional Neural Networks (CNNs) are a type of deep learning architecture that provide accurate results for tasks such as image recognition and object detection.

This study utilizes the state-of-the-art ResNet-50 model to identify and classify plant leaf diseases, combined with transfer learning to enhance its accuracy and efficiency. The proposed methodology involves dataset curation, preprocessing, and fine-tuning of the ResNet-50 architecture. To evaluate its performance, we have also tested VGG16, Inception V4, AlexNet, and ResNet-50 models.

The results demonstrate that ResNet-50 outperforms these alternative architectures, achieving an impressive accuracy of 99.7% in identifying and categorizing leaf diseases. This high level of accuracy underscores the effectiveness of the ResNet-50 model in automating the detection of plant leaf diseases, which is critical for maintaining crop health and maximizing agricultural productivity.

By leveraging the power of deep learning, particularly transfer learning, this study presents a robust model capable of accurately classifying plant leaf diseases from high-resolution images. The success of the ResNet-50 model highlights the potential for deep learning technologies to revolutionize agricultural practices, offering reliable and precise tools for early disease detection. This advancement not only helps in preserving crop health but also plays a crucial role in ensuring maximum agricultural yields. As deep learning technologies continue to evolve, their application in agriculture promises innovative solutions to longstanding challenges, paving the way for more sustainable and efficient farming practices.

TABLE OF CONTENTS

Page No.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

NAM	Network Animator
CNN	Convolutional Neural Network
SVM	Support Vector Machine
ELM	Extreme Learning Machine
ANN	Artificial Neural Networks
ReLu	Rectified Linear Unit
VGG	Visual Gemetry Group
RESNET	Residual Networks
RGB	Red-Blue-Green

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

Plant diseases disrupt the regular condition and growth of plants, contributing to reduced productivity and economic losses. Detecting plant diseases is crucial for sustainable agriculture and maximizing crop yield, as disease identification can lead to a yield gain of over 60% of overall productivity. Pesticides can protect plants against diseases or infections, but they have detrimental effects on the environment, biodiversity, and human health. Knowledge of a field's phytosanitary conditions helps farmers implement appropriate practices in impacted areas at the necessary time.

Multiple methods exist for identifying plant disorders, but most disorders manifest observable symptoms that can be largely assessed by skilled professionals. A computerized method capable of identifying disease-affected plants based on their fundamental symptoms and appearance simplifies the work of disease diagnosis and improves accuracy. Early detection not only safeguards farmers' livelihoods by preserving crop productivity but also contributes to the overall resilience of the agricultural sector, tackling issues presented by changing plant diseases and climate variations. Recent progress in technology and the affordability of inexpensive devices for capturing photos have enabled the collection of a large number of photographs for image-based diagnosis. However, compressed information found in digital photographs makes it difficult for computing systems to examine. Advancements in computer vision and artificial intelligence have created a platform that allows for more accurate identification of plant diseases and presents an opportunity to connect with precision agriculture. Deep learning enables machines to autonomously discover the best combination of attributes for every domain without the need for human assistance.

The emergence of Convolutional Neural Networks (CNNs) was driven by the demanding nature of machine vision tasks. CNN has proven to be the most effective learning algorithms, particularly for tasks involving picture comprehension. The utilization of CNNs for the analysis of plant diseases has been carried out since its beginning. However, using CNNs for plant leaf disease identification presents several challenges, including data, model complexity, computational requirements, and practical deployment. High-quality labelled datasets are crucial for CNN training, but collecting them can be time-consuming and labor-intensive, leading to poor model performance and generalization issues. Overfitting is a common issue in deep learning models, particularly CNNs, where the model learns too well from limited or unrepresentative training data, capturing noise and details.

Implementing CNNs for real-time disease detection in agriculture is challenging due to the need for efficient processing, low latency, and high accuracy, requiring a balance between performance and computational constraints. The plant disease dataset differs from other picture datasets in terms of its size and the variation of features that need to be extracted for classification. CNN obviates the necessity for manually designed features, enhancing the resilience of plant disease classification models compared to conventional ML models.

The motivation for proposing this research writing over previous studies in this area is twofold: 1) This research offers great promise for a more accurate and reliable diagnosis of Plant disease identification using VGG16, Inception V4, AlexNet, and ResNet 50, and 2) the ResNet-50 model exhibits superior accuracy among all of them.

1.2 PROJECT DESCRIPTION

Our project utilizes the state-of-the-art ResNet-50 model, a deep convolutional neural network (CNN) architecture that was introduced by researchers at Microsoft Research, to identify and classify plant leaf diseases, combined with transfer learning to enhance its accuracy and efficiency. It is a deep network with 50 layers. It is a powerful and widely-used deep learning model that leverages residual learning to improve the training and performance of very deep networks, making it highly effective for a range of computer vision applications.It is composed of several residual blocks, which are the fundamental building units of the network.

The innovation of ResNet lies in the use of residual blocks. Each block has a shortcut connection that skips one or more layers, allowing the network to learn residual functions with reference to the layer inputs. This helps mitigate the vanishing gradient problem, allowing for the training of much deeper networks. ResNet-50 has demonstrated high accuracy on image classification tasks. Its design helps avoid overfitting and allows for the training of very deep networks without performance degradation.By leveraging the power of deep learning, particularly transfer learning, this study presents a robust model capable of accurately classifying plant leaf diseases from high-resolution images.

1. **Data Collection**:

    * Utilize the Plant Village dataset from Kaggle, which contains approximately 70,000 images of healthy and diseased plants, spanning 38 categories.

    * Address dataset imbalance by implementing strategies to ensure an equal number of samples per category, focusing on different stages of leaf health.

2. **Data Preprocessing**:

    * Perform data augmentation to enhance the diversity of the training dataset.

    * Normalize the images and resize them to fit the input requirements of the ResNet-50 model.

3. **Model Architecture**:

    * Implement the ResNet-50 architecture, known for its depth and use of residual blocks to mitigate the vanishing gradient problem.

    * Employ transfer learning to fine-tune the pre-trained ResNet-50 model on the Plant Village dataset.

4. **Training and Validation**:

    * Split the dataset into an 80-20 ratio for training and testing purposes.

    * Train the ResNet-50 model using the curated dataset and validate its performance using the reserved test set.

5. **Performance Evaluation**:

    * Evaluate the model's accuracy, precision, recall, and F1 score.

    * Compare the performance of ResNet-50 with VGG16, Inception V4, and AlexNet models to establish its superiority in plant disease identification.

The ResNet-50 model achieved a remarkable accuracy of 99.7% in identifying and classifying plant diseases. This demonstrates the model's effectiveness and potential for real-world applications in agriculture. This project successfully developed an automated system for plant disease identification using the ResNet-50 architecture. The high accuracy of the model underscores its potential to assist farmers and agricultural professionals in early disease detection, leading to better crop management and improved yields. The findings highlight the significant impact of deep learning in advancing agricultural technologies and addressing critical challenges in plant health monitoring.

CHAPTER 2

LITERATURE REVIEW

In the field of agricultural production, ignoring the early signs of plant disease may lead to losses in food crops, which could eventually destroy the world's economy [1]. This section presents an in-depth survey of state-of-the-art research in the field of leaf disease identification.

A CNN-based deep learning model was proposed for the accurate classification of plant disease in [2], and the model was trained using a publicly available dataset with 87,000 RGB images. Initially, preprocessing was undertaken, followed by segmentation. For classification, a CNN was used. Although this model attained a recognition accuracy of 93.5%, it failed to classify some classes, leading to confusion with the classes in subsequent stages. Further, the performance of the model deteriorated due to limited availability of data. However to improve recognition accuracy, Narayanan et al. [3] proposed a hybrid convolutional neural network to classify banana plant disease. In their approach, the raw input image was preprocessed without altering any default information, and the standard image dimensions were maintained using a median filter. This approach used a fusion SVM along with a CNN. A multiclass SVM was used in the testing phase to identify the type of infection or disease in infected banana leaves, whereas the SVM was used in phase 1 to classify whether the banana leaves were healthy or infected. The classified CNN output was fetched as an input to the support vector machine, attaining a classification accuracy of 99%. The previous work stated that the CNN had better accuracy outcomes than traditional methods but this approach lacked diversity.

Jadhav et al. [4] proposed a CNN for the identification of plant disease. In this approach, they used pre-trained CNN models to identify diseases in soybean plants. The experiments were carried out using pre-trained transfer learning approaches, such as AlexNet and GoogleNet, and attained better outcomes, but the model fell behind in the diversity of classification. Many existing models focus on identifying single classes of plant disease rather than building a model to classify various plant diseases. This is mainly due to the limited databases for training deep learning models with diversified plant species.

Jadhav et al. [5] were the first to propose a novel histogram transformation approach, which enhanced the recognition accuracy of deep learning models by generating synthetic image samples from low-quality test set images. The motive behind this work was to enhance the images in the cassava leaf disease dataset using Gaussian blurring, motion blurring, resolution down-sampling, and over-exposure with a modified MobileNetV2 neural network model. In their approach, synthetic images using modified color value distributions were generated to address the data shortage that a data-hungry deep-learning model faces during its training phase and achieve better outcomes.

Following Olusola et al., Abbas et al. [6], in their work proposed, a conditional generative adversarial network to generate a database of synthetic images of tomato plant leaves. With the advent of generative networks, previously expensive, time-consuming and laborious real-time data acquisition or data collection have become possible. Anh et al. [7] proposed a benchmark dataset-based multi-leaf classification model using a pre-trained MobileNet CNN model and found it efficient in classification, attaining a reliable accuracy of 96.58%. Further, a multi-label CNN was put forward in for the classification of multiple plant diseases using transfer learning approaches, such as DenseNet, Inception, Xception, ResNet, VGG, and MobileNet, and the authors claim that theirs' is the first research work that classifies 28 classes of plant disease using a multi-label CNN. Classification of plant diseases using the Ensemble Classifier was proposed in [8]. The best ensemble classifier was evaluated with two datasets; namely, PlantVillage and Taiwan Tomato Leaves. Pradeep et al. [9] proposed the EfficientNet model using a convolutional neural network for multi-label and multi-class classification. The secret layer network in the CNN had a better impact on the identification of plant diseases. However, the model underperformed when validated with benchmark datasets. An effective, loss-fused, resilient convolutional neural network (CNN) was proposed in [10] using the publicly available benchmark dataset PlantVillage and achieved a classification accuracy of 98.93%. Though this method improved the classification accuracy, the model lagged in its performance when using real-time images under different environmental conditions. Later, Enkvetchakul and Surinta [11] proposed a CNN network with a transfer learning approach for two plant diseases. NASMobileNet and MobileNetV2 were the two pre-trained network models used for the classification of plant diseases, among which the most accurate prediction outcome was that based on the NASMobileNet algorithm. Overfitting in deep learning can be resolved using the data augmentation approach. The data augmentation technique was implemented in an experimental setup that included cut-out, rotation, zoom, shift, brightness, and mix-up. Leaf disease datasets and iCassava 2019 were the two kinds of dataset used. The maximum test accuracy attained after the evaluation was 84.51%. Table 1 shows the different convolutional neural network models that have been proposed to improve accuracy.

Various researchers used CNN based models for solving complex tasks. Sibiya M. et al [12] used CNN for the diseases classification in plants of maize. They used histogram techniques to show the model impact. They were able to achieve overall model accuracy 92.85%. Zhang K. et al [13] applied CNN architectures AlexNet, ResNet and GoogleNet for identifying the tomato leaf diseases. ResNet outperformed over other networks with the highest accuracy of 92.28%. In the paper presented by Amara J. et al [14], LeNet architecture was used to detect the banana leaf diseases. Here, authors used the CA and F1-score to evaluate the model using graysacle image and color image. Konstantinos P. Ferentinos [15] compared the classification accuracy of the leaf disease using AlexNet, GoogleNet and VGG CNN architecture. The VGG outperformed then all other networks with the plant, disease performance reaching to 99.53%. Türkoglu M. et al [16] classified 8 different plant disease using different classifiers. They considered KNN, SVM and ELM combined with the features obtained from the state-of-the-art deep learning models. They considered various CNN models like ResNet-50 and 101, InceptionResNetv2 and Inception-v3 and other models. The ResNet-50 with SVM provided best result evaluated using different performance metrics. Amanda Ramcharan et al [17] used Inception-V3 for the detection of cassava disease and they are able to achieve average accuracy around 95% with six classes of the disease. Fujita E [18] used the two different variations of CNN and achieved the 82.3% accuracy in classification of the cucumber plant diseases. The tomato disease classification using CNN was performed by Yamamoto K et al using highresolution, super-resolution and low-resolution to evaluate super-resolution accuracy over other methods [19]. Results in the paper indicated super-resolution method outperformed conventional methods with large margin in term of accuracy. Durmu s H et al [20] presented classification of tomato plant disease using pre-trained nets AlexNet and SqueezNet V1.1. However, performance of AlexNet outperforms with the accuracy of the 95.65% in disease classification. Edna Chebet Too and et al in [21] presented the comparison deep learning nets VGG 16, Inception-V4, ResNet-50, ResNet-101, ResNet152 and DenseNet-121 for leaf disease classification. The results in the papers presented that DenseNet requires less number of parameters compare to other models and achieved the accuracy of 99.75%. Rangarajan A.K et al [22] were performed classification of tomato leaf diseases using AlexNet and VGG-16 deep learning architectures, and AlexNet provided best accuracy 96.38% with minibatch size of 32 and learning rate of 40. Brahimi M. et al [23] presented saliency map for symptoms visualization of plant disease. The accuracy achieved by the proposed architecture was 99.76%. Sladojevic S. et al [24] identified different types of 13 diseases with the help of CaffeNet CNN model and obtained 96.30% classification accuracy which was better compared to typical machine learning algorithm like SVM. Sharada P. Mohanty et al in [25] compared performance of the two CNN architectures i.e. AlexNet and GoogleNet on PlantVillage dataset of leaf diseases. The performance measures considered were precision, F1 score, recall and accuracy of the model. They have done implementation on three scenario i.e. color, grayscale and segmented images for measuring the CNN performance where they found that GoogleNet outperformed AlexNet.

