



## Light-Weight Deep Learning Model for Accelerating the Classification of Mango-Leaf Disease

Bahar Uddin Mahmud <sup>1\*</sup> , Abdullah Al Mamun <sup>2</sup>, Md Jakir Hossen <sup>3\*</sup> ,  
Guan Yue Hong <sup>1</sup> , Busrat Jahan <sup>4</sup>

<sup>1</sup> Department of Computer Science, Western Michigan University, Kalamazoo, Michigan, United States.

<sup>2</sup> School of Information and Communication, Griffith University, Nathan 4111, Australia.

<sup>3</sup> Faculty Engineering and Technology, Multimedia University, Melaka 75450, Malaysia.

<sup>4</sup> Department of Computer Science and Engineering, Feni University, Feni 3900, Bangladesh.

### Abstract

Mango leaf diseases represent a serious threat to world agriculture, necessitating prompt and accurate detection to avert catastrophic effects. In response, this study suggests a light-weight, deep learning-based method for automatically classifying mango leaf diseases. The model is based on the original DenseNet architecture, which is well known for its effectiveness in image classification tasks. Custom layers have been added over the existing layer of the original DenseNet model. The proposed model has been compared with other existing pre-trained models. Based on comparisons, the proposed model, DenseNet78, proved to be efficient even on a relatively small dataset, where the conventional model failed. The proposed model ensured generalization across regions, disease variants, and diverse datasets of mango leaves. The results demonstrate that the fine-tuned DenseNet architecture (DenseNet78), along with an ideal growth rate, modifying block size, and a number of layers, provides optimum accuracy, with 99.47% accuracy in identifying healthy mango leaves and 99.44% accuracy in detecting various mango leaf diseases. The results also demonstrate that the model is effective in accelerating the training process because of careful comparative analysis of all the available alternatives, including the most effective combination of optimizers, learning rate schedulers, and loss functions. The study's conclusion is an automated approach for diagnosing mango leaf disease using an improved and optimized DenseNet architecture (DenseNet78).

### Keywords:

Image Classification;  
Custom DenseNet;  
Deep Learning;  
Computer Vision;  
Leaf Disease;  
Disease Classification.

### Article History:

<b>Received:</b>	22	October	2023
<b>Revised:</b>	09	January	2024
<b>Accepted:</b>	21	January	2024
<b>Published:</b>	01	February	2024

## 1- Introduction

Mango cultivation has a significant role in the world economy, but the presence of diseases presents significant risks, which require precise and prompt diagnosis. Previous studies have utilized machine learning techniques to detect diseases in mangoes. However, they had difficulties dealing with variations in images and achieving generalization, which we discussed in the related work section. Recent advancements in deep learning show potential; however, they mostly rely on intricate structures that demand large amounts of data and resources, which restricts their accessibility. Over 90% accuracy is achieved by models such as ResNet50, DenseNet201, and feed-forward networks; however, these models come with millions of parameters and require longer training times. This restricts their adoption and accessibility for numerous practical applications.

\* **CONTACT:** baharuddin.mahmud@wmich.edu; jakir.hossen@mmu.edu.my

**DOI:** <http://dx.doi.org/10.28991/ESJ-2024-08-01-03>

© 2024 by the authors. Licensee ESJ, Italy. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC-BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Using pre-designed architectures such as DenseNet, ResNet, EfficientNet, etc. is common practice in computer vision. These models often possess a substantial number of trainable parameters, necessitating a sizable dataset to determine the ideal values for such parameters. Therefore, these models are typically trained on a huge dataset, such as ImageNet. Subsequently, the acquired weights are employed for transfer learning. Attaining a high level of accuracy is based on the second dataset sharing a similar domain with the first dataset. However, when the domain is different, the models may have difficulties achieving satisfactory accuracy on small datasets. Furthermore, the pre-designed structures typically consist of complex models that require extensive computations, decreasing prediction speed. In order to meet the requirements of real-time applications, it is crucial to ensure that the prediction process is optimized for maximum speed. One may ask if it is consistently advantageous to opt for transfer learning or if lightweight custom architectures can attain satisfactory performance.

This paper suggests a simplified deep-learning method that utilizes a customized DenseNet structure for the categorization of mango diseases. Our model achieves state-of-the-art accuracy while minimizing complexity by using a shallow depth. The adaptations enhance the speed of training while maintaining high performance, making it easier to deploy, especially in situations where there are constraints on datasets and resources. Specifically, this work makes the following contributions:

- Proposes a simplified and more efficient DenseNet-based model with reduced parameters and shallower depth.
- Although the model complexity has been greatly decreased, it achieves accuracy comparable to top models.
- Uses selective parameter adjustment to shorten training times, which is advantageous for applications with limited resources.
- Examine extensive combinations to find the best ratio of precision to productivity.

## 2- Literature Review

Various techniques and methods have been established to detect plant leaf diseases. These are generally characterized by disease finding or disease detection methods and disease sorting or disease classification methods. Many techniques use segmentation, feature fusion, and image classification implemented on cotton, strawberry, mango, tomato, rice, sugarcane, and citrus. Similarly, these methods are appropriate for leaf, flower, and fruit diseases because they use proper segmentation [1], feature extraction, and classification [2]. In recent years, several solutions for the automatic diagnosis of mango leaf diseases have been proposed. These solutions can be divided into two categories: those based on classical ML algorithms on the one hand, and those based on DL on the other. Mango fruit is important in the agricultural sector due to its massive production volume in several parts of the world, especially in the Indian subcontinent. Therefore, several approaches for the detection and identification of mango plant leaf diseases have been proposed to prevent a loss of harvest. A technique to identify and categorize diseases based on crops and weeds was introduced by Le et al. [3].

In order to reduce the noise in the input image, morphological opening and closing procedures were first carried out. The features from the processed sample were then computed using a tailored framework called the filtered local binary pattern technique with contour mask and coefficient  $k$  (k-FLBPCM). The SVM classifier was trained to categorize various plant diseases using the retrieved features. Although the method in the research showed improved accuracy in classifying plant diseases, it may not perform well when applied to data with perspective distortions. Ramesh et al. [4] presented a method for classifying plant diseases. The data were divided into healthy and diseased groups using features from the Histogram of Oriented Gradients (HOGs), which were then used to train the Random Forest (RF) classifier. Although the method is reliable for classifying plant diseases, performance still has to be enhanced. Kuricheti et al. [5] developed a system for categorizing illnesses caused by turmeric leaves. After preprocessing, the K-means method was used to conduct image segmentation on the input image. Following the application of the GLCM technique for feature extraction, the SVM classifier was trained to classify the leaves. This method shows superior results for classifying plant diseases but is unable to perform better on samples with significant brightness changes. By taking into account the vein pattern of the mango sick leaf, Saleem et al. [6] suggested a unique segmentation approach to segment the diseased region. In this study, powdery mildew and sooty mold are diseases that are treated. Canonical correlation analysis (CCA)-based fusion was used to extract the leaf's attributes based on color and texture.

To identify leaf illnesses, they tested ten different classifiers, but the cubic SVM classifier (95.5% accuracy) produced the best results. The time required to identify diseased leaves is considerable, despite the fact that very little data was used in the process. Furthermore, real-time identification of disorders that have been treated is not possible with the suggested design. A method for the automatic recognition and categorization of mango leaf diseases was proposed by Srunitha et al. [7]. The authors employed GLCM (gray-level color co-occurrence metrics) for feature extraction, k-means clustering for image segmentation, and SVM for disease classification. The accuracy of the SVM classifier is up to 96%. One of the shortcomings of the suggested model is the difficulty of image segmentation due to the numerous diseases present in the same area of a leaf. The variety in leaf color, texture, and form also complicated the feature selection

process. Mia et al. [8] applied K-means clustering for picture segmentation as well as SVM and Neural Network Ensemble (NNE) to detect and identify the symptoms of four mango leaf diseases: Dag disease, Golmachi disease, Shutimold disease, and Red Moricha disease. The proposed method may detect and diagnose diseases with up to 80% accuracy. However, the authors believe that boosting the training data would improve the model's performance.

