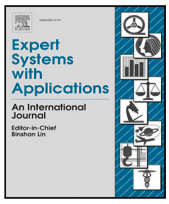
Expert Systems With Applications 267 (2025) 126066

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



Enhancing crop productivity with fined-tuned deep convolution neural network for Potato leaf disease detection

Prit Mhala *, Anushka Bilandani, Sanjeev Sharma

Department of Computer Science and Engineering, Indian Institute of Information Technology Pune, Pune, India

ARTICLE INFO

Keywords:
Potato leaf detection
Deep learning
Transfer learning architectures
Computer vision
Image classification
Agricultural productivity
Plant diseases
Precision agriculture

ABSTRACT

Potato plants (Solanum tuberosum) are prone to various diseases that result in substantial economic losses for farmers. This research presents a deep learning-based approach to accurately detect and classify six distinct diseases affecting potato leaves: bacteria, viruses, fungi, phytophthora, pests, and nematodes. Addressing the challenges of class imbalance, we employed strategic data augmentation, L2 regularization, and transfer learning to enhance model performance. Three pre-trained convolutional neural networks, DenseNet201, ResNet152V2, and NasNetMobile were fine-tuned on a diverse dataset of 3076 images collected under real-world conditions. DenseNet201 achieved the highest accuracy of 77.14% on the original dataset and demonstrated further improvements with data augmentation, reaching an average accuracy of 81.31% through k-fold cross-validation, marking a 4.17% improvement over initial results and a 7.68% increase compared to previously reported findings. While DenseNet201 maintained stability across augmented and regularized settings, NasNetMobile and ResNet152V2 experienced performance declines due to overfitting and limited capacity to handle increased data variability. Our findings highlight the significance of mitigating class imbalance through tailored augmentation and regularization techniques, contributing to more reliable disease detection systems. This approach offers a scalable solution for real-world agricultural challenges, aiding farmers in reducing crop losses and promoting sustainable farming practices.

1. Introduction

Potatoes (Solanum tuberosum) are one of the most widely consumed crops globally, ranking just after wheat, rice, and maize in terms of dietary importance (Reddy, Mandal, Chakroborty, Hijam, and Dutta, 2018). Their versatility in culinary applications and their role as staple foods make them a critical component of the global food supply. India is a significant contributor to global potato production, although China remains the largest producer worldwide. The high yield potential and short growth cycle of potatoes make them an essential crop in addressing food security challenges (Spooner, 2013). However, the widespread cultivation of potatoes also makes them vulnerable to various diseases, which can lead to substantial economic losses and threaten food security (Sharma, Azeem, and Sharma, 2022). Potatoes are a rich source of vitamins, minerals, and phytochemicals, contributing to improved digestion, immune function, and heart health (Tian, Chen, Ye, and Chen, 2016). Their high starch content provides significant energy, while resistant starch acts as a prebiotic, promoting gut health (Andre et al., 2014). Despite these benefits, the susceptibility of potatoes to diseases poses a significant challenge. The impact of these diseases is not only felt in terms of yield reduction but also in the increased costs of disease management and control measures (Visvanathan, Jayathilake, Chaminda Jayawardana, and Liyanage, 2016). Recent technological advancements have significantly influenced potato cultivation. Innovations such as computer vision, robotics, and satellite imaging have been employed to detect viral diseases and monitor crop conditions (Lefebvre et al., Yang, Everitt, Du, Luo, and Chanussot, 1993, 2012). Automated systems for planting and harvesting have improved the efficiency of potato cultivation, while genetic engineering has produced disease-resistant potato varieties (Almanzor, Birell, and Iida, Seitov, Saitov, and Rakintsev, 2023, 2022). However, despite these advances, diseases continue to threaten potato crops, necessitating ongoing research into effective detection and management strategies (Binnar and Sharma, 2023).