Based on the review conducted, it has been found that, for end to end learning, in many domains deep neural networks applied successfully. It provides mapping between an image of diseased leaf (input) to crop-disease pair (output). The major challenge associated with creating deep neural network is that structure of the network where it is essential to correctly map nodes and edge weights from the input to the output. Deep neural network training has been done by fine tuning the network parameters using process that mapping between input and output layer and it improves during the training process. This challenging process improved dramatically by various conceptual and engineering breakthroughs in recent times. To develop deep neural network based accurate plant disease diagnosis model, large and confirmed dataset of images of healthy and diseased plants are needed. These large dataset was not available until very recently, and even dataset with few images were not also freely available. But in 2015, the project PlanVillage has begun and provided thousands of images of diseased and healthy crops plants openly and freely. However, after this many dataset were introduced considering PlantVillage as a base.

CHAPTER 3

PROPOSED METHODOLOGY

3.1 DATA COLLECTION

The Plant Village dataset is a publicly available collection of images specifically designed for research in the field of deep learning-based plant disease identification. It is a fundamental process in systematic research, essential for gathering information to answer esearch questions, test hypothesis, and evaluate outcomes. The accuracy , integrity , and deoth of any research largely depend on the quality of the data collected, here's a general description based on common practices for creating such datasets:

Figure 1 Sample Images from the Plant Village Dataset

3.1.1 IMAGE CAPTURE

- **Controlled Environment**: The images likely originate from a controlled environment, such as a greenhouse or laboratory setting. This helps minimize background variations and ensure consistent lighting conditions, leading to higher quality image data.
- **Image Acquisition Equipment**: Standardized digital cameras with appropriate resolutions and lighting setups are likely used to capture the plant images. This ensures consistency in image quality and minimizes variations due to camera equipment.

3.1.2 IMAGE LABELING

Domain experts (e.g., plant pathologists, agricultural scientists) likely play a crucial role in meticulously labeling each image. These labels typically include:

- **Plant Species**: The specific type of plant depicted in the image (e.g., tomato, pepper, corn).
- **Disease Status**: Whether the plant is healthy or infected with a particular disease. If diseased, the specific disease type is likely also labeled.
- **Growth Stage**: The developmental stage of the plant (e.g., seedling, vegetative, flowering) might be included depending on the research focus.

3.1.3 DATA SPLITTING

Once captured and labeled, the image dataset is typically split into distinct subsets for training, validation, and testing purposes:

- **Training Set**: The largest portion of the data used to train the deep learning model. The model learns to identify patterns and relationships between image features and corresponding labels within this set.
- **Validation Set**: A smaller subset used to monitor the model's performance during the training process. This helps prevent overfitting and allows for adjustments to the model's hyperparameters.
- **Test Set**: This unseen data represents real-world scenarios and is used to evaluate the final performance of the trained model on unseen plant disease identification tasks.

3.1.4 LIMITATIONS OF DATASET

It's important to acknowledge potential limitations associated with the data collection process:

- **Controlled Environment Bias**: Since images are captured in a controlled setting, the dataset might not fully generalize to real-world field conditions with diverse lighting, weather variations, and background complexities.

- **Limited Disease Coverage**: The dataset might focus on a specific set of plant diseases, and the generalizability to a broader range of diseases might need to be evaluated.

3.2 DATA PREPROCESSING

Deep learning models have achieved remarkable success in various image classification tasks, including plant disease identification. A critical step in this process is data preprocessing, which transforms raw data into a format suitable for model training and improves the model's ability to learn meaningful patterns. This section details the essential data preprocessing techniques employed for plant disease identification using the Plant Village dataset. It is a crucial step in the machine learning pipeline, essential for transforming raw data into a clean dataset that a model can learn from more effectively. This step typically involves several sub-processes aimed at making the data more suitable for modelling and improving the quality and predictive power of the data. Common data preprocessing techniques that are followed include Data cleaning, Data transformation, Data reduction, Handling imbalanced data.

3.2.1 DATA CLEANING

The Plant Village dataset primarily consists of images, and missing values are less common. However, some images might be corrupted, contain missing regions, or exhibit poor quality (e.g., excessive noise, blur). For small datasets, manual inspection of outliers is recommended to determine their validity and potential removal. Alternatively, image thresholding techniques can be implemented to automatically identify and remove outliers based on predefined noise or blur thresholds.

Data consistency is crucial for deep learning models. The Plant Village dataset likely provides images in a common format (e.g., JPEG, PNG). However, resizing all images to a uniform dimension is often necessary to ensure compatibility with the chosen deep learning architecture.

3.2.2 DATA TRANSFORMATION

Normalization is a widely used technique for image data in deep learning. It scales pixel intensities to a specific range (e.g., 0-1 or -1 to 1). This improves the training convergence and stability of the deep learning model. Common normalization techniques include Min-Max Scaling and Z-score normalization.

The Plant Village dataset likely uses labels indicating plant species and disease types. If these labels are stored as text data, they need to be converted into a numerical representation suitable for deep learning models. One-Hot Encoding is a common approach that transforms each categorical label into a separate binary feature vector.

3.2.3 DATA AUGMENTATION

Data augmentation is particularly valuable for enhancing the size and diversity of a dataset, especially for image data like the Plant Village dataset. This is achieved by applying various transformations to existing images, effectively creating new training examples without requiring additional data collection. These are some of data augmentation techniques specifically suited for plant disease identification:

- **Random Cropping**: Extracting random sub-regions from the original images allows the model to learn features relevant to disease identification regardless of the object's position within the image frame.
- **Rotation**: Randomly rotating images helps the model generalize to different plant orientations, improving its ability to identify diseases even when the leaf appears at an angle.
- **Flipping**: Flipping images horizontally or vertically increases the model's robustness to variations in leaf orientation, ensuring it can recognize diseases irrespective of the leaf's side facing the camera.
- **Color Jittering**: This technique involves randomly adjusting image properties like brightness, contrast, saturation, and hue. It helps the model become more robust to variations in lighting conditions encountered in real-world scenarios.

By applying these data augmentation techniques, we can significantly increase the size and diversity of the training data, leading to a more robust and generalizable model for plant disease identification.

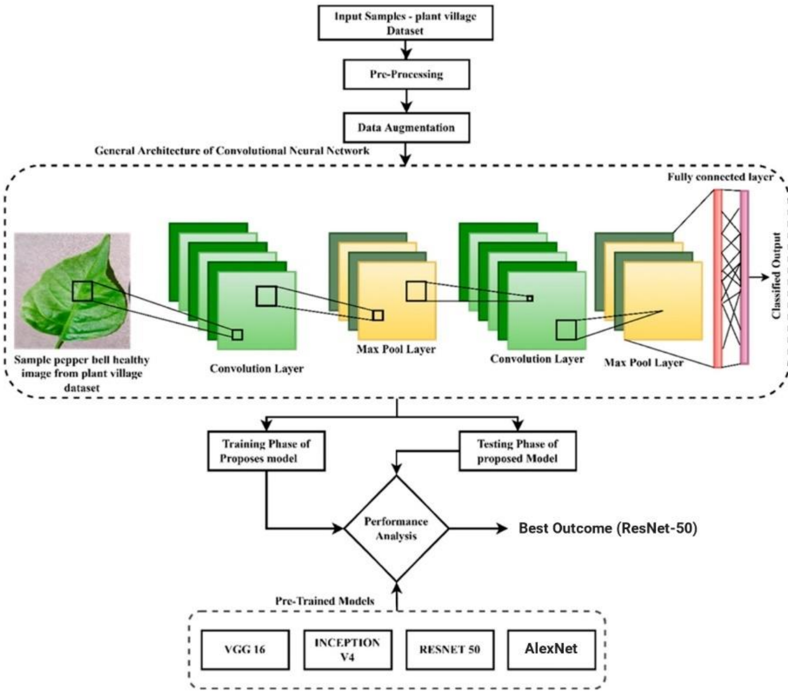


Figure. 2 Data Augmentation schematic diagram

Additional Considerations:

- **Class Imbalance:** The Plant Village dataset might exhibit an imbalance between healthy and diseased plant classes. Techniques like oversampling (replicating minority class images) or undersampling (carefully removing majority class images) can be explored to address this imbalance and improve model performance.
- **Image Preprocessing Libraries:** Deep learning frameworks like TensorFlow offer functionalities through libraries such as `tensorflow.keras.preprocessing.image` to streamline the data preprocessing pipeline. These libraries provide convenient functions for image resizing, normalization, and data augmentation.

Effective data preprocessing is essential for preparing the Plant Village dataset for deep learning-based plant disease identification. By employing data cleaning techniques to ensure data quality and consistency, data transformation for compatibility with the model, and data augmentation to enrich the training data, we can significantly improve the model's ability to learn robust and generalizable features for accurate plant disease classification.

3.3 CONVOLUTIONAL NEURAL NETWORK

Convolutional neural networks use three-dimensional data to for image classification and object recognition tasks. Neural networks are a subset of machine learning, and they are at the heart of deep learning algorithms. They are comprised of node layers, containing an input layer, one or more hidden layers, and an output layer. Each node connects to another and has an associated weight and threshold. If the output of any individual node is above the specified threshold value, that node is activated, sending data to the next layer of the network. Otherwise, no data is passed along to the next layer of the network.

Convolutional neural networks are distinguished from other neural networks by their superior performance with image, speech, or audio signal inputs.

They have three main types of layers which are stated as follows :

- **Convolutional layer**
- **Pooling layer**
- **Fully-connected (FC) layer**

The convolutional layer is the first layer of a convolutional network. While convolutional layers can be followed by additional convolutional layers or pooling layers, the fully-connected layer is the final layer. With each layer, the CNN increases in its complexity, identifying greater portions of the image. Earlier layers focus on simple features, such as colours and edges.

As the image data progresses through the layers of the CNN, it starts to recognize larger elements or shapes of the object until it finally identifies the intended object.

3.3.1 CONVOLUTIONAL LAYER

The convolutional layer is the core building block of a CNN, and it is where the majority of computation occurs. It requires a few components, which are input data, a filter, and a feature map. Let’s assume that the input will be a colour image, which is made up of a matrix of pixels in 3D. This means that the input will have three dimensions—a height, width, and depth—which correspond to RGB in an image. We also have a feature detector, also known as a kernel or a filter, which will move across the receptive fields of the image, checking if the feature is present. This process is known as a convolution. The feature detector is a two-dimensional (2-D) array of weights, which represents part of the image. While they can vary in size, the filter size is typically a 3x3 matrix; this also determines the size of the receptive field. The filter is then applied to an area of the image, and a dot product is calculated between the input pixels and the filter. This dot product is then fed into an output array. Afterwards, the filter shifts by a stride, repeating the process until the kernel has swept across the entire image. The final output from the series of dot products from the input and the filter is known as a feature map, activation map, or a convolved feature.

Note that the weights in the feature detector remain fixed as it moves across the image, which is also known as parameter sharing. Some parameters, like the weight values, adjust during training through the process of backpropagation and gradient descent. However, there are three hyper- parameters which affect the volume size of the output that need to be set before the training of the neural network begins. These include:

1) The number of filters affects the depth of the output. For example, three distinct filters would yield three different feature maps, creating a depth of three.
2) Stride is the distance, or number of pixels, that the kernel moves over the input matrix. While stride values of two or greater is rare, a larger stride yields a smaller output.
3) Zero-padding is usually used when the filters do not fit the input image. This sets all elements that fall outside of the input matrix to zero, producing a larger or equally sized output. There are three types of padding:
    - **Valid padding:** This is also known as no padding. In this case, the last convolutionis dropped if dimensions do not align.
    - **Same padding:** This padding ensures that the output layer has the same size as theinput layer.
    - **Full padding:** This type of padding increases the size of the output by adding zerosto the border of the input.

After each convolution operation, a CNN applies a Rectified Linear Unit (ReLU) transformation to the feature map, introducing nonlinearity to the model.

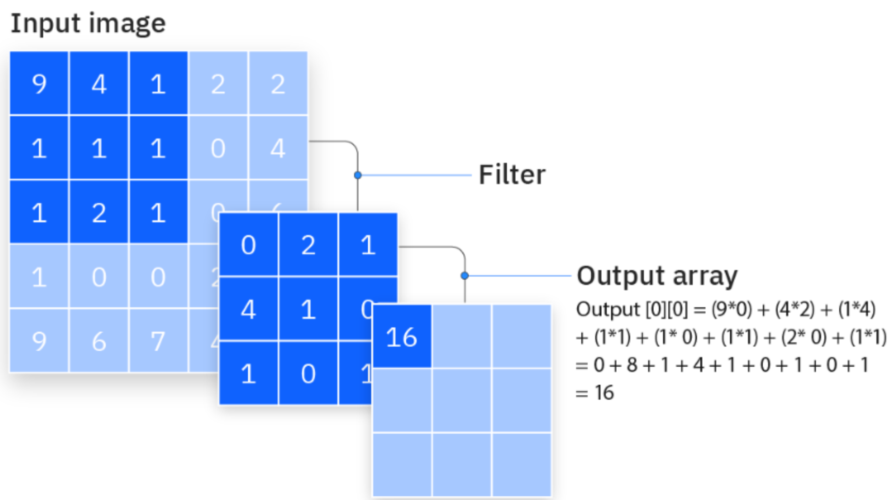


Figure 3 Convolution Layer

As we mentioned earlier, another convolution layer can follow the initial convolution layer. When this happens, the structure of the CNN can become hierarchical as the later layers can see the pixels within the receptive fields of prior layers. As an example, let's assume that we're trying to determine if an image contains a bicycle. You can think of the bicycle as a sum of parts. It is comprised of a frame, handlebars, wheels, pedals, et cetera. Each individual part of the bicycle makes up a lower- level pattern in the neural net, and the combination of its parts represents a higher-level pattern, creating a feature hierarchy within the CNN. Ultimately, the convolutional layer converts the image into numerical values, allowing the neural network to interpret and extract relevant patterns.