Ullagaddi & Raju [9] suggested an Artificial Neural Network (ANN) integrated with a Modified Rotation Kernel Transformation (MRKT) approach based on directed feature extraction (shape, color, or other misleading aspects) for mango disease recognition and classification. Using the proposed MRKT, they achieved an accuracy of 98%. The results gained are quite substantial, yet the proposed system is only applicable to one ailment. They proposed an enhanced Ullagaddi & Raju [10] to more accurately recognize diseases. The authors present an improved Wavelet-PCA-based Statistical Feature Extraction technique for plant disease identification using MRKT-based directional characteristics. Powdery Mildew and Anthracnose sickness were both diagnosed using an ANN. They achieved 98.50%, 98.70%, and 98.75% accuracy for flowers, leaves, and fruits, respectively. From the ML-based approaches that have been addressed above, it can be inferred that these works are straightforward to implement but necessitate considerable training data and rely heavily on human experience. Additionally, these methods are not resistant to the wide differences in the dimensions, hues, and forms of leaf plant diseases. In order to increase the accuracy of recognizing various Plant leaf diseases, a more comprehensive method utilizing the most recent approaches is therefore required [11].

Recently, DL approaches have been highly explored in several automated applications because of their high accuracy rate [12]. To identify and classify the tomato crop disease, Agarwal et al. [13] suggested a CNN-based architecture. Three convolutional layers and a max-pooling layer were used in this technique to separate the important information from the input samples and categorize it. Although this method performs more accurately when classifying tomato diseases, it has the drawback of over-fitting to a limited number of classes. A mobile app-based method to identify and categorize maize crop disease was introduced by Richey et al. [14]. To extract the deep key points from the input images and categorize them into the appropriate categories, the ImageNet database was used to train a DL-based model called ResNet50. The method offers a mobile phone-based approach to crop disease classification, but due to limitations in memory, processing, and battery capacity, this method is computationally demanding and not well suited for cell phones. A strategy to recognize and classify tomato leaf disease at an early stage was put out by Batool et al. [15]. The AlexNet framework was utilized in the initial stage to extract the deep key points from the input sample, which were later used to train the KNN to classify the images as healthy or impacted. This method exhibits higher classification accuracy, although KNN is a time-consuming and slow algorithm. A DL-based framework for localizing and classifying tomato leaf diseases was introduced by Tm et al. [16]. Before using the input samples for additional processing, they were first downsized. The samples were then separated into classes of healthy and afflicted samples using a DL model called LeNet. The method offers a low-cost method for classifying diseases in tomato crops; however, it is unable to demonstrate robust performance for noisy samples.

Turkoglu et al. [17] developed an ensemble technique in which the deep key points of various plants were computed using a number of DL-based models, including AlexNet, GoogleNet, DenseNet201, ResNet50, and ResNet101 frameworks. The computed characteristics were then used for SVM training to classify various plant diseases in the following step. The method performs better at classifying plant leaves, but at the expense of more expensive feature computation. For the automatic detection and identification of the four mango leaf diseases, anthracnose, powdery mildew, red rust, and golmich, Gulavnai & Patil [18] proposed ResNet-CNNs (ResNet18, ResNet34, and ResNet50) in conjunction with Transfer Learning (TL). Results indicate that ResNet50 performs better, with an accuracy of 91.50%. A Feed-Forward Neural Network (FFNN) with Hybrid Metaheuristic Feature Selection (HMFS) was used by Pham et al. [19] to classify the three mango diseases: anthracnose, gall midge, and powdery mildew.

The suggested model outperformed comparable CNNs like AlexNet (78.64%), VGG16 (79.92%), and ResNet (84.88%) in accuracy by 89.41%. A Multilayer Convolutional Neural Network (MCNN) model is utilized in the study of Singh et al. [20] to categorize mango leaves that have anthracnose fungal disease. The authors combined real-time collected mango leaf images with photographs of mango leaves from the Plant Village dataset. According to experimental results, sick leaves could be classified with 97.13% accuracy. However, images captured in actual conditions suffer greatly from the issues of temperature variation, shadowing, leaf overlapping, and the presence of several things. An innovative classification scheme for the diseases Anthracnose, Bacterial Black Spot, and Sooty Mold was put up by Prabu et al. [21]. For better feature selection, they employed a CNN with crossover-based Levy flight distribution, a MobileNetV2 model for the learning stage, and an SVM model to classify disorders. The experimental findings demonstrate classification performance in comparison to other cutting-edge techniques. A CNN model based on the AlexNet architecture was proposed by Wongsila et al. [22] for the detection of anthracnose mango leaf disease. The system is created using the Tensor Flow framework and a dataset of mango images taken with a CDD camera in real-world settings. More than 70% of the time, the developed approach successfully isolated unhealthy mango leaves. The model is particular to anthracnose and is based on data collected under actual field settings.

### 3- Proposed Methodology

#### 3-1-Introduction to DenseNet

DenseNet, or "Densely Connected Convolutional Networks," is a powerful deep-learning architecture that was introduced by Huang et al. (2017) [23]. DenseNet's dense connection design encourages feature reuse and efficient information flow because it allows each layer to directly input data from all layers above it. The architecture is composed of dense blocks, intermediary levels, and a top classification layer. Dense blocks are composed of a succession of batch normalization, ReLU activation, and  $3 \times 3$  convolutional layers to encourage dense connections between layers. Transition layers reduce the number of channels and the volume of the space to improve computing performance. There are 121 layers in a specific form known as DenseNet-121, encompassing four dense blocks with [6, 12, 16, 23] levels, transition layers, and other components. This architecture has demonstrated excellent performance on a number of computer vision applications and delivers parameter efficiency, high feature reuse, and cutting-edge outcomes while maintaining a balance between model size and accuracy.

DenseNet has established itself as a significant milestone in deep learning and has drawn substantial interest as a result of its alluring advantages. Dense connections prevent vanishing gradients and encourage gradient flow, enabling efficient training of extremely deep networks. By encouraging feature reuse, which reduces the likelihood of overfitting, DenseNet increases robustness with limited training data. Because there are not any fully connected layers, the model also uses less memory because there are fewer parameters. DenseNet-121 maintains a balance between computational complexity and depth with 121 layers, making it a popular choice for many real-world applications. Appreciating its capability to capture complicated features and state-of-the-art performance on benchmark datasets, it has evolved into a standard architecture for transfer learning and feature extraction across a range of disciplines. Further variants of DenseNet, such as DenseNet-169 and DenseNet-201, which provide even more potent tools for tackling difficult visual identification challenges, were also inspired by DenseNet. Overall, DenseNet has benefited the fields of deep learning and computer vision, promoting more research and advancements in the area. The goal of DenseNet was to overcome some of the drawbacks associated with conventional deep neural network topologies, including the vanishing gradient issue and the requirement for a large number of parameters in order to attain high accuracy. With DenseNet, we hope to boost network performance, promote feature reuse, and facilitate effective training. The main features of DenseNet are:

**Dense Connections:** The dense connectivity pattern between layers is the main innovation in DenseNet. The output of one layer is concatenated to the input of the following layer in conventional convolutional networks like VGG or ResNet. However, in DenseNet, every layer in the network receives direct inputs from all layers that came before it. Every layer can easily access and use the feature maps produced at all preceding layers thanks to this tight interconnectedness. As a result, feature reuse is strongly promoted, resulting in networks that are more effective and information-rich.

**Dense Blocks:** There are numerous dense blocks that makeup DenseNet. A collection of successive layers with close connections to one another is referred to as a dense block. Each layer within a dense block receives the concatenated feature maps from all preceding layers within that block as input. The secret to dense connections is this concatenation.

**Transition Layers:** Between dense blocks, DenseNet adds transition layers. These transition layers have two functions: they minimize the width and height of feature maps, which lessens their spatial dimensions, and they compress the number of channels, which lessens the computational load. A batch normalization layer is usually the first in a transition layer stack, followed by a  $1 \times 1$  convolutional layer and average pooling.

**Growth Rate:** In DenseNet, the growth rate controls how many new feature maps are added to the feature maps of succeeding layers by each layer in a dense block. A network is more expressive when there are more feature maps and a faster growth rate. However, this also results in an increase in memory usage and parameter count.

DenseNet has advantages; however, because of its dense connections, it could need more memory, which makes it less suitable for situations or devices with limited resources. In recent years, deep learning techniques have made significant advancements in various fields, including computer vision and medical imaging. One such powerful architecture is DenseNet (Densely Connected Convolutional Networks), which has demonstrated exceptional performance in image classification tasks. This paper presents the application of DenseNet for the classification of mango leaf diseases. The dataset consists of 16,000 images of mango leaves, each categorized into one of eight disease classes.