Several studies have explored the application of deep learning and transfer learning techniques for detecting potato leaf diseases. For instance, Lanjewar, Morajkar, and P, (2024) conducted research using modified transfer learning frameworks to identify potato leaf diseases such as early blight and late blight. The study employed pre-trained models like VGG19, NASNetMobile, and DenseNet169, modifying them

* Corresponding author.
E-mail addresses: 112115089@cse.iiitp.ac.in (P. Mhala), 112215027@cse.iiitp.ac.in (A. Bilandani), sanjeevsharma@iiitp.ac.in (S. Sharma).

https://doi.org/10.1016/j.eswa.2024.126066
Received 30 March 2024; Received in revised form 13 November 2024; Accepted 3 December 2024
Available online 16 December 2024

by introducing additional layers to reduce trainable parameters and improve performance. The modified DenseNet achieved a remarkable accuracy of 99% on the test set and 100% on the validation set. While this approach demonstrated high accuracy, it primarily focused on specific diseases and relied on a dataset from Kaggle, which may limit its generalizability to real-world agricultural settings. Jha, Dembla, and Dubey, (2024) proposed a deep learning ensemble model combining Residual Network, MobileNet, and Inception models to enhance potato leaf disease prediction. This model achieved an overall accuracy of 98.86%, showcasing its effectiveness in classifying potato leaf diseases. However, the approach, while robust, concentrated on a limited number of disease classes, and the reliance on extensive datasets for training raises concerns about scalability in diverse environments. Paul et al., (2024) focused on the application of AI techniques to detect and classify four types of potato leaf diseases: early blight, septoria disease, late blight, and black-leg disease. Their deep ensemble algorithm, integrating CNN, CNN-SVM, and DNN, achieved an impressive accuracy of 99.98%. Despite its high accuracy, the study was constrained by the limited scope of diseases it addressed and the complexity of the ensemble models, which may hinder deployment in resource-limited settings. Reis and Turk, (2024) developed a novel deep learning model based on depthwise separable convolution and transformer networks for classifying potato leaf diseases. The proposed MDSCIRNet architecture, combined with techniques like CLAHE and ESRGAN for image enhancement, achieved an accuracy of 99.33% when integrated with SVM. While the model's performance is commendable, the integration of multiple complex techniques may present challenges in terms of computational efficiency and real-time application. Ashikuzzaman, Roy, Lamon, and Abedin, (2024) conducted a comparative study using nine transfer-learning deep CNN models to detect potato leaf diseases. DenseNet201 achieved the highest validation accuracy of 96% with low losses, demonstrating the effectiveness of transfer learning in disease detection. However, the study primarily focused on a narrow set of diseases and was trained on images from a specific region, limiting its generalizability.

While these studies have made significant contributions to the detection and classification of potato leaf diseases, they predominantly focus on specific diseases such as early blight, late blight, bacterial wilt, leaf spot, and powdery mildew, often relying on high-quality, controlled datasets. These approaches, while accurate, may not generalize well to real-world conditions where image quality, environmental factors, and disease manifestations can vary significantly. Moreover, the scalability of these models to diverse agricultural environments remains largely unexplored, particularly in the context of resource-limited settings where computational efficiency is crucial. Addressing these limitations is essential for developing more robust and widely applicable disease detection models.

In addition, a major challenge in agricultural datasets is the issue of class imbalance, where certain disease categories are underrepresented due to the natural variability in disease occurrences. Class imbalance can severely impact the performance of machine learning models, especially in terms of generalization and accuracy. To address this, our research proposes a set of strategic augmentations designed to mimic real-world image capturing scenarios, helping the model to learn fine details and improve generalization across classes. These augmentations include adjustments for brightness, contrast, rotation, and noise, which introduce variability and allow the model to better handle underrepresented classes. Furthermore, we apply early stopping and L2 regularization techniques to reduce overfitting, which is a common issue in highly imbalanced datasets.

This research aims to bridge these gaps by developing a fine-tuned deep convolutional neural network (CNN) model specifically designed to detect a broader spectrum of potato leaf diseases, including those caused by Bacteria, Fungi, Nematodes, Pests, Phytophthora, and Viruses. By leveraging advanced deep learning techniques and integrating innovative model architectures, this study seeks to enhance model accuracy, generalizability, and robustness across diverse agricultural environments, especially when dealing with class imbalance.

Early detection facilitated by this model will enable timely and targeted interventions, reducing the spread of diseases and minimizing crop losses (Tewari, Azeem, and Sharma, 2023).