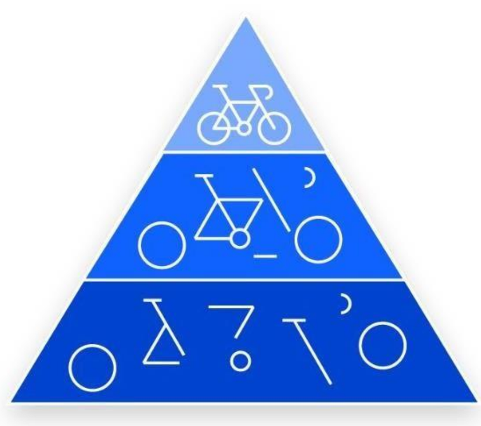


Figure 4 Feature Hierarchy

3.3.2 POOLING LAYER

Pooling layers, also known as down-sampling, conducts dimensionality reduction, reducing the number of parameters in the input. Similar to the convolutional layer, the pooling operation sweeps a filter across the entire input, but the difference is that this filter does not have any weights. Instead, the kernel applies an aggregation function to the values within the receptive field, populating the output array. There are two main types of pooling:

• **Max pooling**: As the filter moves across the input, it selects the pixel with the maximum value to send to the output array. As an aside, this approach tends to be used more often compared to average pooling.

• **Average pooling**: As the filter moves across the input, it calculates the average value within the receptive field to send to the output array.

While a lot of information is lost in the pooling layer, it also has a number of benefits to the CNN. They help to reduce complexity, improve efficiency, and limit risk of overfitting.

3.3.3 FULLY-CONNECTED LAYER

The name of the full-connected layer aptly describes itself. As mentioned earlier, the pixel values of the input image are not directly connected to the output layer in partially connected layers. However, in the fully-connected layer, each node in the output layer connects directly to a node in the previous layer. This layer performs the task of classification based on the features extracted through the previous layers and their different filters. While convolutional and pooling layers tend to use ReLu functions, FC layers usually leverage a softmax activation function to classify inputs appropriately, producing a probability from 0 to 1.

3.4 TRANSFER MACHINE LEARNING

Humans possess a remarkable ability to apply knowledge gained from previous tasks to new challenges. This means that when faced with unfamiliar problems, we can swiftly recognize them and leverage relevant insights acquired from past experiences. This facilitates efficient and effective task completion. For instance, if someone can ride a bicycle and is then asked to operate a motorbike, their proficiency in cycling will prove beneficial. They can leverage their understanding of balance and steering, easing the learning curve compared to starting from scratch. Such lessons learned in various situations are invaluable in real-life scenarios, enabling continual improvement and accumulation of expertise.

Similarly, in the realm of machine learning, Transfer Learning adopts a comparable strategy. It entails utilizing knowledge obtained from one task to address challenges in another task. While many machine learning algorithms are typically tailored for specific tasks, there's a growing interest in developing methods for transfer learning.

A notable characteristic observed in many deep neural networks designed for image processing is their adeptness at identifying edges, colours, intensity variations, and other features in the initial layers. These features are not tied to any specific dataset or task. Whether discerning lions or cars, these fundamental features must be recognized. They persist irrespective of the image data or objective function used. These features, acquired in tasks like lion detection, can also be leveraged for human detection. Transfer learning encapsulates this concept precisely. Nowadays, it's uncommon to encounter individuals training entire convolutional neural networks from scratch. Instead, relying on pre-trained models trained on diverse image sets, such as ImageNet (comprising 1.2 million images spanning 1000 categories), and then repurposing these features to tackle new tasks, is the prevailing practice.

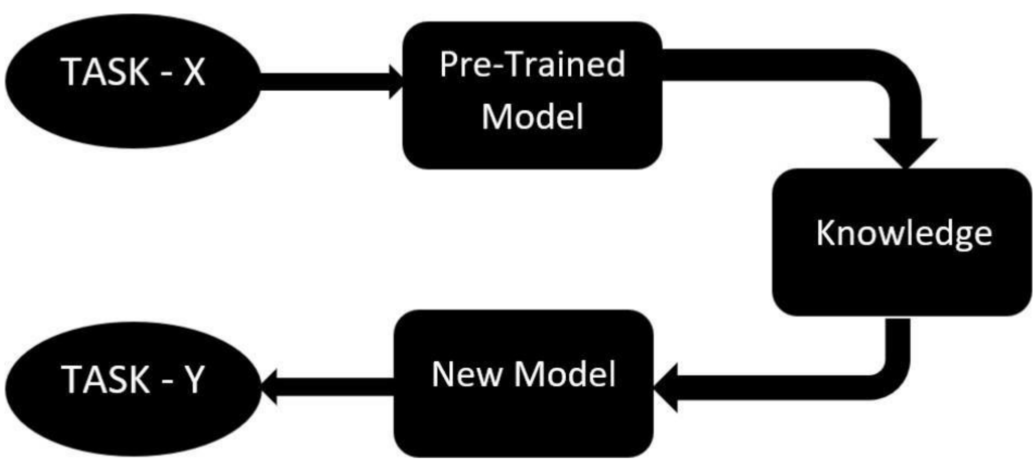


Figure 5 Flow Diagram of Transfer Machine Learning

3.5 RESNET 50

ResNet stands for Residual Network and is a specific type of convolutional neural network (CNN) introduced in the 2015 paper "Deep Residual Learning for Image Recognition" by He Kaiming, Zhang Xiangyu, Ren Shaoqing, and Sun Jian. CNNs are commonly used to power computer vision applications. ResNet-50 is a 50-layer convolutional neural network (48 convolutional layers, one MaxPool layer, and one average pool layer). Residual neural networks are a type of artificial neural network (ANN) that forms networks by stacking residual blocks. The original ResNet architecture was ResNet-34, which comprised 34 weighted layers. It provided a novel way to add more convolutional layers to a CNN, without running into the vanishing gradient problem, using the concept of shortcut connections. A shortcut connection "skips over" some layers, converting a regular network to a residual network. The regular network was based on the VGG neural networks (VGG-16 and VGG-19)—each convolutional network had a 3×3 filter. However, a ResNet has fewer filters and is less complex than a VGGNet. A 34-layer ResNet can achieve a performance of 3.6 billion FLOPs, and a smaller 18-layer ResNet can achieve 1.8 billion FLOPs, which is significantly faster than a VGG-19 Network with 19.6 billion FLOPs (read more in the ResNet paper, He et, al, 2015). The ResNet architecture follows two basic design rules. First, the number of filters in each layer is the same depending on the size of the output feature map. Second, if the feature map's size is halved, it has double the number of filters to maintain the time complexity of each layer.

Special characteristics of ResNet-50

ResNet-50 has an architecture based on the model depicted above, but with one important difference. The 50-layer ResNet uses a bottleneck design for the building block. A bottleneck residual block uses 1×1 convolutions, known as a "bottleneck", which reduces the number of parameters and matrix multiplications. This enables much faster training of each layer. It uses a stack of three layers rather than two layers.

The 50-layer ResNet architecture includes the following elements, as shown in the table below:

- **A 7×7 kernel convolution** alongside 64 other kernels with a 2-sized stride.
- **A max pooling layer** with a 2-sized stride.

- **9 more layers**—3×3, 64 kernel convolution, another with 1×1, 64 kernels, and a third with1×1,256 kernels. These 3 layers are repeated 3 times.
- **12 more layers** with 1×1,128 kernels, 3×3,128 kernels, and 1×1,512 kernels, iterated 4times.
- **18 more layers** with 1×1,256 cores, and 2 cores 3×3,256 and 1×1, 1024, iterated 6 times.
- **9 more layers** with 1×1,512 cores, 3×3,512 cores, and 1×1, 2048 cores iterated 3 times.(up to this point the network has 50 layers)
- **Average pooling**, followed by a fully connected layer with 1000 nodes, using the softmaxactivation function.

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 6 ResNet50 Convolution Layers

3.6 INCEPTION V4

The Inception-v4 model is a deep convolutional neural network (CNN) architecture that builds upon the Inception architecture introduced in the "Going Deeper with Convolutions" paper by Christian Szegedy et al. The Inception-v4 model was proposed by Szegedy et al. in collaboration with other researchers at Google, and it represents an evolution of the original Inception architecture with improved performance and efficiency.

The Inception-v4 architecture incorporates several design principles aimed at improving the model's accuracy and efficiency. These include:

1) **Inception modules**: The Inception-v4 model features updated versions of the original Inception modules, which consist of multiple parallel convolutional operations with different filter sizes. These modules allow the network to capture features at multiple scales and resolutions, enhancing its representational power.
2) **Factorized convolutions**: To reduce computational complexity and improve efficiency, the Inception-v4 model utilizes factorized convolutions, which decompose standard convolutions into smaller, more computationally efficient operations.
3) **Reduction blocks**: In addition to standard convolutional layers, the Inception-v4 model includes reduction blocks that incorporate max pooling and stride convolutions to reduce spatial dimensions and computational burden while preserving important features.
4) **Residual connections**: Inspired by the success of residual networks (ResNets), the Inception-v4 model incorporates residual connections within its architecture. These connections enable more efficient gradient flow during training and facilitate the training of deeper networks.

The Inception-v4 model has been pre-trained on large-scale image datasets such as ImageNet, allowing it to learn rich and generalizable representations of visual features. This pre-training enables transfer learning, where the pre-trained Inception-v4 model can be fine-tuned on smaller datasets or specific tasks with relatively few additional training samples.

Overall, the Inception-v4 model represents a state-of-the-art CNN architecture for image classification and related computer vision tasks, combining innovative design principles with efficient implementation to achieve high accuracy and performance.

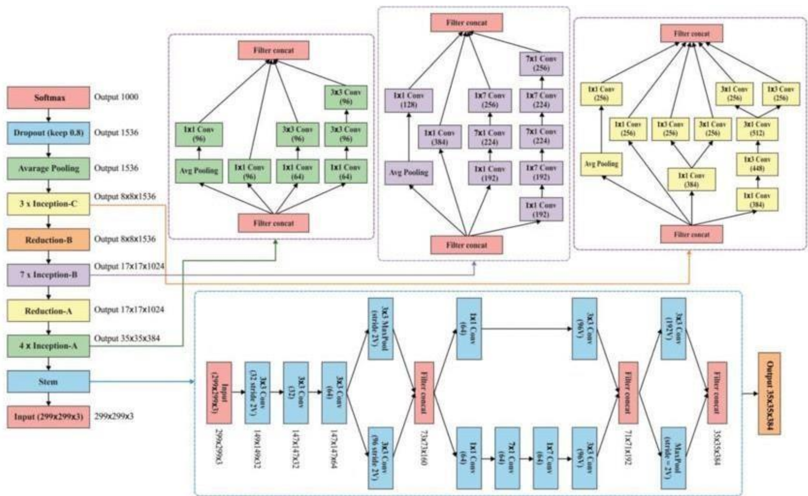


Figure 7 Inception v4 Architecture

3.7 VGG-16

The VGG-16 model is a convolutional neural network (CNN) architecture proposed by Karen Simonyan and Andrew Zisserman from the Visual Geometry Group (VGG) at the University of Oxford. It is characterized by its depth, consisting of 16 layers, including 13 convolutional layers and 3 fully connected layers. The architecture of the VGG-16 model is known for its simplicity and uniformity, with small 3x3 convolutional filters used throughout the network.

The key features of the VGG-16 model include:

1) **Depth**: The VGG-16 model is relatively deep compared to earlier CNN architectures. Its depth allows it to capture complex hierarchical features from input images, leading to improved performance on tasks such as image classification.
2) **Small Convolutional Filters**: The VGG-16 model uses small 3x3 convolutional filters for all its convolutional layers. This uniformity in filter size helps maintain a simple and streamlined architecture, making it easier to understand and implement.

3) **Pooling Layers**: The VGG-16 model employs max pooling layers after every two convolutional layers, reducing spatial dimensions and extracting dominant features from feature maps.
4) **Fully Connected Layers**: The VGG-16 model concludes with three fully connected layers followed by a softmax layer for classification. These fully connected layers integrate high-level features extracted by earlier layers and perform the final classification task.
5) **ReLU Activation**: Rectified Linear Unit (ReLU) activation functions are used throughout the VGG-16 model, providing non-linearity and enabling the network to learn complex features.
6) **Pre-training**: The VGG-16 model has been pre-trained on large-scale image datasets such as ImageNet. This pre-training allows the model to learn generic features from a diverse range of images, making it suitable for transfer learning on various computer vision tasks.

Despite its simplicity, the VGG-16 model has shown impressive performance on tasks such as image classification, object detection, and image segmentation. Its straightforward architecture and effectiveness have made it a popular choice in the field of deep learning and computer vision.

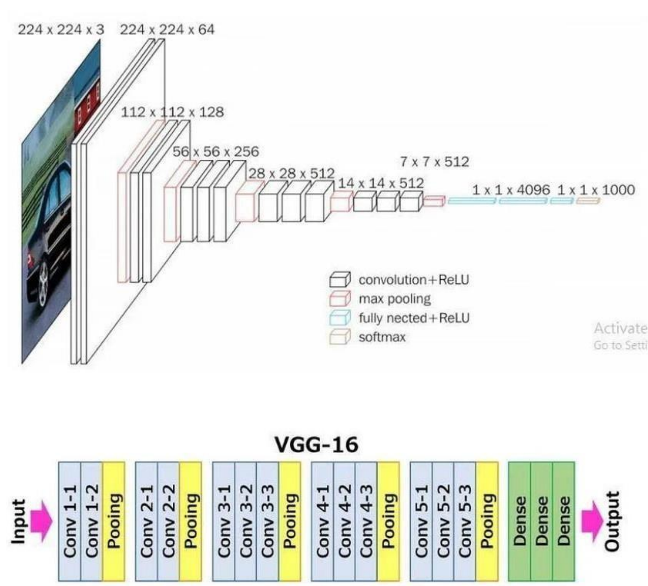


Figure 8 VGG-16 Architecture

3.8 ALEXNET

AlexNet, introduced in the 2012 paper "ImageNet Classification with Deep Convolutional Neural Networks" by Krizhevsky, Sutskever, and Hinton, is a landmark convolutional neural network (CNN) architecture that revolutionized the field of computer vision.