The primary objective of this study is to leverage the strength of DenseNet to accurately identify the disease type based on the leaf images, enabling timely and precise disease diagnosis in mango crops. The proposed DenseNet architecture is designed to capture intricate patterns and representations from the input images while promoting feature reuse across layers. The model comprises multiple dense blocks, each containing multiple convolutional layers that are densely interconnected with each other. Each dense block is followed by a transition block, which reduces spatial dimensions and controls the number of feature maps, facilitating smooth information flow across the network. Table 1 shows the comparative analysis of our model's settings with different variations of the DenseNet model.

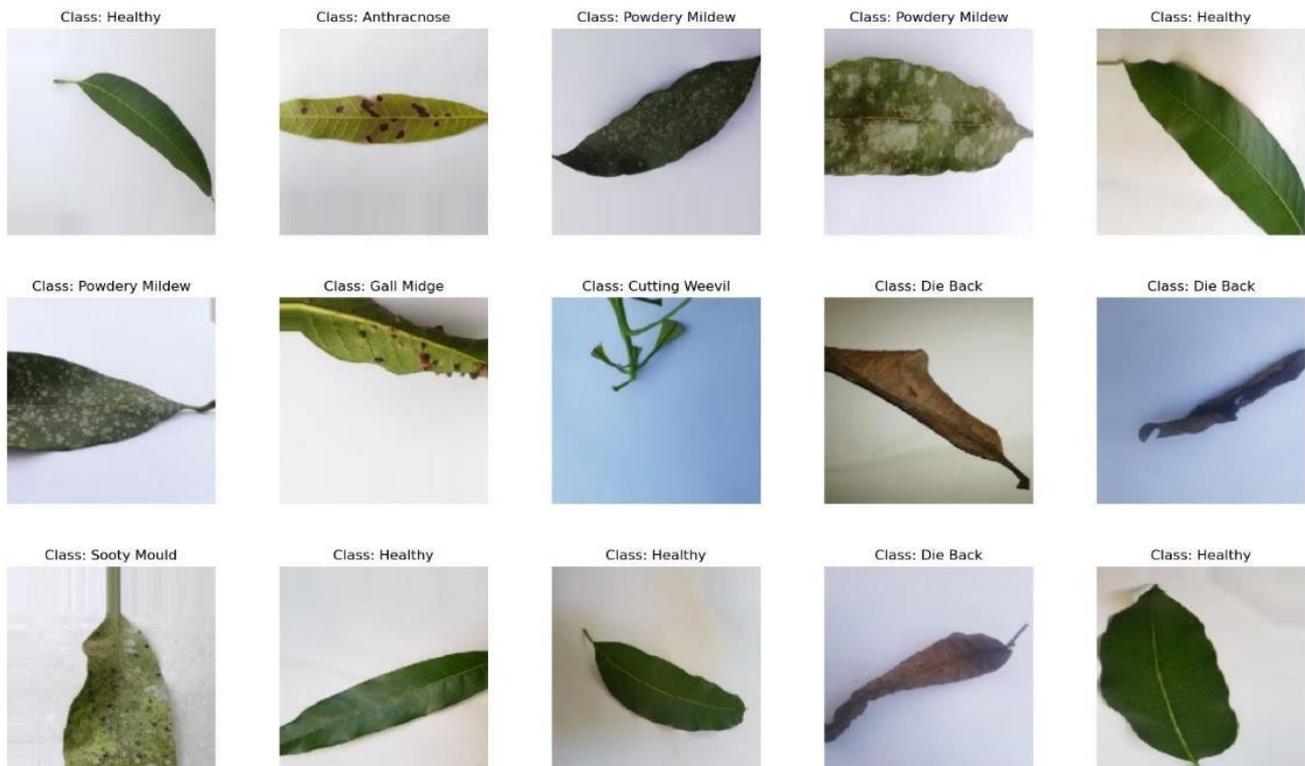
**Table 1. The comparison of different DenseNet Architectures and proposed model. This comparison is based on dataset size and type of task**

Feature	DenseNet121	DenseNet169	DenseNet201	Our Model (DenseNet78)
Dense blocks	4	4	4	3
Layers in dense block	58	82	98	16
Number of parameters	8.1M	14.3M	20.2M	0.1M
Training time	Slower	Slower	Slower	Faster
Computational resources	More	More	More	Less
Accuracy	High	High	High	Similar to the state-of-the-art model
Pretrained weights	ImageNet	ImageNet	ImageNet	No pretrained weights

The comparison of different DenseNet architectures reveals distinct characteristics. DenseNet-121, DenseNet-169, and DenseNet-201 all share 4 dense blocks, but they differ in layers per block (58, 82, and 98, respectively) and the number of parameters (8.1M, 14.3M, and 20.2M, respectively). In contrast, our customized DenseNet stands out with 3 dense blocks and a significantly shallower architecture, featuring only 16 layers per block and 143,984 parameters. This customization leads to faster training and requires fewer computational resources. While DenseNet-121, DenseNet-169, and DenseNet-201 come with pre-trained weights from ImageNet, our model does not utilize such pre-training. Despite its shallower depth, the customized DenseNet achieves higher accuracy, likely due to its tailored design for the specific task or dataset. The choice of which DenseNet architecture to employ depends on the task's complexity, available resources, and dataset size, with the customized approach offering advantages in accuracy and efficiency.

### 3-2-Dataset Processing

The original dataset contains 4000 images of about 1800 distinct leaves covering seven diseases [24]. The images are collected from four mango orchards in Bangladesh, one of the top mango-growing countries in the world. Data augmentation techniques have been applied to increase the size of the dataset. Dataset quality and quantity have been improved by applying several data augmentation techniques. Finally, the number of images becomes 16,000 after applying augmentation. Figure 1 demonstrates the dataset, which consists of 7 disease classes and healthy leaves.



**Figure 1. Visualization of dataset images and their corresponding classes. There is a total of 8 classes representing seven diseases category and a healthy leaf category**

### 3-3- Color Skew

This augmentation randomly adjusts the hue, saturation, and brightness of the image by multiplying each channel by a randomly chosen coefficient. The coefficients are chosen from a range of [0:6;1:4] to ensure that the resulting image is not too distorted. When using color skew augmentation, it is essential to apply the same augmentation to both the original image and its corresponding label (if available) to maintain consistency between the input data and target labels during training.

### 3-4- Rotation

Rotation augmentation is a data augmentation technique used in computer vision tasks to increase the diversity of training data by rotating images. It involves applying random rotations to images within a specified range. By doing so, the model becomes more robust to different orientations of objects, which can lead to improved performance and better generalization to unseen examples. For each image in the training dataset, a random rotation angle is generated within a predefined range. The rotation angle can be positive or negative, indicating the clockwise or counterclockwise direction, respectively. During rotation, the new pixel positions might not correspond to integer coordinates in the original image. Interpolation techniques like bilinear interpolation or nearest-neighbor interpolation are used to obtain pixel values at non-integer coordinates. These interpolation methods estimate the pixel values based on neighboring pixels, resulting in a smooth and visually coherent rotated image. Let us represent the original image coordinates as  $(a, b)$  and the rotated image coordinates as  $(a\_rotation, b\_rotation)$  after applying a rotation of angle  $\theta$  around a point  $(a_0, b_0)$ . The formula for 2D image rotation around a point  $(a_0, b_0)$  is given by:

$$a\_rotation = \cos(\theta) \times (a - a_0) - \sin(\theta) \times (b - b_0) + a_0 \quad (1)$$

$$b\_rotation = \sin(\theta) \times (a - a_0) + \cos(\theta) \times (b - b_0) + b_0 \quad (2)$$