The key contributions of this paper are:

1. The development of deep learning methods for detecting a wide range of diseases in potato leaves, with a focus on improving accuracy, scalability, and handling class imbalance.
2. Implementation of strategic augmentations to mimic real-world image-capturing scenarios, which improves model generalization, especially for underrepresented classes.
3. Comprehensive evaluation metrics, including Accuracy, Precision, Recall, and F1-Score, to ensure robust model performance.
4. Optimization of hyperparameters to achieve superior results in diverse agricultural settings.
5. A comparative analysis of CNN architectures NASNetMobile, DenseNet201, and ResNet152v2 with 3 different experimentation settings to identify the most effective model for potato leaf disease detection.
6. Application of early stopping and L2 regularization techniques to mitigate overfitting in highly imbalanced datasets.
7. Addressing the limitations of previous studies, which primarily focused on specific diseases like early blight, late blight, bacterial wilt, leaf spot, and powdery mildew, by expanding the disease spectrum to include Bacteria, Fungi, Nematodes, Pests, Phytophthora, and Viruses, and enhancing model generalizability across varied real-world conditions.

The structure of this study is summarized as follows: Section 2 presents a Literature Review. Section 3 describes the Materials and Methods used. Section 4 provides the experimental results and inferences obtained. Finally, Section 5 discusses the future scope of this study (see Table 1).

2. Literature study

The application of deep learning and machine learning techniques in detecting potato leaf diseases has made significant progress in recent years. However, most studies have primarily focused on a narrow set of diseases, such as early blight, late blight, bacterial wilt, leaf spot, and powdery mildew, often using controlled datasets that limit their scalability in real-world scenarios.

For instance, Sholihati et al., (2020) utilized VGG16 and VGG19 convolutional neural network (CNN) architectures to classify specific potato leaf diseases, achieving an average accuracy of 91%. Similarly, Khalifa et al., (2021) employed a 14-layer deep CNN to classify potato leaf blight, achieving a high mean testing accuracy of 98%. Despite the impressive accuracy, both studies heavily rely on high-quality image datasets and lack validation in diverse agricultural environments, questioning their generalizability.

Transfer learning approaches have also been explored. Mahum et al., (2023) and Tiwari et al., (2020) leveraged pre-trained VGG19 and a modified DenseNet-201 model, respectively, to classify diseases. While Tiwari et al., (2020) achieved a 97.8% accuracy with logistic regression, Mahum et al., (2023) reached 97.2% accuracy by adding extra transition layers and a reweighted cross-entropy loss function to minimize overfitting. However, both studies fell short in addressing the challenges of varied field conditions, mainly relying on high-quality, controlled images, which limits their real-world applicability. Similarly, Lanjewar et al., (2024) explored transfer learning using models like VGG19, NASNetMobile, and DenseNet169, achieving a high accuracy of 99% on the test set. Although this work demonstrated reduced trainable parameters, its dependence on the Kaggle

Table 1
Literature review Table - Entries 1 to 10.

Sr. No.	Author and citation	Accuracy	Problem addressed
1	Rizqi Amaliatus Sholihati (Sholihati, Sulistijono, Risnumawan, & Kusumawati, 2020)	91%	Potato leaf classification
2	Nour Eldeen M. Khalifa (Khalifa, Taha, Abou El-Maged, & Hassanien, 2021)	98%	Potato leaf blight classification
3	Divyansh Tiwari (Tiwari et al., 2020)	97.8%	Early and late blight detection
4	Rabbia Mahum (Mahum et al., 2023)	97.2%	Potato leaf disease classification (5 categories)
5	Ungsumalee Suttapakti (Suttapakti & Bunpeng, 2019)	Not specified	Potato leaf disease classification via color and texture feature extraction
6	Alok Kumar (Kumar & Patel, 2023)	Not specified	Potato leaf disease classification using HDLCNN
7	Javed Rashid (Rashid et al., 2021)	99.75%	Multi-level deep learning for potato leaf disease recognition
8	Hritwik Ghosh (Ghosh, Rahat, Shaik, Khasim, & Yesubabu, 2023)	Not specified	Potato leaf disease recognition using CNNs
9	Sakshi Sharma (Sharma, Anand, & Singh, 2021)	92.9%	Early and late blight detection
10	Weirong Chen (Chen, Chen, Zeb, Yang, & Zhang, 2022)	97.73%	Potato leaf disease recognition using MobOca_Net

dataset and potential computational complexity revealed gaps in its generalizability across different agricultural settings.