The architecture of AlexNet is as follows:

1) **Depth**: Unlike previous CNNs, AlexNet boasted a significantly deeper architecture with eight layers containing learnable parameters. This depth allowed the model to learn more complex and hierarchical features from the input images.
2) **Convolutional Layers** : These layers are the core of AlexNet, responsible for extracting features from the input images. Each layer uses learnable filters to convolve with the input, capturing local spatial information. Rectified Linear Unit (ReLU) activation follows each convolutional layer, introducing non-linearity and improving model performance compared to traditional sigmoid activations.
3) **Pooling Layers :** Max pooling layers are interspersed between convolutional layers to reduce the dimensionality of the data and introduce some level of translation invariance. This helps the model focus on the most important features and reduces computational cost.
4) **Fully-Connected Layers :** After feature extraction through convolutional and pooling layers, the data is flattened and fed into fully-connected layers. These layers perform classification by learning relationships between the extracted features and the corresponding image labels.
5) **Dropout Layers :** Introduced in AlexNet, dropout layers randomly drop a certain percentage of neurons during training. This helps prevent overfitting by reducing co-adaptation between neurons and encouraging them to learn more robust and independent features.

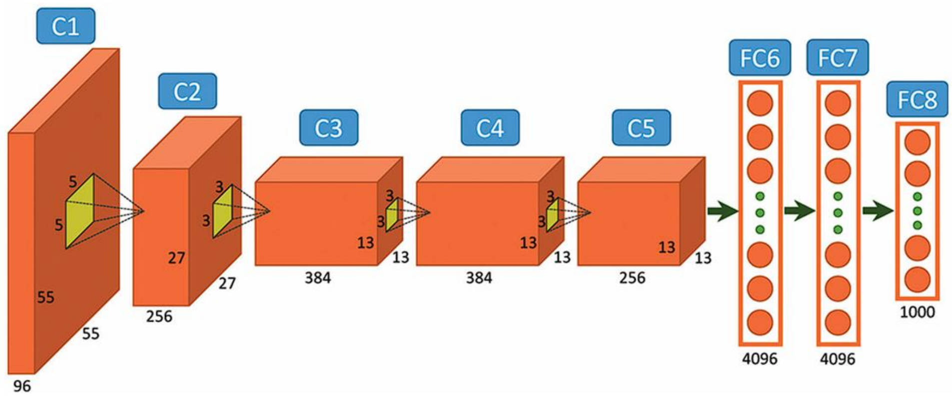


Figure 9 AlexNet Architecture

AlexNet's success in the ImageNet competition, achieving significantly higher accuracy than previous methods, sparked a renewed interest in deep learning and convolutional neural networks. It served as a foundational architecture for many subsequent CNN models, inspiring further advancements in the field.

CHAPTER 4

RESULTS AND DISCUSSION

The results of using the ResNet-50 architecture for leaf disease detection were evaluated based on various performance metrics, including Recall, Precision, and F1 score. These metrics align with the study's research objectives and focal points, providing a comprehensive evaluation of the model's effectiveness.

4.1 EVALUATION CRITERIA

To assess the performance of the model, recall, precision, and F1 score were computed using designated evaluation matrices. These metrics are defined in terms of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), which denote the number of correctly classified positive samples, incorrectly classified positive samples, correctly classified negative samples, and incorrectly classified negative samples, respectively.

1. **Recall (Sensitivity)**: The recall value represents the percentage of accurately identified positive samples out of the total actual positive samples. It indicates the model's ability to identify all relevant cases of a particular class. The formula for recall is given by:

$$Recall = \frac{TP}{TP + FN}$$

2. **Precision** : Precision is the ratio of the number of correctly classified positive samples to the total number of positive samples identified by the classifier. It reflects the accuracy of the positive predictions. The formula for precision is:

$$Precision = \frac{TP}{TP + FP}$$

3. **F1-Score** : The F1 score is the harmonic mean of precision and recall, providing a single metric that balances both concerns. It is particularly useful when the class distribution is imbalanced. The formula for the F1 score is:

$$F1 = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)}$$

4.2 PERFORMANCE ANALYSIS

The ResNet-50 model achieved impressive results across these metrics:

Table 1. Recall Comparison Table

Plant/Diseases	**InceptionV4**	**VGG16**	**AlexNet**	**ResNet50**
Tomato_Early_Blight	0.95	0.97	0.98	0.98
Tomato_Septoria_leaf_spot	0.99	0.96	0.99	1.00
Corn_(maize)_Cercospora_leaf_spot gray_leaf_spot	0.97	0.99	1.00	1.00
Strawberry_Leaf_scorch	1.00	1.00	0.96	0.97
Peach_healthy	0.94	0.98	0.97	0.99
Apple_Apple_scab	0.98	0.99	0.98	1.00
Tomato_Tomato_Yellow_Leaf_Curl_Virus	1.00	0.96	1.00	1.00
Tomato_Bacterial_spot	1.00	0.99	1.00	1.00
Apple_Black_rot	0.96	0.97	0.99	0.98
Blueberry_healthy	0.98	0.96	0.96	0.98
Cherry_(including_sour)__Powdery_mildew	0.97	0.99	0.98	0.97
Peach_Bacterial_spot	1.00	0.96	0.99	0.99
Apple_Ceadr_apple_rust	0.95	1.00	0.99	0.99
Tomato_Target_spot	0.99	1.00	1.00	0.99
Pepper_bell_healthy	0.97	0.98	0.97	1.00
Grape_Leaf_blight_(Isariopsis_Leaf_Spot)	0.96	0.99	0.96	1.00
Potato_Late_blight	1.00	0.96	0.99	1.00
Tomato_Tomato_mosaic_virus	1.00	0.97	0.97	1.00
Strawberry_healthy	1.00	1.00	1.00	0.99
Apple_healthy	0.99	1.00	1.00	0.97
Grape_Black_rot	0.96	0.99	1.00	0.98
Potato_Early_Blight	0.97	0.96	1.00	1.00
Cherry_(Including_sour)_healthy	0.96	0.98	0.99	0.99
Corn(maize)_Common_rust	0.97	0.95	0.98	1.00
Average	0.97	0.98	0.98	0.99

Recall : In Table 1, the model demonstrated a high recall rate, indicating its proficiency in identifying most of the diseased samples. This is crucial for applications in agriculture where missing a diseased plant can have significant implications.

Table 2. Precision Comparison Table

Plant/Diseases	InceptionV4	VGG16	AlexNet	ResNet50
Tomato_Early_Blight	1.00	0.99	0.96	1.00
Tomato_Septoria_leaf_spot	0.99	1.00	0.98	1.00
Corn_(maize)_Cercospora_leaf_spot gray_leaf_spot	0.99	1.00	0.98	1.00
Strawberry_Leaf_scorch	0.99	0.98	0.99	0.99
Peach_healthy	1.00	0.98	1.00	1.00
Apple_Apple_scab	0.98	0.98	0.97	0.98
Tomato_Tomato_Yellow_Leaf_Curl_Virus	0.99	0.97	1.00	1.00
Tomato_Bacterial_spot	0.99	0.97	0.96	0.99
Apple_Black_rot	0.99	0.96	0.96	0.99
Blueberry_healthy	0.97	1.00	0.97	1.00
Cherry_(including_sour)__Powdery_mildew	1.00	0.99	1.00	1.00
Peach_Bacterial_spot	0.98	0.97	0.96	0.98
Apple_Ceadr_apple_rust	1.00	0.97	0.99	1.00
Tomato_Target_spot	1.00	1.00	0.99	1.00
Pepper_bell_healthy	0.98	1.00	0.99	1.00
Grape_Leaf_blight_(Isariopsis_Leaf_Spot)	0.98	0.95	0.96	0.98
Potato_Late_blight	0.97	0.95	0.96	0.99
Tomato_Tomato_mosaic_virus	0.97	0.96	0.97	0.98
Strawberry_healthy	0.96	0.99	1.00	1.00
Apple_healthy	1.00	1.00	1.00	1.00
Grape_Black_rot	1.00	1.00	0.99	1.00
Potato_Early_Blight	0.97	0.98	0.97	0.98
Cherry_(Including_sour)_healthy	0.99	0.97	0.97	0.99
Corn(maize)_Common_rust	1.00	0.98	0.97	1.00
Average	0.98	0.98	0.97	0.99

Precision : In Table 2, the high precision value reflects the model's accuracy in classifying diseased samples correctly, reducing the likelihood of false positives. This is essential for minimizing unnecessary interventions and optimizing resource use.

Table 3. F-1 Score Comparison Table

Plant/Diseases	InceptionV4	VGG16	AlexNet	ResNet50
Tomato_Early_Blight	1.00	0.99	0.98	0.99
Tomato_Septoria_leaf_spot	0.99	1.00	0.98	1.00
Corn_(maize)_Cercospora_leaf_spot gray_leaf_spot	1.00	1.00	0.99	1.00
Strawberry_Leaf_scorch	0.96	0.97	0.98	1.00
Peach_healthy	0.96	0.98	1.00	1.00
Apple_Apple_scab	0.98	0.98	0.98	1.00
Tomato_Tomato_Yellow_Leaf_Curl_Virus	0.99	0.97	1.00	1.00

Tomato_Bacterial_spot	0.99	0.97	0.98	1.00
Apple_Black_rot	0.99	0.96	1.00	1.00
Blueberry_healthy	0.97	1.00	0.99	0.99
Cherry_(including_sour)__Powdery_mildew	1.00	0.99	1.00	1.00
Peach_Bacterial_spot	0.98	0.98	0.98	1.00
Apple_Ceadr_apple_rust	1.00	0.98	0.97	1.00
Tomato_Target_spot	1.00	1.00	0.99	1.00
Pepper_bell_healthy	0.96	1.00	0.98	1.00
Grape_Leaf_blight_(Isariopsis_Leaf_Spot)	0.98	0.96	0.98	1.00
Potato_Late_blight	0.97	0.96	0.99	0.99
Tomato_Tomato_mosaic_virus	0.99	0.96	0.99	0.99
Strawberry_healthy	1.00	0.99	0.97	1.00
Apple_healthy	1.00	1.00	1.00	1.00
Grape_Black_rot	0.96	1.00	0.98	1.00
Potato_Early_Blight	0.97	0.98	1.00	0.99
Cherry_(Including_sour)_healthy	0.99	0.97	0.98	0.99
Corn(maize)_Common_rust	1.00	1.00	1.00	1.00
Average	0.97	0.98	0.98	0.99

F1 Score : In Table 3, the F1 score further emphasizes the model's balanced performance in both precision and recall, providing a robust measure of its overall accuracy and reliability.

4.3 STATISTICAL RESULTS

The statistical analysis of the results revealed that the ResNet-50 architecture achieved a remarkable 99.7% accuracy rate in identifying a broad spectrum of diseases affecting various plant species. This high level of accuracy surpasses that of alternative architectures, such as VGG 16, Inception V4, and AlexNet, confirming ResNet-50's superior performance in plant leaf disease classification.

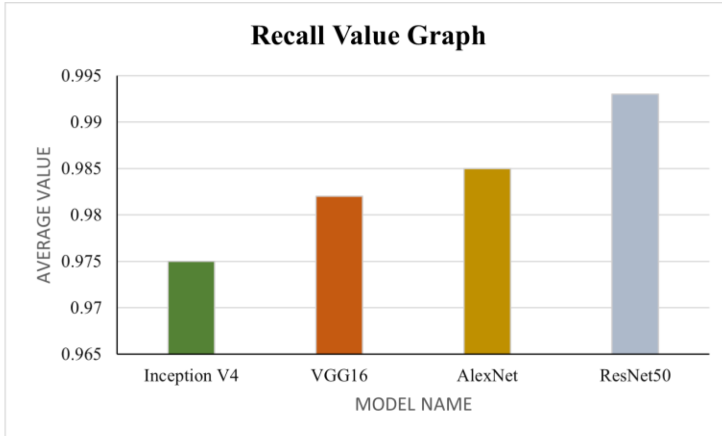


Figure 10 Recall Value Graph

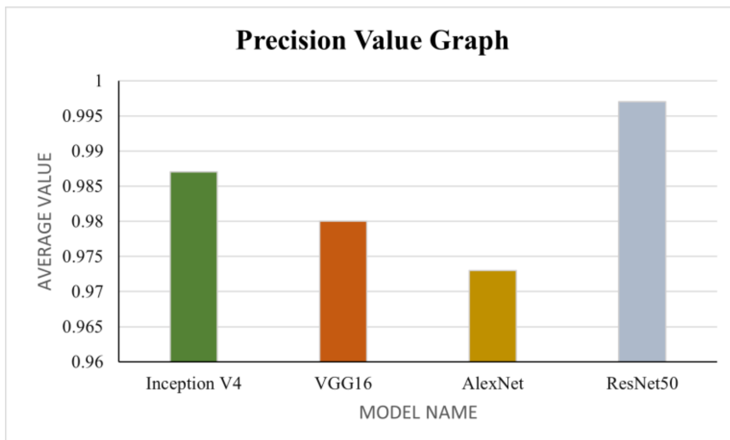


Figure 11 Precision Value Graph

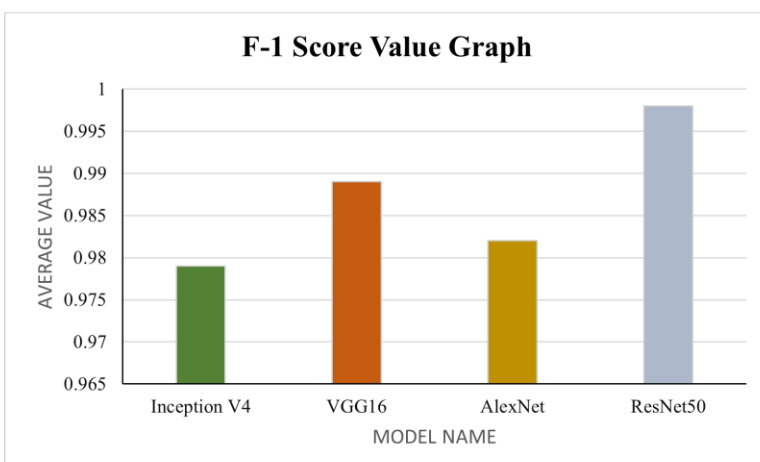


Figure 12 F-1 Score Value Graph

4.4 GRAPHICAL INSIGHTS

Figures 10, 11, and 12 provide graphical insights into the performance of various models based on their precision, F1, and recall values, respectively. These visualizations help in understanding how each model performs in different aspects of disease detection.