### 3-5- Flip

Flip augmentation is a common data augmentation technique used in computer vision tasks to increase the diversity of training data. It involves flipping the images horizontally and/or vertically, creating mirrored versions of the original images. Flip augmentation is particularly useful when left-right or up-down symmetry is present in the data or when the model needs to be invariant to such transformations. The flip augmentation can be applied randomly during training by choosing whether to perform horizontal, vertical, or both flips with equal probability. By applying flip augmentation, the model learns to recognize objects regardless of their orientation and position in the image. This helps improve the model's robustness and generalization to images captured from various viewpoints and angles. In horizontal flip augmentation, the image is flipped along the vertical axis. The transformation can be represented using the following mathematical equation:

$$I\_flip(a, b) = I(width - 1 - a, b) \quad (3)$$

In vertical flip augmentation, the image is flipped along the horizontal axis. The transformation can be represented using the following mathematical equation:

$$I\_flip(a, b) = I(a, height - 1 - b) \quad (4)$$

### 3-6- Adding Noise

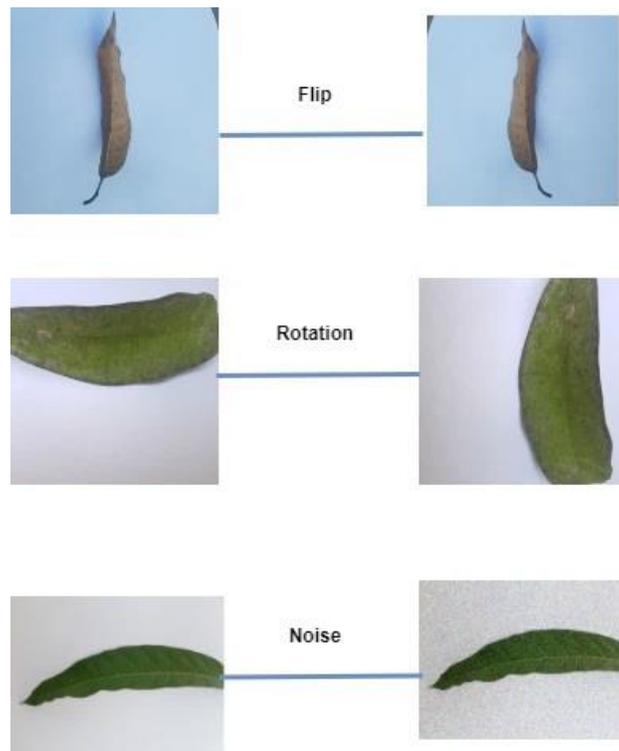
Gaussian noise is a type of random noise that is added to an image to simulate real-world variations and increase the robustness of models during training. It is named after the Gaussian distribution, also known as the normal distribution, which characterizes its statistical properties. Gaussian noise is typically represented as random values sampled from a Gaussian distribution with a mean of 0 and a specified standard deviation ( $\sigma$ ). The noise is added to each pixel in the image, altering its intensity by a random amount. This results in a subtle yet effective perturbation of the image, making it more robust and less sensitive to small variations. The formula for adding Gaussian noise to an image is as follows:

$$I\_noise(a, b) = I(a, b) + N(0, \sigma) \quad (5)$$

Where,

- $I\_noise(a, b)$  is the pixel value of the noisy image at position  $(a, b)$ .
- $I(a, b)$  is the original pixel value at position  $(a, b)$ .
- $N(0, \sigma)$  is a random value sampled from a Gaussian distribution with a mean of 0 and a standard deviation of  $\sigma$ .

Figure 2 represents the original images and the augmented images. Left-side images are original images, and right-side images are generated using the above-mentioned augmentation techniques.



**Figure 2.** Represents the augmented samples of the original dataset. The left side images are the original image and the right-side images are the results of augmentation

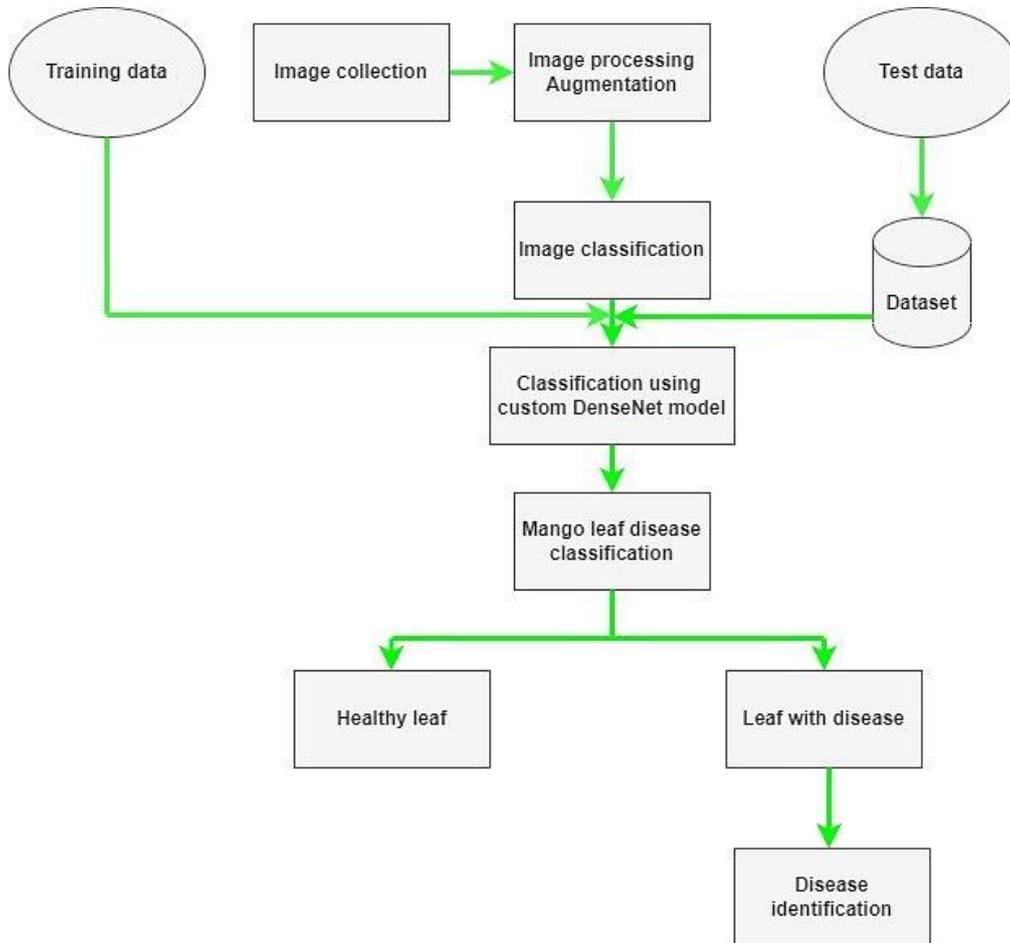
### 3-7- Model Architecture

The model architecture proposed for mango leaf disease classification is a custom DenseNet-like neural network designed to extract and capture relevant features from input images efficiently. The model starts with an input layer that takes RGB images of size  $224 \times 224$  pixels. The initial preprocessing involves a convolutional layer with 24 filters of size  $3 \times 3$ , followed by batch normalization and ReLU activation. To facilitate feature extraction, a dense block is constructed with three repetitions of four custom convolutional layers within each repetition. Each convolutional layer within the dense block has a filter size of  $3 \times 3$  and a growth rate of 12. The growth rate controls the number of filters to be added to subsequent layers, and the output from each convolutional layer is densely connected by concatenating it with the input of that block. Following each dense block, a custom transition block is introduced to reduce spatial dimensions and down sample information. The transition block consists of batch normalization, ReLU activation, and a  $1 \times 1$  convolutional layer with a growth rate of 12. Average pooling with a  $2 \times 2$  kernel and stride of 2 is then applied. This pattern of dense blocks and transition blocks is repeated to deepen the model's capacity and allow for more robust feature learning. The last dense block is followed by batch normalization, ReLU activation, and global average pooling to condense spatial information into a compact representation.