Advanced methodologies have also been proposed for disease detection. Rashid et al., (2021) used a multi-level deep learning model combining YOLOv5 for image segmentation and a CNN for disease classification, achieving 99.75% accuracy. This approach was effective in controlled settings, but its reliance on specific regional data and lack of validation in varied environmental conditions limited its generalizability. Similarly, Arshad et al., (2023) introduced PLDPNet, a hybrid deep learning model for automatic segmentation and classification, achieving an accuracy of 98.66%. While robust, the model's dependence on high-quality datasets poses challenges for deployment in resource-constrained environments. In another study focusing on the segmentation and texture analysis approach, Suttapakti and Bunpeng, (2019) employed k-means clustering for color and texture feature extraction. Although this method demonstrated simplicity and effectiveness in controlled environments, it struggled with complex images and was limited by a small dataset size. In a similar vein, Kumar and Patel, (2023) proposed a Hierarchical Deep Learning Convolutional Neural Network (HDLCNN) for potato leaf disease classification using statistical texture features, which improved accuracy. However, the model's limitations in handling complex leaf images and the lack of validation in diverse environments restrict its broader applicability.

Lightweight architectures have been explored to facilitate the practical deployment of disease detection models. Chen et al., (2022) proposed MobOca_Net, an architecture using MobileNet V2 with an attention mechanism, achieving an accuracy of 97.73%. While the model is suitable for mobile devices, its reliance on a single dataset and lack of validation across diverse field conditions pose limitations. Similarly, Ghosh et al., (2023) evaluated VGG19, DenseNet121, and ResNet50 models for potato leaf disease recognition, emphasizing the robustness of VGG19 through extensive data augmentation. However, the study highlighted the need for scalability in large-scale agricultural deployment.

Traditional machine-learning techniques have also been employed in disease classification. Sharma et al., (2021) used Gaussian filtering and K-means clustering for region of interest (ROI) extraction and applied Support Vector Machine (SVM) for classification, achieving a 92.9% accuracy. Despite its effectiveness in controlled environments, the model's reliance on image quality and a small dataset limited its generalizability. Additionally, Suttapakti and Bunpeng, (2019) used Euclidean distance classification, which, while straightforward, was restricted by the limited choice of classifier and dataset size, highlighting the need for more advanced techniques.

Deep CNN-based approaches have been prominent in disease detection. Asif, Rahman, and Hena, (2020) and Rozaqi and Sunyoto, (2020) aimed at early identification to prevent economic losses, achieving around 97% accuracy. However, they primarily focused on specific diseases and lacked extensive comparisons between different CNN architectures. In contrast, Singh and Yogi, (2023) explored transfer learning combined with baseline learning methods, achieving accuracies of up to 99.62% using various optimization techniques. Nevertheless, the reliance on specific datasets and the absence of validation in real-world settings highlighted the need for broader applicability.

Agarwal, Sinha, Gupta, Mishra, and Mishra, (2020) presented a deep learning-based approach targeting early and late blight through visual analysis, reaching 99.8% accuracy. While robust, the study's reliance on controlled experimental conditions and high computational demands posed challenges for practical deployment. Chakraborty, Mukherjee, Chakroborty, and Bora, (2022) enhanced VGG16's performance for automated recognition of potato leaf blight, achieving 97.89% accuracy. Despite the model's improved accuracy, its reliance on the PlantVillage dataset and limited exploration of other models' fine-tuning revealed gaps in its applicability across diverse conditions.

Saeed et al., (2021) utilized deep CNNs, including ResNet-152 and InceptionV3, achieving a high accuracy of 98.34%. While the model effectively classified leaves into healthy, early blight, and late blight categories, its dependence on a specific dataset limited its generalizability, and the computational demands were a concern for resource-limited settings. Hasan, Zahan, Zeba, Khatun, and Haque, (2021) also proposed a deep learning-based approach using AlexNet, ResNet, and GoogLeNet architectures, with ResNet yielding the best performance. However, the study emphasized the need for diverse samples, varying environmental conditions, and optimization for practical agricultural applications. Md Ashraful Islam (Islam and Sikder, 2022) presented a deep learning approach to classify potato leaf diseases using a CNN, achieving a remarkable 100% accuracy. The study's advantages include its high accuracy and the use of a sizable dataset. However, the reliance on manually collected data and the lack of generalizability across diverse agricultural conditions are significant disadvantages.