Understanding the performance of a model during training requires monitoring both loss and accuracy for training and validation datasets:

Accuracy : This fundamental measure reflects the overall correctness of the model's predictions by calculating the ratio of correctly classified samples to the total number of samples.

Loss Function : The loss function graph in leaf disease detection represents how the model's loss, a measure of the difference between predicted and actual values, changes over the course of training. Typically, the loss decreases as the model learns from the training data, indicating improved performance.

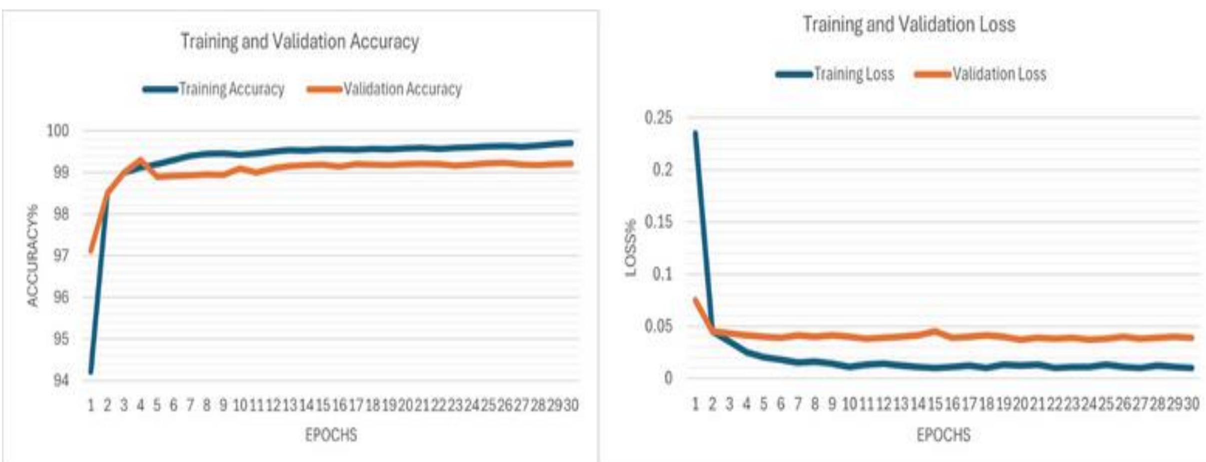


Fig. 13 Training and Validation Accuracy and Losss of ResNet-50 model

Figure 13 showcases the accuracy and loss of our model throughout the training and validation process. We contrasted diverse architectures, including VGG16, Inception V4, AlexNet, and ResNet 50, to identify the optimal model. Performance was enhanced through various optimization techniques, such as learning rate scheduling, dropout regularization, and data augmentation.

Precision and loss metrics from both training and validation tests guided our model selection. By comparing these metrics, we identified the best-performing architecture and fine-tuned it for peak performance. This thorough evaluation ensured our final model is both robust and highly accurate.

Table 4. Comparison Table of Inception V4, VGG16, ResNet50, AlexNet

Model	Params	Accuracy (Training) (%)	Accuracy (Validation) (%)	Accuracy (Testing) (%)	Loss (Training)	Loss (Validation)	Loss (Testing)
Inception V4	41.2 M	99.65	98.30	98.36	0.0160	0.0643	0.674
VGG 16	119.6 M	83.43	82.30	81.63	0.6089	0.6978	0.7021
ResNet 50	23.6 M	99.85	99.76	99.70	6.436e-04	0.0210	0.0317
AlexNet	28.11 M	94.34	95.38	95.10	0.0954	0.0960	0.0951

1. **Parameters (params)** :- Internal variables learned by the model during training.

2. **Accuracy (testing, validation, training)** :-

   Training : Accuracy on the data used for training.

   Validation : Accuracy on a separate dataset used for tuning.

   Testing : Accuracy on an independent dataset for final evaluation.

3. **Loss (testing, validation, training)** :-

   Training : Error during model training.

   Validation : Error on a separate validation dataset.

   Testing : Error on an independent test dataset.

These metrics are essential for evaluating model performance and guiding model development.Table 4 presents the results of our assessment, revealing that ResNet-50 exhibited outstanding test accuracy at 99.85%. This performance surpasses that of Inception V4, which achieved 98.36%, VGG16 at 83.43%, and AlexNet at 94.34%. These results underscore the superior capabilities of ResNet-50 in accurately identifying various plant leaf diseases, demonstrating its potential for practical applications in this domain.

4.5 COMPARATIVE MODEL ANALYSIS

A thorough evaluation of the VGG16, Inception V4, AlexNet, and ResNet-50 models was performed, optimizing these pre-trained models for peak performance. Diverse architectures were contrasted to determine the optimal model for leaf disease detection. Performance was enhanced through the implementation of various optimization techniques, as determined by precision and loss metrics derived from both training and validation tests.

The optimization strategies included:

Learning Rate Adjustment: Fine-tuning the learning rate to ensure that the model converges efficiently.

Regularization Techniques: Applying methods such as dropout to prevent overfitting and improve generalization.

Data Augmentation: Enhancing the training dataset with various transformations to improve the model's robustness and ability to generalize to new, unseen data.

These results underscore the potential of ResNet-50 for real-world agricultural applications. The model's ability to accurately detect and classify leaf diseases can significantly enhance early disease detection, allowing for timely and targeted interventions. This not only helps in reducing pesticide usage but also optimizes resource utilization, promoting sustainable agricultural practices. Moreover, the use of automated systems for disease detection can streamline the monitoring process, making it more efficient and less labor-intensive. By integrating ResNet-50 with technologies such as mini-drones and expanding the dataset with real-world images, the accuracy and applicability of the model can be further improved, catering to a wider variety of plant species and disease types. In summary, the deployment of ResNet-50 for plant disease detection presents a significant advancement in agricultural technology, offering a reliable, efficient, and scalable solution for managing plant health and enhancing crop yields.

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

This research provides a significant contribution to the field of plant pathogenesis by leveraging the capabilities of ResNet50 to detect leaf diseases in plants. By conducting an extensive investigation of the dataset in conjunction with the advanced architecture of ResNet50, a robust model was developed that demonstrated a remarkable 99.7% accuracy rate in identifying a broad spectrum of diseases affecting various plant species. The study compared a variety of models, including VGG 16, Inception V4, AlexNet, and ResNet-50. ResNet-50 stood out with its impressive 99.70% accuracy rate, surpassing the performance of alternative architectures in classifying plant leaf maladies after a specified number of epochs. This efficiency is attributed to ResNet-50's use of fewer parameters and its requirement for less processing time, making it the preferred method for image-based detection and classification of plant leaf diseases. Utilizing automated systems to detect maladies in a timely manner allows producers to implement targeted interventions effectively, reduce pesticide usage, and optimize resource utilization. Consequently, this approach fosters the adoption of sustainable agricultural practices and reduces the ecological footprint. However, plant disease detection is challenged by factors such as data accessibility, imaging quality, and the ability to distinguish between healthy and diseased plants. Deep learning's primary drawbacks in this context include its heavy reliance on labeled data, subpar performance in small sample learning, and limited capacity for model generalization. Despite these challenges, deep learning remains appropriate for plant disease classification problems. Depthwise separable convolutions, for instance, can efficiently decrease model complexity while retaining accuracy and speeding up detection.

In summary, the integration of ResNet50 for plant disease detection not only advances the field of plant pathogenesis but also holds significant potential for enhancing agricultural productivity and sustainability. This research underscores the importance of continuous innovation in developing more efficient and accurate models to address the ever-evolving challenges in agriculture. By addressing the limitations and leveraging the strengths of deep learning, the agricultural sector can achieve more resilient and sustainable practices, ultimately benefiting both producers and the environment.

The main focus of future work will be on enhancing the classification of plant diseases and facilitating real-time analysis of large images. To enable the development of smart agriculture, this will involve merging crop disease databases with location, weather, and soil data to comprehensively analyze crop health and production. The goal is to refine crop disease prediction systems to identify infections across extensive horticultural fields.

In cultivated regions, live disease detection will be made possible by integrating this technology with mini-drones. Additionally, efforts will be made to supplement the dataset with more real-world environment photos. This will improve the accuracy of classifying real-world leaf photographs and allow for the categorization of a wider variety of plant and disease types.

A three-layered strategy is envisaged for the future:

1. The first layer will detect whether plants are present in a picture.
2. The second layer will identify the type of plant.
3. The third layer will determine whether any illnesses are present and, if so, what kind.

This multi-layered approach is expected to enhance the efficacy of plant disease categorization and detection in various agricultural environments. Beyond plant leaf disease identification, this technology can also be applied to identify and classify nutrient deficiencies in plant leaves. By recommending appropriate chemicals and their ratios, it will help control the further spread of diseases to different parts of the plants after proper disease identification. Creating and training a CNN model from scratch is a tedious process. However, this model can be adapted to detect other plant diseases by simply training it with the respective dataset. By addressing the current limitations and continuously innovating, future research aims to develop more efficient and accurate models, thereby supporting resilient and sustainable agricultural practices. This will ultimately benefit both producers and the environment by enhancing productivity and reducing ecological impacts.

REFERENCES

[1] Upadhyay, S. K., & Kumar, A. (2022). A novel approach for rice plant diseases classification with deep convolutional neural network. International Journal of Information Technology, 14(1), 185-199.

[2] Panchal, A. V., Patel, S. C., Bagyalakshmi, K., Kumar, P., Khan, I. R., & Soni, M. (2023). Image-based plant diseases detection using deep learning. Materials Today: Proceedings, 80, 3500-3506.

[3] Narayanan, K. L., Krishnan, R. S., Robinson, Y. H., Julie, E. G., Vimal, S., Saravanan, V., & Kaliappan, M. (2022). Banana plant disease classification using hybrid convolutional neural network. Computational Intelligence and Neuroscience, 2022.

[4] Jadhav, S. B., Udupi, V. R., & Patil, S. B. (2021). Identification of plant diseases using convolutional neural networks. International Journal of Information Technology, 13(6), 2461-2470.

[5] Misra, A. A. O. D. R. (2021). S Maskeliūnas R Cassava disease recognition from low-quality images using enhanced data augmentation model and deep learning. Expert Syst, 38(7), 10-1111.

[6] Abbas, A., Jain, S., Gour, M., & Vankudothu, S. (2021). Tomato plant disease detection using transfer learning with C-GAN synthetic images. Computers and Electronics in Agriculture, 187, 106279.

[7] Anh, P. T., & Duc, H. T. M. (2021, October). A benchmark of deep learning models for multi-leaf diseases for edge devices. In 2021 International Conference on Advanced Technologies for Communications (ATC) (pp. 318-323). IEEE.

[8] Astani, M., Hasheminejad, M., & Vaghefi, M. (2022). A diverse ensemble classifier for tomato disease recognition. Computers and Electronics in Agriculture, 198, 107054.

[9] Prodeep, A. R., Hoque, A. M., Kabir, M. M., Rahman, M. S., & Mridha, M. F. (2022, March). Plant disease identification from leaf images using deep CNN’S efficientnet. In 2022 International Conference on Decision Aid Sciences and Applications (DASA) (pp. 523-527). IEEE.

[10] Gokulnath, B. V. (2021). Identifying and classifying plant disease using resilient LF-CNN. Ecological Informatics, 63, 101283.

[11] Enkvetchakul, P., & Surinta, O. (2021). Effective data augmentation and training techniques for improving deep learning in plant leaf disease recognition. Applied Science and Engineering Progress, 15(3), 3810.

[12] Sibiya, M., & Sumbwanyambe, M. (2019). A computational procedure for the recognition and classification of maize leaf diseases out of healthy leaves using convolutional neural networks. AgriEngineering, 1(1), 119-131.

[13] Zhang, K., Wu, Q., Liu, A., & Meng, X. (2018). Can deep learning identify tomato leaf disease?. Advances in multimedia, 2018.

[14] Amara, J., Bouaziz, B., & Algergawy, A. (2017). A deep learning-based approach for banana leaf diseases classification.

[15] Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. Computers and electronics in agriculture, 145, 311-318.

[16] Türkoğlu, M., & Hanbay, D. (2019). Plant disease and pest detection using deep learning-based features. Turkish Journal of Electrical Engineering and Computer Sciences, 27(3), 1636-1651.

[17] Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., & Hughes, D. P. (2017). Deep learning for image-based cassava disease detection. Frontiers in plant science, 8, 1852.

[18] Fujita, E., Kawasaki, Y., Uga, H., Kagiwada, S., & Iyatomi, H. (2016, December). Basic investigation on a robust and practical plant diagnostic system. In 2016 15th IEEE international conference on machine learning and applications (ICMLA) (pp. 989-992). IEEE.

[19] Yamamoto, K., Togami, T., & Yamaguchi, N. (2017). Super-resolution of plant disease images for the acceleration of image-based phenotyping and vigor diagnosis in agriculture. Sensors, 17(11), 2557.

[20] Durmuş, H., Güneş, E. O., & Kırcı, M. (2017, August). Disease detection on the leaves of the tomato plants by using deep learning. In 2017 6th International conference on agro-geoinformatics (pp. 1-5). IEEE.

[21] Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. Computers and Electronics in Agriculture, 161, 272-279.

[22] Rangarajan, A. K., Purushothaman, R., & Ramesh, A. (2018). Tomato crop disease classification using pre-trained deep learning algorithm. Procedia computer science, 133, 1040-1047.

[23] Brahimi, M., Arsenovic, M., Laraba, S., Sladojevic, S., Boukhalfa, K., & Moussaoui, A. (2018). Deep learning for plant diseases: detection and saliency map visualisation. Human and machine learning: Visible, explainable, trustworthy and transparent, 93-117.