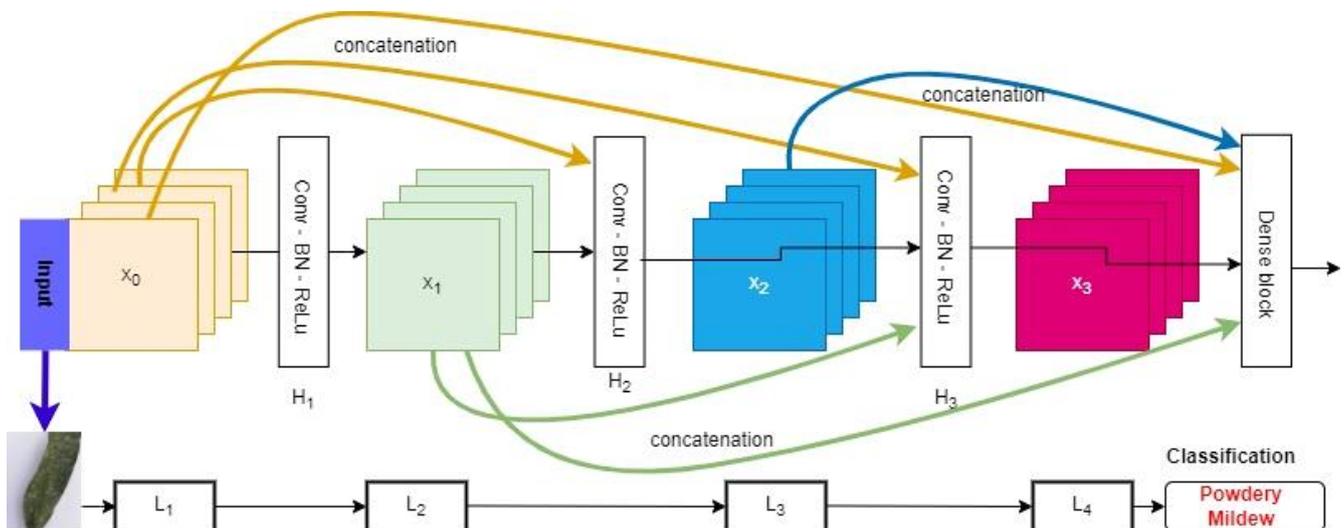
Finally, the output is generated using a dense layer with softmax activation, comprising eight units corresponding to the eight disease classes. The custom layers, including the dense and transition blocks, are designed to enhance the model's ability to capture intricate patterns in mango leaf images efficiently. The model is optimized using the Adam optimizer with a learning rate of 0.0001 and a binary cross-entropy loss function. The custom layer architecture draws inspiration from DenseNet, utilizing dense connectivity for better feature reuse and memory efficiency compared to traditional architectures like VGG and ResNet. Figure 3 represents the flow diagram of our model that includes the steps we performed sequentially.

Figure 4 is the visualization of our proposed model. It includes an input image that goes through the multiple conv layers and finally generates the respective classification. Figure 5 describes the flow of dense block and shows how the features of the previous layer concatenated with the next layer. The input layer of the model architecture is first created to receive images. An early convolutional block extracts features from the input images after the input layer. After that, a unique "dense block" with customized convolutional layers is presented. Concatenation creates a dense

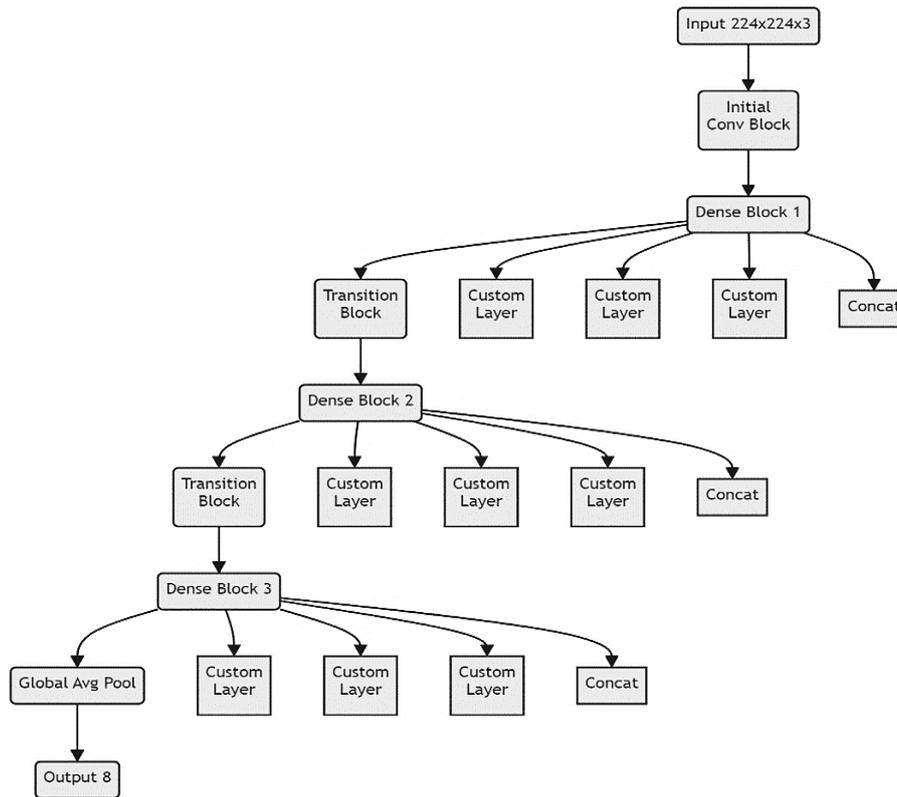
connection between the outputs from various layers, facilitating complex feature extraction and information flow. A "transition block" is used to efficiently lower spatial dimensions after every dense block. The model also makes use of the stacking of several transition and dense blocks to create a hierarchical structure that supports reliable feature extraction. Finally, global average pooling is used to the retrieved features to create a more compact representation, completing the architecture. Ultimately, an output layer processes the compressed features in preparation for the final classification task.



**Figure 3. Flow diagram of model overall activities. It includes image collection, augmentation, processing, the implication of custom model, classification, and verification**



**Figure 4. Pictorial Structures of the Proposed model that is based on the original DenseNet model. The visualization shows the dense block, number of layers, etc.**



**Figure 5.** This figure represents the flow of our custom architecture, which is built based on original DenseNet architecture