While these studies have made substantial contributions to potato leaf disease detection, they largely focus on specific diseases such as early blight, late blight, bacterial wilt, leaf spot, and powdery mildew. Their dependence on high-quality, controlled datasets further restricts scalability to real-world conditions, where image quality, environmental factors, and disease manifestations vary significantly. Additionally, most existing research does not address diseases caused by a broader range of pathogens, including bacteria, fungi, nematodes, pests, phytophthora, and viruses, nor do they comprehensively explore the issue of generalizability across diverse agricultural settings.

To address these gaps, this research expands the disease spectrum to include a wider variety of pathogens affecting potato leaves, such as bacteria, fungi, nematodes, pests, phytophthora, and viruses. We emphasize enhancing model generalizability across varied real-world conditions to improve the practical applicability of deep learning models in agriculture. Unlike previous studies, which assume balanced datasets, our research acknowledges the inherent imbalance in agricultural data, where images of certain diseases are less common than

Table 2
Literature review Table - Entries 11 to 20.

Sr. No.	Author and Citation	Accuracy	Problem addressed
11	Zubair Saeed (Saeed et al., 2021)	98.34%	Potato leaf disease classification (ResNet-152 and InceptionV3)
12	Md. Zahid Hasan (Hasan et al., 2021)	Not specified	Early potato disease detection using CNNs
13	Abdul Jalil Rozaqi (Rozaqi & Sunyoto, 2020)	97%	Potato leaf disease detection using CNN
14	Md. Khalid Rayhan Asif (Asif et al., 2020)	97%	Potato leaf disease detection using multiple CNN architectures
15	Fizzah Arshad (Arshad et al., 2023)	98.66%	PLDPNet for potato leaf disease segmentation and classification
16	Madhusudan G. Lanjewar (Lanjewar et al., 2024)	99%	Modified transfer learning for potato leaf disease detection
17	Mohit Agarwal (Agarwal et al., 2020)	99.8%	Potato crop disease classification under varied conditions
18	Gulbir Singh (Singh & Yogi, 2023)	99.62%	Potato leaf disease detection using transfer learning
19	Kulendu Kashyap Chakraborty (Chakraborty et al., 2022)	97.89%	Automated recognition of potato leaf blight diseases
20	Md Ashraful Islam (Islam & Sikder, 2022)	100%	Potato leaf disease classification using CNN

others. We apply strategic data augmentations to mitigate this imbalance, ensuring the development of a more robust model that reflects real-world scenarios where data imbalance is a critical challenge (see Table 2).

3. Material and methods

In this research, we employed a comprehensive methodology designed to enhance the classification of potato leaf diseases using deep learning models. The overall workflow integrates key processes such as dataset preprocessing, strategic data augmentation, fine-tuning of pre-trained models, and rigorous evaluation of model performance. To address common challenges like class imbalance and overfitting, we applied a combination of advanced augmentation techniques, early stopping, and L2 regularization, ensuring that the models are not only accurate but also robust and generalizable to real-world scenarios. The complete methodology is visually represented in Fig. 1.

3.1. Dataset description

The dataset utilized in this study is the Potato Leaf Disease Dataset, captured in an uncontrolled environment (Shabrina et al., 2023). This dataset contains a total of 3076 images, organized into seven distinct classes: Bacteria, Fungi, Nematode, Pest, Phytophthora, Virus, and Healthy. Unlike prior datasets, which were typically collected under controlled conditions and often lacked comprehensive coverage of disease types, this dataset provides a more realistic representation of potato leaf diseases by reflecting the natural variability encountered in real-world settings. The data collection took place across several potato farms in Central Java, Indonesia, using multiple smartphone cameras. This method resulted in substantial variability in image backgrounds, lighting conditions, and leaf orientations, making the dataset especially valuable for developing robust models capable of generalizing to diverse environments.