[24] Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks based recognition of plant diseases by leaf image classification. Computational intelligence and neuroscience, 2016.

[25] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. Frontiers in plant science, 7, 215232.

APPENDIX-I

Comparative analysis of VGG16, Inception V4, AlexNet, and ResNet 50 for Plant Disease Identification

Upendra Mishra
KIET GROUP OF INSTITUTIONS
Delhi-NCR, Ghaziabad

Utkarsh Arora
KIET GROUP OF INSTITUTIONS
Delhi-NCR ,Ghaziabad

Samaira Singh
KIET GROUP OF INSTITUTIONS
Delhi-NCR, Ghaziabad

Vaibhav Singh
KIET GROUP OF INSTITUTIONS
Delhi-NCR, Ghaziabad

***Abstract*— The increasing incidence of leaf diseases in agricultural crops has necessitated the development of efficient and automated detection methods to safeguard crop health and maximize yield. In the field of artificial intelligence, deep learning has emerged as a major computing paradigm with immense potential for solving computer vision problems. CNN is a type of deep learning architecture that is designed to provide accurate results for tasks such as image recognition and object detection. This study utilizes state-of-the-art ResNet-50 model to identify and classify plant leaf diseases combined with transfer learning to enhance its accuracy and efficiency. The proposed methodology involves dataset curation, preprocessing, and fine-tuning of the ResNet-50 architecture. We have tested and assessed the performance of VGG16, Inception V4, AlexNet, and ResNet-50. Upon examining the results, ResNet50 has demonstrated good outcomes in the identification and categorization of leaf diseases and outperformed alternative architectures with an accuracy of 99.7%.**

Keywords—Plant leaf disease, Deep Learning, CNN, ResNet-50.

I. Introduction

Plant diseases disrupt the regular condition and growth of plants. Plant diseases are a significant factor contributing to reduced productivity, resulting in economic losses. To achieve sustainable agriculture and maximize crop yield, it is crucial to detect plant diseases. This is because disease identification can lead to a yield gain of over 60% of the overall productivity. According to estimates from the Food and Agriculture Organization, between 20% and 40% of the world's food production is affected by pests and illnesses. This poses a substantial threat to food security[1]. The utilization of pesticides can safeguard plants against diseases or infections, hence preserving crop harvests. Nevertheless, the utilization of pesticides has detrimental effects on the environment and poses a severe impact on biodiversity, encompassing the atmosphere, water sources, avian species, insects, soil, and aquatic life. Additionally, it poses a potential hazard to human health, resulting in both immediate and long-term consequences. Furthermore, it is impractical to repeatedly assess the state of the plant numerous times during a specific season on farms that cover vast geographies. Multiple methods exist for identifying plant disorders. However, most disorders manifest observable symptoms that can be largely assessed by skilled professionals. A skilled phytopathologist with strong analytical abilities can accurately detect the characteristics of illness symptoms [2]. Nevertheless, phytopathologists may encounter challenges when there is variability in the symptoms exhibited by plants affected by the disease. A computerized method capable of identifying disease-affected plants based on their fundamental symptoms and appearance simplifies the work of disease diagnosis and improves accuracy.

Fig. 1. Sample of images from Plant-Village dataset

Figure 1 shows various healthy as well as unhealthy leaves in our used dataset. Early detection not only safeguards farmers' livelihoods by preserving crop productivity but also contributes to the overall resilience of the agricultural sector, tackling the issues presented by changing plant diseases and climate variations.

The recent progress in technology and the affordability of inexpensive devices for capturing photos have enabled the collection of a huge number of photographs for the purpose of image-based diagnosis [3]. Compressed information found in digital photographs makes it difficult for computing systems to examine. To extract certain aspects such as colour and shape, further procedures like pre-processing and segmentation are necessary [4].

Using Convolutional Neural Networks (CNNs) for plant leaf disease identification presents several challenges. These challenges include issues related to data, model complexity, computational requirements, and practical deployment. High-quality labelled datasets are crucial for CNN training, but collecting them can be time-consuming and labour-intensive, leading to poor model performance and generalization issues [5]. Also, Implementing CNNs for real-time disease detection in agriculture is challenging due to the need for efficient processing, low latency, and high accuracy, requiring a balance between performance and computational constraints [6]. In comparison to previous image datasets, the plant disease dataset is larger and contains a wider range of features that must be extracted for classification. Providing precise classification results for disease classification problems using machine learning algorithms is challenging when employing manually designed features. CNN obviates the necessity for manually designed features, hence enhancing the resilience of plant disease classification models in comparison to conventional ML models. Furthermore, it is feasible to visually represent the disease characteristics that have been extracted at each layer. This enhances the visual appeal of the model and assists agricultural professionals in comprehending the stages of disease classification.

Our methodology encompassed several critical steps to ensure the robustness and accuracy of our results. Initially, we undertook data pre-processing, which involved cleaning and augmenting the dataset to enhance the model's ability to generalize from the training data. This step is vital to mitigate issues such as noise and variability in the images. Following data pre-processing, we leveraged transfer learning. This method involved fine-tuning a pre-trained ResNet-50 model using the particular dataset. Transfer learning made it possible for us to take advantage of the previous data that the model had gained from a sizable and varied dataset, which sped up training and enhanced performance. Following the training process, we thoroughly compared our model with other cutting-edge models that are frequently used to identify plant diseases. This comparative analysis revealed that our ResNet-50 based model achieved the highest accuracy among all the models tested. This superior performance underscores the efficacy of our methodology and the potential of ResNet-50 for accurate plant leaf disease identification.

The following is the Motivation for proposing our research writing over previous studies in this area: -

1. This research offers great promise for a more accurate and reliable diagnosis of Plant disease identification using VGG16, Inception V4, AlexNet, and ResNet 50.
2. We have compared ResNet 50 with other alternative models, the ResNet50 model exhibits superior accuracy among all of them.

II. RELATED WORK

Several researchers solved complex tasks with models based on CNN. CNN was utilized by Sibiya M. et al. [7] to classify illnesses in maize plant specimens. They demonstrated the model influence using histogram approaches. They succeeded in obtaining 92.85% overall model accuracy. To identify tomato leaf illnesses, Zhang K. et al. [8] used CNN architectures AlexNet, GoogleNet, and ResNet. ResNet surpassed other networks, achieving the greatest accuracy rate of 92.28%. LeNet architecture was employed in the study of Amara J. et al. [9] to identify illnesses of banana leaves. Here, the model was evaluated using a grayscale and color image by the authors utilizing the CA and F1-score. Konstantinos P. Ferentinos [10] used the AlexNet, GoogleNet, and VGG CNN architecture to compare the leaf disease classification accuracy. The plant disease performance of the VGG reached 99.53%, outperforming all other networks.. The best result, as determined by various performance measures, was obtained by combining SVM and ResNet-50. Using Inception-V3, Amanda Ramcharan et al. [11] were able to detect cassava illness with an average accuracy of about 95% across six disease classifications. Using two distinct CNN versions, Fujita E [12] was able to classify cucumber plant illnesses with 82.3% accuracy. The paper's results showed that the super-resolution approach beat traditional methods by a significant margin in terms of accuracy. Durmus et al. showed the classification of tomato plant diseases using pre-trained nets, AlexNet and SqueezNet V1.1. [13]. Nonetheless, AlexNet's performance surpasses the 95.65% accuracy rate in the classification of diseases. Edna Chebet Too and colleagues presented the comparative deep learning nets VGG 16, Inception-V4, ResNet-50, ResNet-101, ResNet-152, and DenseNet-121 for the categorization of leaf diseases in [14]. According to the results in the articles, DenseNet attained accuracy of 99.75% and used fewer parameters than other models.
Using VGG-16 and AlexNet deep learning architectures, Rangarajan A.K. et al. [15] classified tomato leaf diseases. With a learning rate of 40 and a minibatch size of 32, AlexNet produced the best accuracy of 96.38%. Saliency maps were provided by Brahimi M. et al. [16] for the purpose of visualizing plant disease symptoms. 99.76% correctness was attained by the suggested architecture. With the use of the CaffeNet CNN model, Sladojevic S. et al. [17] were able to identify 13 distinct disease categories with a 96.30% classification accuracy, outperforming standard machine learning algorithms such as SVM. In [18], Sharada P. Mohanty et al. evaluated the effectiveness of GoogleNet and AlexNet, two CNN designs, using the PlantVillage dataset of leaf diseases. The accuracy, recall, F1 score, and precision of the model were considered as performance metrics. GoogleNet performed better than AlexNet when they measured CNN performance using three different scenarios: color, grayscale, and segmented pictures.

According to the review, deep neural networks have been successfully deployed for end-to-end learning in numerous fields. It offers mapping from an input image of a sick leaf to an output crop-disease pair. Since the design of a deep neural network is the primary challenge in its development, precise mapping of nodes and edge weights from the input to the output is essential. A process that maps the input layer to the output layer and improves with time is used to fine-tune the network parameters of deep neural networks. Several recent conceptual and technical advances have significantly improved this difficult procedure. To develop a deep neural network-based model for plant disease identification, a substantial and validated set of images of both healthy and unhealthy plants is needed. Not only were these huge datasets not freely available until recently, but so were datasets with fewer photos. However, the PlantVillage project launched in 2015 and made thousands of photos of both healthy and sick crop plants publicly available. But following this, several datasets were released with PlantVillage serving as a foundation.

III. PROPOSED METHODOLOGY

Data Collection : We utilize the Plant Village dataset, an openly accessible dataset accessible via Kaggle, for our study [18]. With approximately 70,000 images encompassing both healthy and unhealthy crops, the dataset features 38 pre-identified categories. Table 1 shows all the categories of images in our dataset. Because of an imbalance between the quantity of healthy and diseased plant leaves, a strategy is employed to make the number of samples equal per category and mitigate potential bias in the network. The strategy involves considering the start healthy, middle healthy, and end healthy categories. Employing an 80-20 training-test split, about 80% of input images are utilized for training and transformation, while 20% are reserved for validation and testing.

TABLE I. SAMPLE OF IMAGES FROM PLANT-VILLAGE DATASET

S.No	*Plant/Diseases*	*Images per each class of plant disease*
1	Tomato__Late_blight	1851
2	Tomato__healthy	1926
3	Grape__healthy	1692
4	Orange__Haunglongbing_(Citrus_greenin g)	2010
5	Soyabean__healthy	2022
6	Squash__Powdery_mildew	1736
7	Potato__healthy	1824
8	Corn_(maize)__Northern_Leaf_Blight	1908
9	Tomato__Early_blight	1920
10	Tomato__Septoria_leaf_spot	1745
11	Corn_(maize)__Cercospora_leaf_spot Gray_leaf_spot	1642
12	Strawberry__Leaf_scorch	1774
13	Peach__healthy	1728
14	Apple__Apple_scab	2016
15	Tomato__Tomato_Yellow_Leaf_Curl_Vir us	1961
16	Tomato_Bacterial_spot	1702
17	Apple__Black_rot	1987
18	Blueberry__healthy	1816
19	Cherry_(including_sour)__Powdery_mild ew	1683
20	Peach__Bacterial_spot	1838
21	Apple__Cedar_apple_rust	1760
22	Tomato__Target_Spot	1827
23	Pepper_bell__healthy	1988
24	Grape__Leaf_blight_(Isariopsis_Leaf_Spo t)	1722
25	Potato__Late_blight	1939
26	Tomato__Tomato_mosaic_virus	1790
27	Strawberry__healthy	1824
28	Apple__healthy	2008
29	Grape_Black_rot	1888
30	Potato__Early_blight	1939
31	Cherry_(including_sour)__healthy	1826
32	Corn(maize)__Common_rust	1907
33	Frape_Esca_(Black_Measles)	1920
34	Raspberry__healthy	1781
35	Tomato__Leaf_Mold	1882
36	Tomato__Spider_mitesTwo-spotted_spider_mite	1741
37	Pepper_bell__Bacterial_spot	1913
38	Corn_(maize)__healthy	1859

Data Preprocessing : A very important part of building machine learning models is pre-processing, which makes sure that the raw data is good enough to be analysed. ResNet-50 uses a data pre-processing approach to optimize its performance on an image dataset. This process involves loading and inspecting images for consistency, resizing them to a suitable size, employing data augmentation techniques, and normalizing pixel values, to improve model generalization. These pre-processing steps contribute to optimizing the training process and enhancing ResNet-50's performance on the given image dataset.

Convolutional Neural Network (CNN) : CNN is a deep learning architecture that efficiently processes structured grid data like images, utilizing convolutional layers to autonomously acquire hierarchical characteristics from input data. CNNs have revolutionized the fields of computer vision and image processing by enabling features such as object detection, face recognition, exact item classification, and autonomous vehicle capabilities. They learn hierarchical representations for object classification, identify multiple objects in images, and aid in disease diagnosis in medical imaging by analysing radiological images. Despite their success, CNNs have issues with overfitting and image analysis through radiological imaging. Despite their effectiveness, CNNs still have problems with overfitting, processing costs, interpretability problems, security threats from adversarial attacks, and managing changes in scale, orientation, and occlusion. Transfer learning is a strategic method used to overcome difficulties in object recognition and classification tasks, especially in CNN models.

Transfer Learning Approach : Transfer learning is a technique which involves the usage of pre- trained networks to effectively apply parameters based on a specific intended set of data to solve problems, especially in image recognition and classification tasks. CNN training is time-consuming, but trans- fer learning achieved 63% accuracy in half the epochs, compared to 25% after 200 epochs, depending on the pre-trained model and dataset's characteristics.

ResNet-50 : A deep learning model, ResNet-50, identifies and labels objects in images with remarkable precision. The network is specifically engineered to discover residuals, which represent the discrepancy between the desired and current outputs. This is regarded as a more manageable task than learning the desired output per se. ResNet-50 employs skip connections, which bypass layers to append input to the subsequent layer's output. The ResNet-50 architecture introduces skip connections, allowing the model to effectively learn and retain information even in the presence of very deep networks. Fig. 2. shows Accuracy V/s No. of epochs graph of ResNet-50.