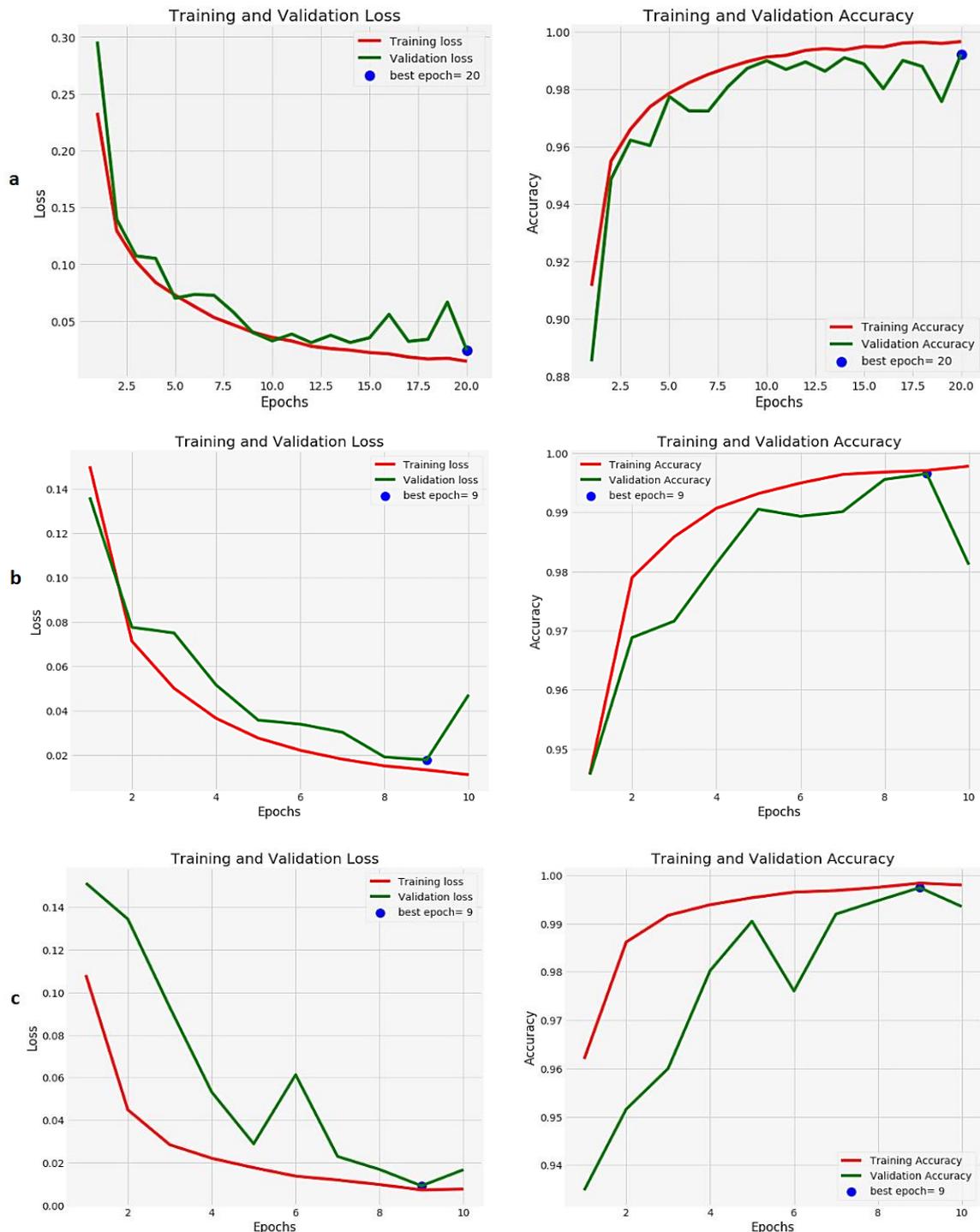
#### 4- Results and Discussions

Table 2, which is given below, compares model settings based on various parameter combinations. The number of layers, the pace of growth, and the size of the dense blocks vary amongst these designs. A study of the data reveals definite trends in terms of general parameters and accuracy measures. For instance, when employing a relatively conservative design with a dense block size of 3, 4 layers in each block, and a growth rate of 12, model 1 yields reasonable results. It has a high training accuracy of 99.47% and a stunning test accuracy of 99.17%. Model 2 becomes more complex, with 535,088 total parameters when the growth rate is increased to 24 while preserving the same structure of dense blocks and layers. The training accuracy remains high at 99.28%, but the test accuracy only slightly declines to 98.39%, indicating overfitting. It's interesting to observe that model 3, which has more layers (12) and a bigger, denser block size (4) but a slower growth rate of 12, nevertheless performs admirably. It manages to attain 99.47% training accuracy and 99.44% test accuracy despite having 1,990,160 parameters. With 502,256 total parameters, the optimum configuration preserves the dense block size at 3, increases the number of layers to 8, and keeps the growth rate at 12. Model 3's 99.54% training accuracy and 99.42% test accuracy draw attention to the intricate relationship between accuracy and parameters. The comparison emphasizes how important parameter tweaking is for increasing model complexity and effectiveness on the provided dataset. Figures 6(a), (b), and (c) show the training validation loss and training validation accuracy of our model in terms of different setups, respectively.

While previous research has made strides in addressing agricultural challenges, a literature review reveals significant gaps in the context of mango leaf disease diagnosis. Existing methods lack efficiency, especially in regions with limited resources. This study aimed to bridge this gap by proposing an innovative deep-learning model for the rapid and accurate classification of mango leaf diseases. Our research builds upon recent advancements in computer vision and deep learning. We introduced an improved lightweight DenseNet-based classification model designed specifically for mango leaf diseases. By addressing the limitations of existing models, our approach aimed to revolutionize the diagnosis process, offering a quicker and more accurate alternative. This paper contributes a novel approach to mango leaf disease diagnosis and presents a detailed comparative analysis with existing methods. The proposed model, characterized by its reduced parameters, enhanced efficiency, and applicability to small datasets, represents a significant advancement in the field. Through this research, we seek to provide a valuable tool for mango cultivators and plant pathologists, enabling them to make informed decisions for disease management and ultimately contributing to the sustainability of mango production globally."

**Table 2.** Represents the different combinations and fine-tuning of our proposed model, which includes variations in a number of dense block sizes, layers in the block, growth rate, parameters, and corresponding accuracy

Models	Dense block size + Layer in block + Growth rate	Total params	Training accuracy	Test accuracy
M1	3+4+12	143,984	0.9947	0.9917
M2	3+4+24	535,088	0.9928	0.9839
M3	4+12+12	1,990,160	0.9947	0.9944
M4	3+8+12	502,256	0.9954	0.9942



**Figure 6.** (a) Represents the visualization of loss and accuracy when the combination of dense block size, layers in the block, growth rate, and params are, respectively, 3,4,12 and 143,984. (b) Represents the visualization of loss and accuracy when the combination of dense block size, layers in the block, growth rate, and params are 3,4,24 and 535,088, respectively. (c) Represents the visualization of loss and accuracy when the combination of dense block size, layers in the block, growth rate, and params are 4,12,12 and 1.99M, respectively.

**4-1-Evaluation Metrics**

Evaluation indicators, such as Precision, Recall, and F1 Score correctness, are computed from the performance measurement matrices. A common setup to communicate forecast accuracy is the performance matrix. It is an N×N matrix with the original classes from the dataset in each column and each row representing the predicted classes. The depicted matrices provide the values for True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), and the evaluation parameters are generated as in the equation:

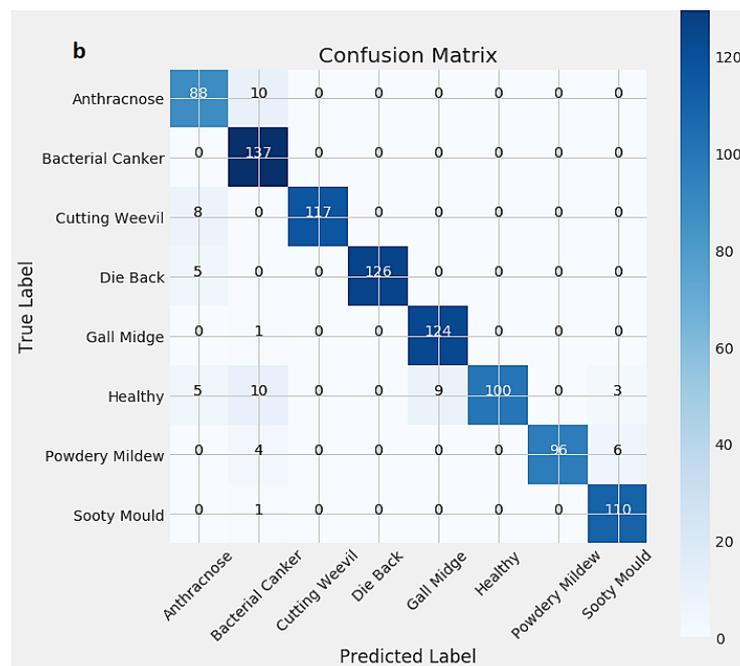
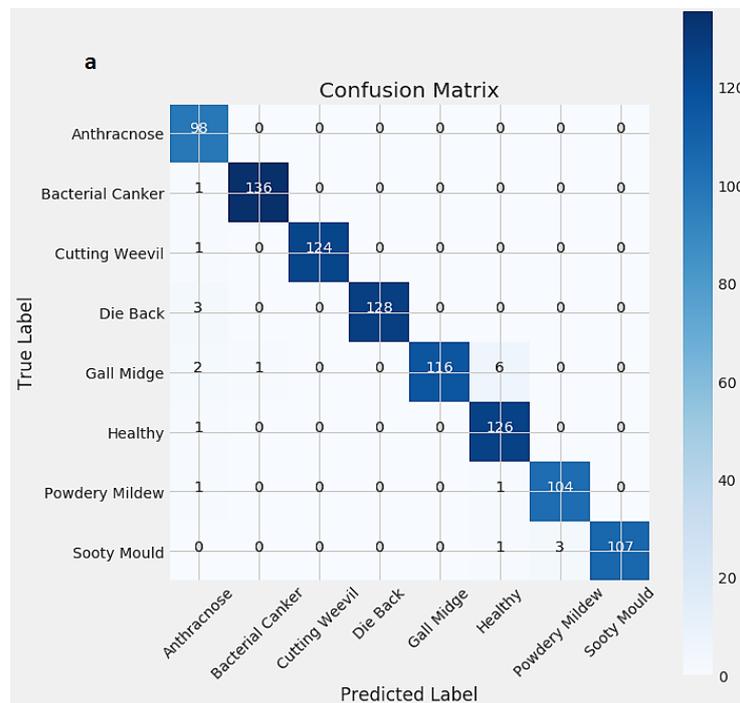
$$Precision = TP / TP + FP \tag{6}$$

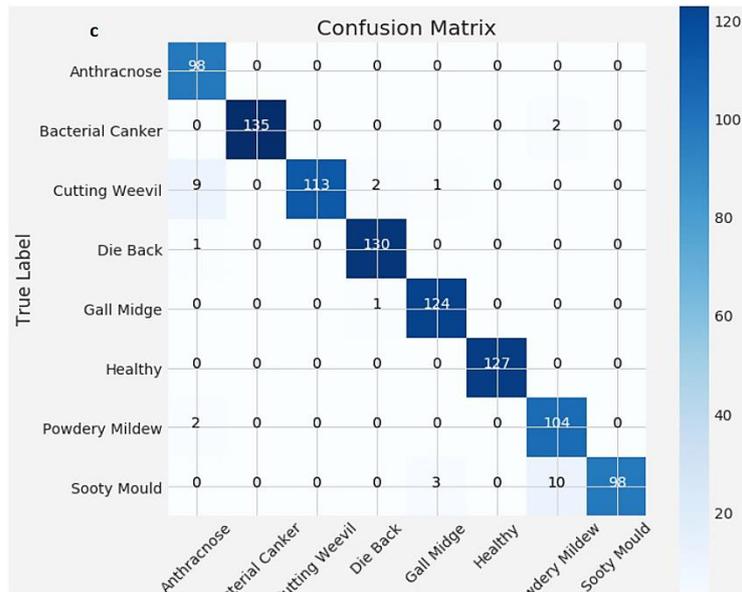
$$Recall = TP / TP + FN \tag{7}$$

$$F1Score = 2 \times Precision \times Recall / Precision + Recall \tag{8}$$

$$Accuracy = TP + TN / TP + FP + FN + TN + FN \tag{9}$$

Figures 7a, 7b and 7c show the performance of our model in terms of accurate classification in different parameter setups. Table 3 provides a thorough analysis of the effectiveness of a multi-class classification model across multiple classes.