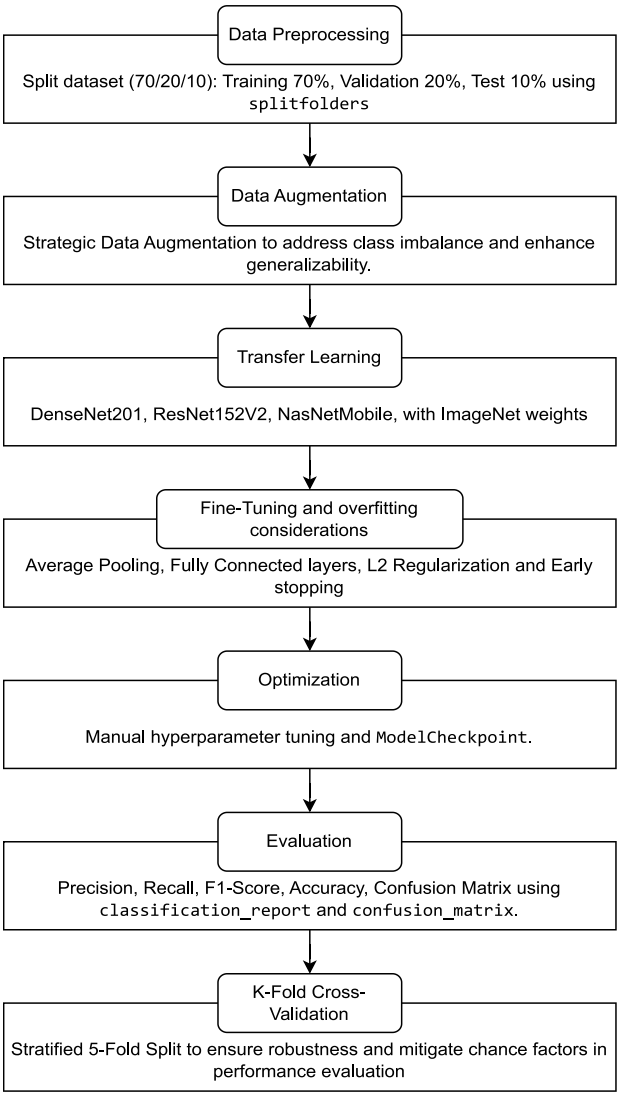


Fig. 1. Methodology flowchart.

The dataset features 201 images of healthy potato leaves, which serve as a baseline for comparing diseased samples. Among the disease categories, the Fungi class is the largest, comprising 748 images, while the Nematode class is the smallest, with only 68 images. The other classes include Bacteria (569 images), Pest (611 images), Virus (532 images), and Phytophthora (347 images). Fig. 2 offers a visual representation of the distribution of images across these classes. Each image in the dataset has a high resolution of 1500 × 1500 pixels, stored in .jpg format, which preserves the fine details crucial for precise disease identification. The combination of varied image capture conditions and comprehensive disease representation makes this dataset a valuable resource for advancing research in potato leaf disease detection. Fig. 3 showcases sample image of each class present in the dataset.

3.2. Dataset preprocessing

In this study dataset preprocessing is a crucial step to prepare the data for model training and subsequent stages. The dataset is split into training, validation, and testing in 70/20/10 ratio. Images are then resized to a fixed dimension to maintain consistency across all samples, a standard practice in deep learning to ensure compatibility with various neural network architectures. Additionally, pixel values are normalized to the [0, 1] range to aid in faster convergence during

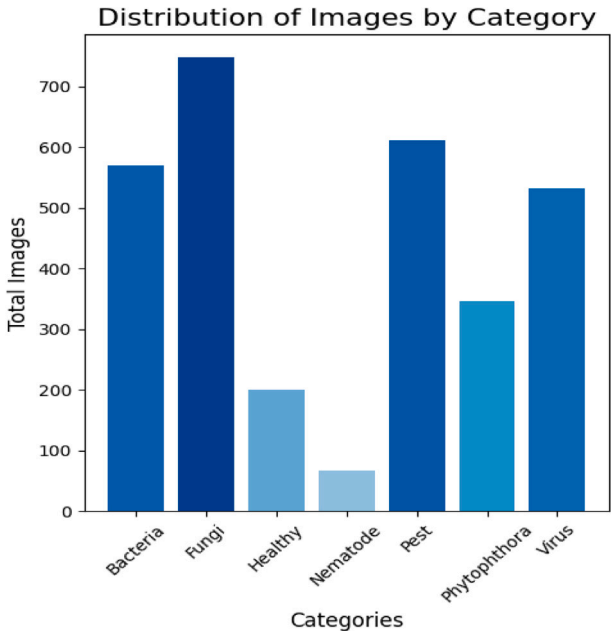


Fig. 2. Dataset distribution among various classes (Original).