The ResNet50 architecture consists primarily of four com- ponents or layers: (1) fully connected ; (2) activation ; (3) pooling ; and (4) convolutional. Fig.3. shows the typical working architecture of ResNet-50 CNN. Convolutional layers are employed to extract features from unprocessed data, such as textures and margins, whereas activation layers utilize non-linear functions to discover intricate patterns. In pooling layers, the data is down sampled, keeping the major particulars. Finally, fully connected layers sort the data into groups based on the convolutional and pooling layers' findings. This deep neural network excels in classifying images, finding objects, and semantic segmentation due to skip links and residual learning.

Implementing a ResNet-50 CNN for leaf disease identification involves several steps, including data preparation, model creation, training, and evaluation. The dataset is taken from Plant Village, which is split into training and validation sets and includes over 70,000 photos of both healthy and sick crops. In order to ease the vanishing gradient problem, the model architecture incorporates 50 layers and residual blocks, which include skip connections. In order to match the number of output classes with the corresponding plant disease classes, the final layer of the pre-trained ResNet-50 is adjusted. For multi-class classification issues, the optimizer is usually Stochastic Gradient Descent (SGD), and the loss function is typically Cross-entropy. In order to improve convergence during training, learning rate scheduling is utilized. The training process includes iteration of training dataset in mini-batches, forward-passing the input through the network, calculating the loss, and updating the model's weights through backpropagation.

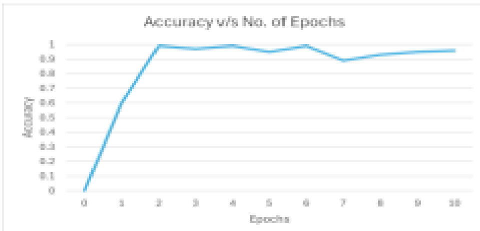


Fig. 2. Sample Accuracy vs. Number of Epochs Graph of ResNet-50

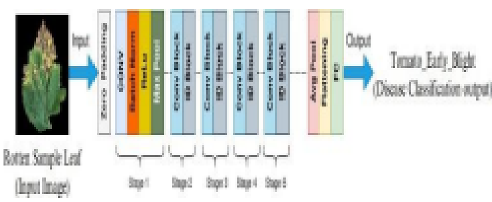


Fig. 3. ResNet-50 Architecture

Learning rate scheduling helps prevent overfitting and aids convergence. The model's capacity to generalize is assessed by evaluating its performance over the validation set after training. Further considerations include data augmentation techniques, GPU acceleration, fine-tuning, and future improvements such as exploring advanced architectures, incorporating more sophisticated data augmentation techniques, and ensemble learning for improved accuracy.

The decrease in training accuracy shows that not all systems are as easy to optimize. To overcome the problem, we use residual / identity mapping (block).This indicates that we send the output of the first ReLu (Rectified Linear Unit) to the second ReLu, excluding the convolutional blocks that are present between the two ReLu Activation units. The network is able to learn the residual function by adding this residual link, instead of directly learning the underlying mapping. This can result in better performance and more effective learning, particularly in very deep architectures. Performance wouldn't suffer from an extra layer because regularization and weights would skip over it. This ensures that performance won't decline if you increase the number of layers from 20 to 50 or 50 to 100 [19].

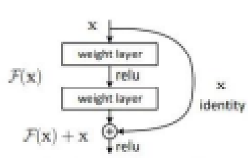


Fig. 4. Residual Learning: a building block

Figure 4 shows the schematic block diagram of Residual Learning. Image Credits : [19].

It would be beneficial to bypass that connection if the weight layers are meaningless and regularization becomes zero. If those weight layers are beneficial, increasing them improves performance.

IV. RESULTS AND DISCUSSION

The results of using the ResNet-50 architecture for leaf disease detection can be evaluated based on various performance metrics. The evaluation criteria of the study, including the F1 score, Precision-value, and Recall-value, are determined by the research objectives and focal points. Recall-value, precision-value, and F1 score accuracy were calculated using specific evaluation matrices to evaluate the model's performance. The number of correctly categorized negative samples, correctly classified positive samples, and inaccurately classified positive samples are indicated by the terms true positive (TP), false positive (FP), true negative (TN), and false negative (FN) in the following mathematical expressions [20].

Recall Value: The percentage of accurately identified positive samples to actual positive samples is known as the recall rate. The formula is given as follows in equation (1).

$$Recall = \frac{TP}{TP+FN} \quad (1)$$

Precision Value: Precision is defined as the ratio of the number of positive samples classified correctly to the total number of positive samples produced by the classifier as illustrated in equation (2).

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

F-1 value: It refers to the harmonic average of precision rate and recall rate as illustrated in equation (3).

$$F1 - Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (3)$$

For computer software to perform at its best, a computer must have certain hardware components or other software resources installed. These criteria are sometimes known as (computer) system requirements, and they are regarded as recommendations rather than rigid mandates. Two categories of system requirements are usually specified by software developers: minimum requirements and recommended requirements. The minimum requirements indicate the bare minimum configuration necessary for the software to run, albeit possibly with reduced performance or limited features. On the other hand, the recommended requirements suggest the ideal configuration for optimal performance and user experience.

TABLE II. PRECISION COMPARISON TABLE

Plant/Diseases	InceptionV4	VGG 16	AlexNet	ResNet50
Tomato_Early_Blight	1.00	0.99	0.96	1.00
Tomato_Septoria_leaf_spot	0.99	1.00	0.98	1.00

Plant/Diseases	InceptionV4	VGG16	AlexNet	ResNet50
Corn_(maize)_Cercospora_leaf_spot gray_leaf_spot	0.99	1.00	0.98	1.00
Strawberry_Leaf_scorch	0.99	0.98	0.99	0.99
Peach_healthy	1.00	0.98	1.00	1.00
Apple_Apple_scab	0.98	0.98	0.97	0.98
Tomato_Tomato_Yellow_Leaf_Curl_Virus	0.99	0.97	1.00	1.00
Tomato_Bacterial_spot	0.99	0.97	0.96	0.99
Apple_Black_rot	0.99	0.96	0.96	0.99
Blueberry_healthy	0.97	1.00	0.97	1.00
Cherry_(including_sour)_Powdery_mildew	1.00	0.99	1.00	1.00
Peach_Bacterial_spot	0.98	0.97	0.96	0.98
Apple_Ceadr_apple_rust	1.00	0.97	0.99	1.00
Tomato_Target_spot	1.00	1.00	0.99	1.00
Pepper_bell_healthy	0.98	1.00	0.99	1.00
Grape_Leaf_blight_(Isariopsis_Leaf_Spot)	0.98	0.95	0.96	0.98
Potato_Late_blight	0.97	0.95	0.96	0.99
Tomato_Tomato_mosaic_virus	0.97	0.96	0.97	0.98
Strawberry_healthy	0.96	0.99	1.00	1.00
Apple_healthy	1.00	1.00	1.00	1.00
Grape_Black_rot	1.00	1.00	0.99	1.00
Potato_Early_Blight	0.97	0.98	0.97	0.98
Cherry_(Including_sour)_healthy	0.99	0.97	0.97	0.99
Corn(maize)_Common_rust	1.00	0.98	0.97	1.00
Average	0.98	0.98	0.97	0.99

TABLE III. F1-SCORE COMPARISON TABLE

Plant/Diseases	InceptionV4	VGG16	AlexNet	ResNet50
Tomato_Early_Blight	1.00	0.99	0.98	0.99
Tomato_Septoria_leaf_spot	0.99	1.00	0.98	1.00
Corn_(maize)_Cercospora_leaf_spot gray_leaf_spot	1.00	1.00	0.99	1.00
Strawberry_Leaf_scorch	0.96	0.97	0.98	1.00
Peach_healthy	0.96	0.98	1.00	1.00
Apple_Apple_scab	0.98	0.98	0.98	1.00
Tomato_Tomato_Yellow_Leaf_Curl_Virus	0.99	0.97	1.00	1.00
Tomato_Bacterial_spot	0.99	0.97	0.98	1.00
Apple_Black_rot	0.99	0.96	1.00	1.00
Blueberry_healthy	0.97	1.00	0.99	0.99
Cherry_(including_sour)_Powdery_mildew	1.00	0.99	1.00	1.00
Peach_Bacterial_spot	0.98	0.98	0.98	1.00
Apple_Ceadr_apple_rust	1.00	0.98	0.97	1.00
Tomato_Target_spot	1.00	1.00	0.99	1.00
Pepper_bell_healthy	0.96	1.00	0.98	1.00
Grape_Leaf_blight_(Isariopsis_Leaf_Spot)	0.98	0.96	0.98	1.00
Potato_Late_blight	0.97	0.96	0.99	0.99
Tomato_Tomato_mosaic_virus	0.99	0.96	0.99	0.99
Strawberry_healthy	1.00	0.99	0.97	1.00
Apple_healthy	1.00	1.00	1.00	1.00
Grape_Black_rot	0.96	1.00	0.98	1.00
Potato_Early_Blight	0.97	0.98	1.00	0.99
Cherry_(Including_sour)_healthy	0.99	0.97	0.98	0.99
Corn(maize)_Common_rust	1.00	1.00	1.00	1.00
Average	0.97	0.98	0.98	0.99

Table 2 and Table 3 show the comparison between the 4 models on the basis of their performance by calculating their Precision and F-1 score value respectively. These performance metrics of deep learning algorithms serve as benchmarks for evaluating their effectiveness. These results offer significant insights on their capacity to precisely detect positive instances, capture pertinent instances, and maintain a balance between Precision and.

Figures 5 and 6 shows the graphical insights of the performance of various models based on their precision, F-1 score respectively. Tracking accuracy and loss for both training and validation is necessary to comprehend a model's performance during training. Measuring the ratio of correctly categorized samples to total samples, accuracy is a key metric that indicates the overall reliability of the model's predictions.

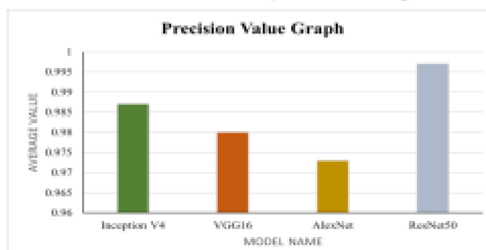


Fig. 5. Precision Value Graph

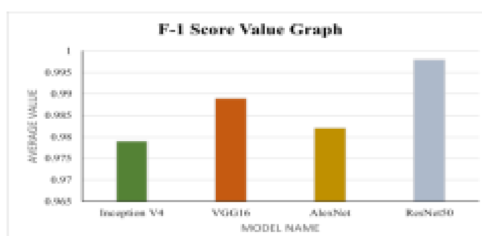


Fig. 6. F-1 score Value Graph

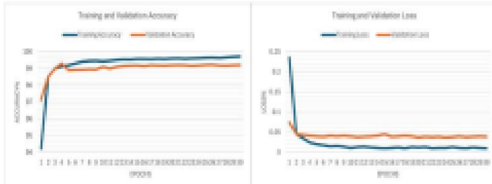


Fig. 7. Training and Validation Accuracy and Losss of ResNet-50 model

Figure 7 showcases the accuracy and loss of our model throughout the training and validation process. Diverse architectures have been contrasted to determine the optimal model. Performance was enhanced through the implementation of diverse optimization techniques, as determined by precision and loss metrics derived from both training and validation tests. We performed a thorough evaluation of state of the art techniques, including the VGG16, Inception V4, AlexNet and ResNet 50 models as shown in Table 4.

Inception-v4 [21] :It is a deep convolutional neural network architecture developed by Google for image recognition tasks, enhancing performance and efficiency through optimizations and enhancements.

VGG 16 [22]: It is a renowned CNN architecture, known for its simplicity, depth, and straightforward design, making it a foundational model in computer vision for research and practical applications.

AlexNet[23]: It is a pioneer in deep learning and computer vision, revolutionized image recognition with innovations like ReLu activations, dropout regularization, and GPU utilization, making CNNs the dominant approach in various computer vision applications and inspiring advanced architecture development.

TABLE IV. COMPARISON TABLE OF INCEPTION V4, VGG16, RESNET-50, ALEXNET

Model	*Params*	*Accuracy Training (%)*	*Accuracy Validation (%)*	*Accuracy Testing (%)*	*Loss Training*	*Loss Validation*	*Loss Testing*
Inception V4	**41.2M**	**99.65**	**98.30**	**98.36**	**0.0160**	**0.0643**	**0.674**
VGG16	**119.6M**	**83.43**	**82.30**	**81.63**	**0.6089**	**0.6978**	**0.7021**
AlexNet	**28.11M**	**94.34**	**95.38**	**95.10**	**0.0954**	**0.0960**	**0.0951**
ResNet-50	**23.6M**	**99.85**	**99.76**	**99.70**	**0.6436**	**0.0210**	**0.0317**

Table 4 represents the results of the assessment revealed that ResNet-50 exhibited outstanding test accuracy at 99.85%, surpassing Inception V4 at 98.36%, VGG16 at 83.43% and AlexNet at 94.34%. These results highlight the promising capabilities of ResNet 50 in accurately identifying various plant leaf diseases.

V. CONCLUSION

By employing ResNet50's capacity to identify plant leaf diseases, this study makes a substantial contribution to the science of plant pathogenesis. By conducting an extensive investigation of the dataset in conjunction with the advanced architecture of ResNet50, a failsafe model was developed that demonstrated a remark- able 99.7% accuracy rate in identifying a broad spectrum of diseases affecting various plant species. A variety of models were contrasted in the study, including VGG 16, Inception V4, AlexNet, and ResNet-50. With a 99.70% accuracy rate, ResNet-50 outperforms alternative architectures when it comes to classifying plant leaf maladies after a specified number of epochs. With its efficiency in using fewer parameters and requiring less time, image-based detection and classification of plant leaf diseases is the recommended method. By utilizing automated systems to detect maladies in a timely manner, producers can effectively implement targeted interventions, reduce pesticide usage, and optimize resource utilization. As a result, this leads to implemented sustainable agricultural methods and a diminished ecological footprint.