**Figure 7.** (a) Represents the visualization of confusion matrix when the combination of dense block size, layers in block, growth rate, and params are respectively 3,4,12 and 143,984. (b) Represents the visualization of confusion matrix when the combination of dense block size, layers in block, growth rate and params are respectively 3,4,24 and 535,088. (c) Represents the visualization of confusion matrix when the combination of dense block size, layers in block, growth rate and params are respectively 4,12,12 and 1.99M.

**Table 3.** Calculation of precision, recall, f1 score, and support of corresponding disease class

Class	Precision	Recall	F1 Score	Support
Anthracnose	0.92	0.99	0.96	98
Bacterial Canker	0.99	0.98	0.99	137
Cutting Weevil	0.99	0.98	0.99	125
Die Back	0.97	0.99	0.98	131
Gall Midge	0.97	0.92	0.95	125
Healthy	1.00	0.98	0.99	127
Powdery Mildew	0.95	1.00	0.98	106
Sooty Mould	0.98	0.96	0.97	111

Precision, Recall, F1 Score, and Support are crucial metrics that are used to evaluate each class, including "Anthracnose," "Bacterial Canker," "Cutting Weevil," and others. Precision describes how well a model predicts successful occurrences within a class. In particular, the "Healthy" class achieves a precision score of 1.00, which is excellent and demonstrates precise positive predictions. Recall, which measures the model's capacity to find real positives, exhibits remarkable values like 1.00 for "Powdery Mildew," signifying accurate identification of occurrences falling under this category. The total effectiveness of the model in classifying cases is highlighted by the F1 Score, a measure of precision and recall that strikes a balance. Notable classes include "Bacterial Canker" and "Cutting Weevil," both of which received high F1 Scores of 0.99, demonstrating a harmonic balance between precision and recall. The Support column shows how many examples of each class there are overall, ranging from 98 occurrences for "Anthracnose" to 137 instances for "Bacterial Canker." In conclusion, this thorough table provides a useful tool for evaluating the model's classification abilities across several classes, illuminating its precision and efficacy in identifying and classifying occurrences inside particular categories.

#### 4-2- Comparison with State of the Art Model

With each model having a unique set of parameter configurations and weights derived from ImageNet, the following discussion offers an interesting comparison of multiple pre-trained deep-learning models. MobileNet, ResNet50, EfficientNetV2B3, DenseNet121, DenseNet201, and the unique model created for this study were among the models assessed. It is clear from the data analysis that each model performs differently depending on its architecture and pre-trained weights. ResNet50 exceeds MobileNet with a higher training accuracy of 99.47%, along with a validation accuracy of 99.47% and a slightly lower test accuracy of 99.37%. MobileNet obtains a commendable 98.66% training accuracy. With a training accuracy of 74.16%, a validation accuracy of 81.46%, and a test accuracy of 81.49%, EfficientNetV2B3 displays an intermediate level of performance.

Interestingly, the custom model created for this study performs well, earning accuracy scores of 99.47% during training, 99.44% during validation, and 99.1% during testing. Although DenseNet121 and DenseNet201 showed similar accuracy to our proposed model, the approximate time for running an epoch and completing the entire training is exponentially higher than ours. It's significant to notice that the custom model is trained entirely from scratch and has the fewest parameters (0.1M), indicating efficient learning and optimization. Overall, the data above illustrates the variety of model capabilities as well as the competitive performance of the custom model with only a few parameters. With 4.3 million parameters, MobileNet effectively uses time and space resources. It maintains a minimal parameter count while achieving competitive accuracy, indicating a good compromise between complexity and performance. Contrarily, ResNet50 is accurate with its 25.6 million parameters, but its greater parameter count suggests a higher temporal complexity during training and inference. This means a higher need for computational resources, which might be considered when resources are limited. With its 14.5 million parameters, EfficientNetV2B3 efficiently balances complexity and accuracy. Its results imply that parameters were used well, producing competitive precision without burdening the computer.

Surprisingly, the proposed model matches the accuracy of more complex models while having only 0.1 million parameters. This implies effective resource management, making it a desirable choice for applications with computational limits. It also boasts a validation accuracy of 99.44 percent, which strengthens its generalizability. Even though the test accuracy of 99.1% is slightly lower than the scores for training and validation, it still shows strong predictive ability.

#### 4-3- Training Time Evaluation

Based on our experiment, our model clearly shows that the training is much faster than the existing state-of-the-art model. Compared with several existing DenseNet models, based on the above figure, our model takes 34.4s per epoch of training and 125.5ms per step. On the other hand, for each epoch, denseNet121, denseNet169, and denseNet201 are taking 115s, 144.9s, and 188.5s, respectively. Also, the time per step iteration is exponentially higher than our proposed model. Figure 8 explains the efficiency of our model in terms of time spent training it.

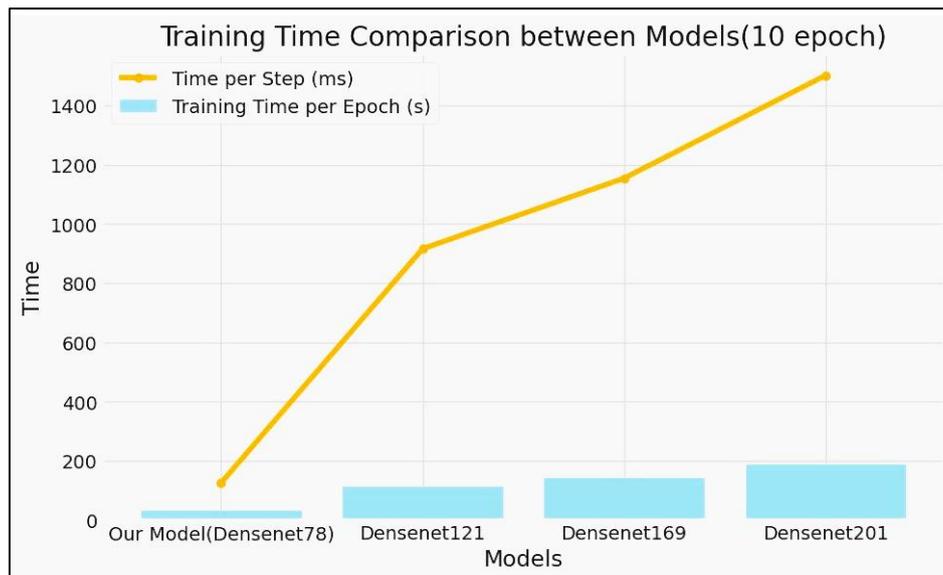


Figure 8. Comparison of our model with pre trained dense net models in terms of time taken for training

## 5- Conclusion

The methodology that is being described introduces a novel framework for the automatic classification and categorization of mango leaf disease. In this work, we created a Custom DenseNet that, compared to the most recent classification model, performed equally well, if not better, with much fewer layers and parameters. For the purpose of removing the representative group of features from the input sample, we especially introduce the DenseNet-78. Using the computed vital points, the classifier is trained to identify and group eight types of mango leaf diseases into distinct categories. The suggested approach can rapidly recognize and classify the eight distinct mango leaf diseases in the dataset. Furthermore, despite the presence of different aberrations, such as variations in the light's intensity, color, size, orientation, and leaf morphology, our method is still accurate in classifying leaf diseases. Experimental results demonstrate that the suggested model outperforms the most recent methods for categorizing leaf diseases already in use. Even if we have obtained comparable results and the suggested model might not be the ideal choice for datasets from other domains, we still expect to offer some simple models for leaf disease recognition. A better feature extractor might be employed to reduce time complexity even more. We will evaluate the custom model using images from real situations in our next study and adapt this work to other plant diseases. We also want to consider additional state-of-the-art DL techniques and test the recognition capability of our approach on increasingly complex datasets.