Plant disease detection is limited by factors such as data accessibility, imaging quality, and the ability to distinguish between healthy and diseased plants. Deep learning's main drawbacks in plant detection are its heavy reliance on labeled data, its subpar performance in small sample learning, and its limited capacity for model generalization. It is appropriate for plant disease classification problems because depth wise separable convolutions can efficiently decrease model complexity while retaining accuracy and speeding up detection.

REFERENCES

[1] Agarwal, M., Gupta, S. K., & Biswas, K. K. (2019, October). Grape disease identification using convolution neural network. In 2019 23rd International Computer Science and Engineering Conference (ICSEC) (pp. 224-229). IEEE. Wolfenson, K. D. M. (2013). Coping with the food and agriculture challenge: smallholders' agenda. Food and Agriculture Organization of the United Nations, Rome.

[2] Agarwal, M., Singh, A., Arjaria, S., Sinha, A., & Gupta, S. (2020). ToLeD: Tomato leaf disease detection using convolution neural network. Procedia Computer Science, 167, 293-301.

[3] Anandhakrishnan, T., & Murugaiyan, J. S. (2020). Identification of tomato leaf disease detection using pretrained deep convolutional neural network models. Scalable Computing: Practice and Experience, 21(4), 625-635.

[4] Ariyapadath, S. (2021). Plant Leaf Classification and Comparative Analysis of Combined Feature Set Using Machine Learning Techniques. Traitement du Signal, 38(6).

[5] Uddin AH, Chen Y-L, Borkatullah B, Khatun MS, Ferdous J, Mahmud P, Yang J, Ku CS, Por LY. Deep-Learning-Based Classification of Bangladeshi Medicinal Plants Using Neural Ensemble Models. Mathematics. 2023; 11(16):3504.

[6] Iman M, Arabnia HR, Rasheed K. A Review of Deep Transfer Learning and Recent Advancements. Technologies. 2023; 11(2):40.

[7] Sibiya, M., & Sumbwanyambe, M. (2019). A computational procedure for the recognition and classification of maize leaf diseases out of healthy leaves using convolutional neural networks. AgriEngineering, 1(1), 119-131.

[8] Zhang, K., Wu, Q., Liu, A., & Meng, X. (2018). Can deep learning identify tomato leaf disease? Advances in multimedia, 2018.

[9] Amara, J., Bouaziz, B., & Algergawy, A. (2017). A deep learning-based approach for banana leaf diseases classification.

[10] Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis. Computers and electronics in agriculture, 145, 311-318.

[11] Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J., & Hughes, D. P. (2017). Deep learning for image-based cassava disease detection. Frontiers in plant science, 8, 1852.

[12] Fujita, E., Kawasaki, Y., Uga, H., Kagiwada, S., & Iyatomi, H. (2016, December). Basic investigation on a robust and practical plant diagnostic system. In 2016 15th IEEE international conference on machine learning and applications (ICMLA) (pp. 989-992). IEEE.

[13] Durmuş, H., Güneş, E. O., & Kırcı, M. (2017, August). Disease detection on the leaves of the tomato plants by using deep learning. In 2017 6th International conference on agro-geoinformatics (pp. 1-

5). IEEE.

[14] Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A comparative study of fine-tuning deep learning models for plant disease identification. Computers and Electronics in Agriculture, 161, 272-279.

[15] Rangarajan, A. K., Purushothaman, R., & Ramesh, A. (2018). Tomato crop disease classification using pre-trained deep learning algorithm. Procedia computer science, 133, 1040-1047.

[16] Brahimi, M., Arsenovic, M., Laraba, S., Sladojevic, S., Boukhalfa, K., & Moussaoui, A. (2018). Deep learning for plant diseases: detection and saliency map visualization. Human and machine learning: Visible, explainable, trustworthy, and transparent, 93-117.

[17] Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D., & Stefanovic, D. (2016). Deep neural networks-based recognition of plant diseases by leaf image classification. Computational intelligence and neuroscience, 2016.

[18] Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. Frontiers in plant science, 7, 215232.

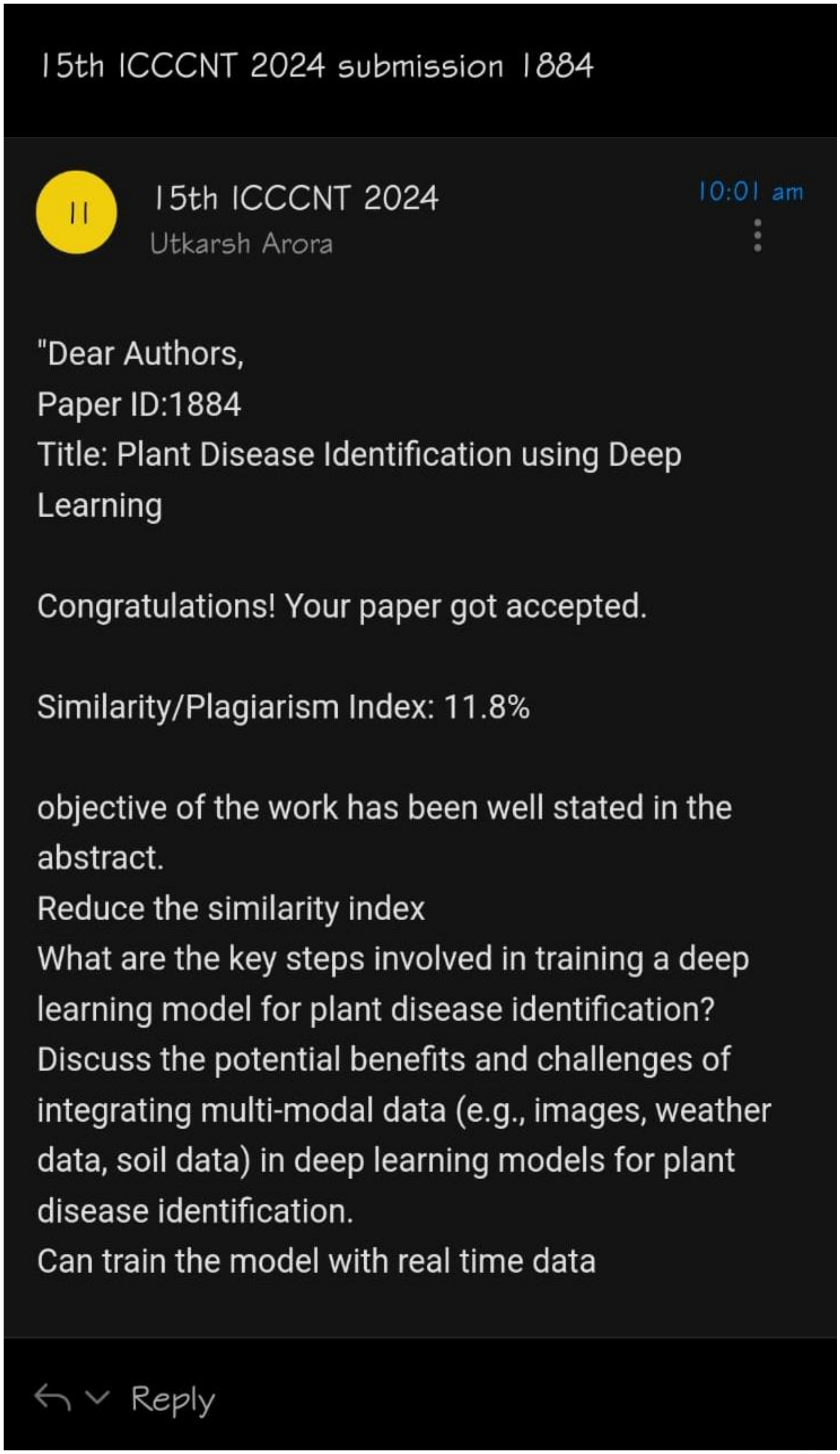
[19] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778)

[20] Liang, J., & Jiang, W. (2023). A ResNet50-DPA model for tomato leaf disease identification. Frontiers in Plant Science, 14, 1258658.

[21] Li, Z., Li, C., Deng, L., Fan, Y., Xiao, X., Ma, H., Qin, J., & Zhu, L. (2022). Improved AlexNet with Inception-V4 for Plant Disease Diagnosis. Computational intelligence and neuroscience, 2022, 5862600.

[22] Paymode, A. S., & Malode, V. B. (2022). Transfer learning for multi-crop leaf disease image classification using convolutional neural network VGG. Artificial Intelligence in Agriculture, 6, 23-33.

[23] Wagle, S. A. (2021). Comparison of Plant Leaf Classification Using Modified AlexNet and Support Vector Machine. Traitement du Signal, 38(1).



15th ICCCNT 2024 submission 1884

15th ICCCNT 2024
Utkarsh Arora
10:01 am

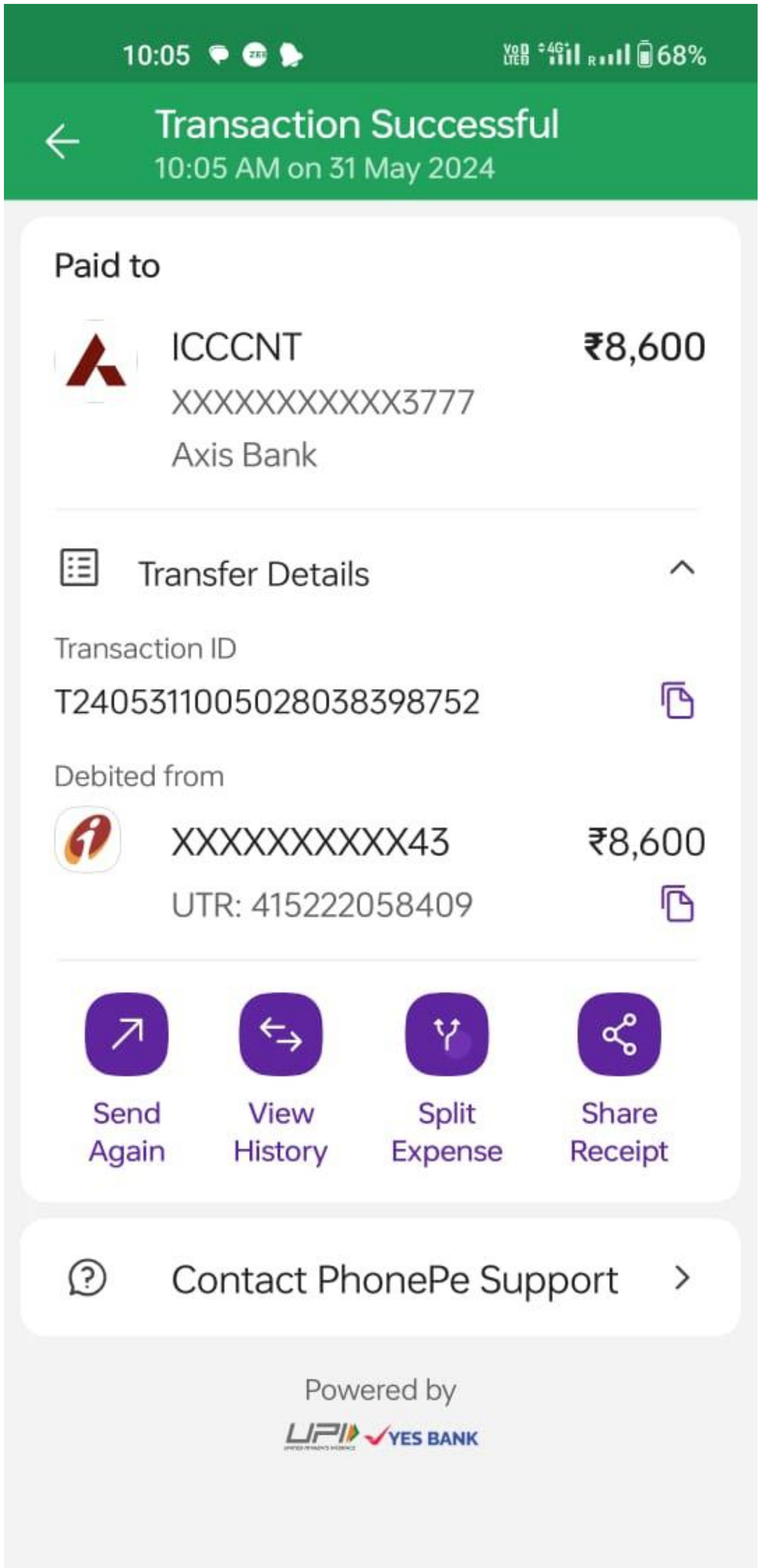
"Dear Authors,
Paper ID:1884
Title: Plant Disease Identification using Deep Learning

Congratulations! Your paper got accepted.

Similarity/Plagiarism Index: 11.8%

objective of the work has been well stated in the abstract.
Reduce the similarity index
What are the key steps involved in training a deep learning model for plant disease identification?
Discuss the potential benefits and challenges of integrating multi-modal data (e.g., images, weather data, soil data) in deep learning models for plant disease identification.
Can train the model with real time data

Reply


10:05
68%
Transaction Successful
10:05 AM on 31 May 2024
Paid to
ICCCNT
₹8,600
XXXXXXXXXXX3777
Axis Bank
Transfer Details
Transaction ID
T2405311005028038398752
Debited from
XXXXXXXXXX43
₹8,600
UTR: 415222058409
Send Again
View History
Split Expense
Share Receipt
Contact PhonePe Support
Powered by
UPI
YES BANK

APPENDIX-II

plag check

ORIGINALITY REPORT

6%	2%	1%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	sersc.org Internet Source	3%
2	Submitted to The Indian Institute Of Management And Engineering Society Student Paper	3%
3	Avnish Panwar, Siddharth Gupta, Nisha Chaube, Sonali Gupta, Akanksha Kapruwan. "Classification of Apple Leaf Diseases using Distinct Machine Learning Algorithms", 2022 IEEE Delhi Section Conference (DELCON), 2022 Publication	1%
4	medium.com Internet Source	1%
5	www.slideshare.net Internet Source	1%
6	Nitin Vamsi Dantu, Shriram K. Vasudevan, K. Vimalkumar. "An innovative artificial intelligence approach for disease classification in plants", International Journal	1%