## 6- Declarations

### 6-1-Author Contributions

Conceptualization, B.U.M.; methodology, B.U.M.; software, B.U.M.; validation, B.U.M. and A.A.M.; formal analysis, B.U.M.; investigation, B.U.M. and G.Y.H.; resources, B.U.M.; data curation, B.U.M.; writing—original draft preparation, B.U.M.; writing—review and editing, B.U.M., G.Y.H., A.A.M., M.J.H., and B.J.; visualization, B.U.M. and B.J.; supervision, G.Y.H., A.A.M., and M.J.H.; project administration, M.J.H. and G.Y.H.; funding acquisition, M.J.H. and A.A.M. All authors have read and agreed to the published version of the manuscript.

### 6-2-Data Availability Statement

The data presented in this study are openly available in a comprehensive image dataset to classify diseased and healthy mango leaves. Data Brief. 2023 Jan 30; 47:108941. doi:10.1016/j.dib.2023.108941. PMID: 36819904; PMCID: PMC9932726.

### 6-3-Funding

The present research was done under the project number MMU/RMC/PC/2024/131001.

### 6-4-Institutional Review Board Statement

Not applicable.

### 6-5-Informed Consent Statement

Not applicable.

### 6-6-Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

## 7- References

- [1] Mahmud, B. U., Hong, G. Y., Mamun, A. Al, Ping, E. P., & Wu, Q. (2023). Deep Learning-Based Segmentation of 3D Volumetric Image and Microstructural Analysis. *Sensors*, 23(5), 2640. doi:10.3390/s23052640.
- [2] Jahan, B., Mahmud, B. U., Mamun, A. A., Mujibur Rahman Majumder, Md., & Alam, M. (2021). Impact Analysis of Harassment Against Women in Bangladesh Using Machine Learning Approaches. *Recent Trends in Mechatronics Towards Industry 4.0*, 549–559. doi:10.1007/978-981-33-4597-3\_50.
- [3] Le, V. N. T., Ahderom, S., Apopei, B., & Alameh, K. (2020). A novel method for detecting morphologically similar crops and weeds based on the combination of contour masks and filtered Local Binary Pattern operators. *GigaScience*, 9(3), 17. doi:10.1093/gigascience/giaa017.
- [4] Ramesh, S., Hebbar, R., Niveditha, M., Pooja, R., Shashank, N., & Vinod, P. V. (2018). Plant disease detection using machine learning. In *2018 International conference on design innovations for 3Cs compute communicate control (ICDI3C)*, 41–45. doi:10.1109/ICDI3C.2018.00017.
- [5] Kuricheti, G., & Supriya, P. (2019). Computer vision based turmeric leaf disease detection and classification: A step to smart agriculture. *Proceedings of the International Conference on Trends in Electronics and Informatics, ICOEI 2019*, 545–549. doi:10.1109/ICOEI.2019.8862706.
- [6] Saleem, R., Shah, J. H., Sharif, M., Yasmin, M., Yong, H. S., & Cha, J. (2021). Mango leaf disease recognition and classification using novel segmentation and vein pattern technique. *Applied Sciences (Switzerland)*, 11(24), 11901. doi:10.3390/app112411901.
- [7] Srunitha, K., & Bharathi, D. (2018). Mango leaf unhealthy region detection and classification. In D. Hemanth & S. Smys (Eds.): *Lecture Notes in Computational Vision and Biomechanics*, 28, 422–436. doi:10.1007/978-3-319-71767-8\_35.
- [8] Mia, M. R., Roy, S., Das, S. K., & Rahman, M. A. (2020). Mango leaf disease recognition using neural network and support vector machine. *Iran Journal of Computer Science*, 3(3), 185–193. doi:10.1007/s42044-020-00057-z.
- [9] Ullagaddi, S. B., & Raju, S. V. (2017). Disease recognition in Mango crop using modified rotational kernel transform features. *4th International Conference on Advanced Computing and Communication Systems, ICACCS 2017*, Coimbatore, India. doi:10.1109/ICACCS.2017.8014610.

- [10] Ullagaddi, S. B., & Viswanadha Raju, S. (2017). An Enhanced Feature Extraction Technique for Diagnosis of Pathological Problems in Mango Crop. *International Journal of Image, Graphics and Signal Processing*, 9(9), 28–39. doi:10.5815/ijigsp.2017.09.04.
- [11] Abdollahi, A., & Pradhan, B. (2021). Urban vegetation mapping from aerial imagery using explainable AI (XAI). *Sensors*, 21(14), 4738. doi:10.3390/s21144738.
- [12] Fuentes, A., Yoon, S., Kim, S. C., & Park, D. S. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors (Switzerland)*, 17(9), 2022. doi:10.3390/s17092022.
- [13] 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. doi:10.1016/j.procs.2020.03.225.
- [14] Richey, B., Majumder, S., Shirvaikar, M. V., & Kehtarnavaz, N. (2020). Real-time detection of maize crop disease via a deep learning-based smartphone app. *Real-Time Image Processing and Deep Learning 2020. International Society for Optics and Photonics*, 10. doi:10.1117/12.2557317.
- [15] Batool, A., Hyder, S. B., Rahim, A., Waheed, N., Asghar, M. A., & Fawad. (2020). Classification and Identification of Tomato Leaf Disease Using Deep Neural Network. *2020 International Conference on Engineering and Emerging Technologies, ICEET 2020*. doi:10.1109/ICEET48479.2020.9048207.
- [16] Tm, P., Pranathi, A., SaiAshritha, K., Chittaragi, N. B., & Koolagudi, S. G. (2018, August). Tomato leaf disease detection using convolutional neural networks. In *2018 Eleventh International Conference on Contemporary Computing (IC3)*, 1-5. doi:10.1109/IC3.2018.8530532.
- [17] Turkoglu, M., Yanikoğlu, B., & Hanbay, D. (2022). PlantDiseaseNet: convolutional neural network ensemble for plant disease and pest detection. *Signal, Image and Video Processing*, 16(2), 301-309. doi:10.1007/s11760-021-01909-2.
- [18] Gulavnai, S., & Patil, R. (2019). Deep Learning for Image Based Mango Leaf Disease Detection. *International Journal of Recent Technology and Engineering*, 8(3S3), 54–56. doi:10.35940/ijrte.c1030.1183s319.
- [19] Pham, T. N., Tran, L. Van, & Dao, S. V. T. (2020). Early Disease Classification of Mango Leaves Using Feed-Forward Neural Network and Hybrid Metaheuristic Feature Selection. *IEEE Access*, 8, 189960–189973. doi:10.1109/ACCESS.2020.3031914.
- [20] Singh, U. P., Chouhan, S. S., Jain, S., & Jain, S. (2019). Multilayer Convolution Neural Network for the Classification of Mango Leaves Infected by Anthracnose Disease. *IEEE Access*, 7, 43721–43729. doi:10.1109/ACCESS.2019.2907383.
- [21] Prabu, M., & Chelliah, B. J. (2022). Mango leaf disease identification and classification using a CNN architecture optimized by crossover-based levy flight distribution algorithm. *Neural Computing and Applications*, 34(9), 7311–7324. doi:10.1007/s00521-021-06726-9.
- [22] Wongsila, S., Chantrasri, P., & Sureephong, P. (2021). Machine Learning Algorithm Development for detection of Mango infected by Anthracnose Disease. *2021 Joint 6th International Conference on Digital Arts, Media and Technology with 4th ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering, ECTI DAMT and NCON 2021*, 249–252. doi:10.1109/ECTIDAMTNCN51128.2021.9425737.
- [23] Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, United States*. doi:10.1109/CVPR.2017.243.
- [24] Ahmed, S. I., Ibrahim, M., Nadim, M., Rahman, M. M., Shejunti, M. M., Jabid, T., & Ali, M. S. (2023). MangoLeafBD: A comprehensive image dataset to classify diseased and healthy mango leaves. *Data in Brief*, 47, 108941. doi:10.1016/j.dib.2023.108941.