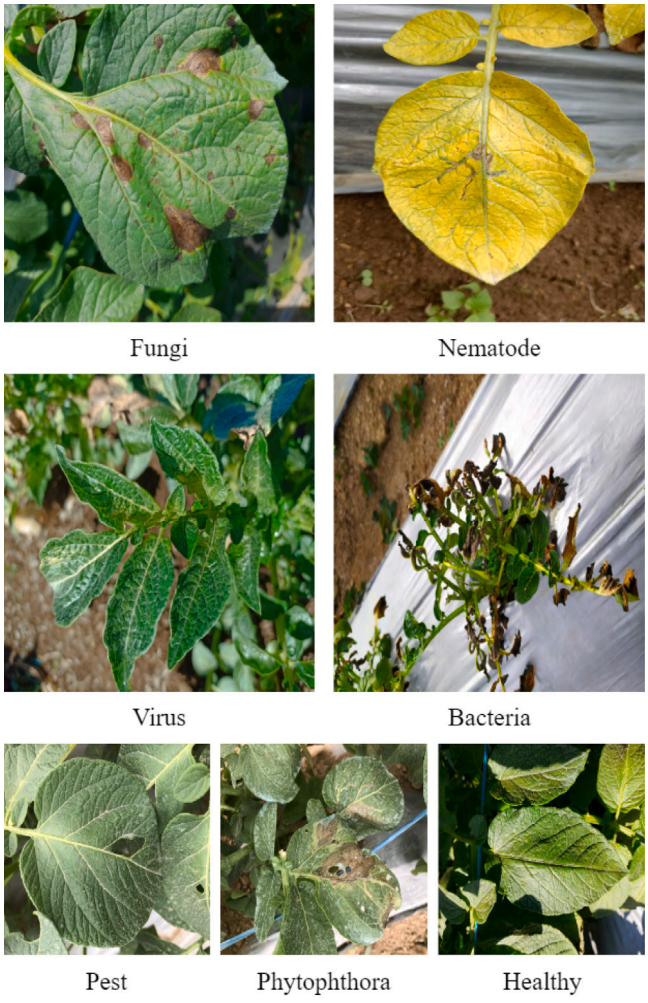


Fig. 3. Sample images illustrating the potato leaf diseases across various classes in the dataset.

Table 3
Image distribution across disease categories, with allocation to training, validation, and test sets before data augmentation.

Category	Original images	Train images	Validation images	Test images
Nematode	68	48	14	6
Bacteria	569	398	114	57
Healthy	201	140	40	21
Phytophthora	347	243	69	35
Fungi	748	523	150	75
Pest	611	428	122	61
Virus	532	372	106	54
Total	**3076**	**2152**	**615**	**309**

training and to prevent large numerical values from affecting the model's learning process. An analysis of class distribution is conducted, revealing significant class imbalances, as shown in the dataset distribution image (Fig. 2). This imbalance introduces several drawbacks, including a model's tendency to bias toward the majority classes, resulting in poor generalization and accuracy when predicting minority classes. Additionally, there is an increased risk of overfitting, as the model may learn to rely heavily on patterns from the over-represented classes, limiting its ability to recognize and correctly classify less common instances. To address this issue, class balancing is incorporated by applying augmentations to both the minority and majority classes, ensuring a more balanced distribution in the final dataset. By carefully increasing the representation of minority classes while also augmenting majority classes, this approach helps mitigate biases and prevents the model from overfitting to any particular class.

3.3. Training, validation and testing data

The Potato Leaf Disease Dataset captured in an uncontrolled environment (Shabrina et al., 2023), was divided into three subsets for this study: 70% for training, 20% for validation, and 10% for testing. This division facilitates efficient model training, enables precise fine-tuning, and ensures a thorough evaluation of the model's performance. Such a split ensures that the model provides accurate and robust results across the dataset while maintaining its ability to generalize effectively to new, unseen data. Table 3 outlines the distribution of images across the various disease classes in the dataset and their respective allocation to the training, validation, and test sets. It should be noted that the data in the table reflects the distribution prior to any data augmentation.

3.4. Data augmentation

The primary goal of data augmentation in this study is to mitigate the risk of overfitting and to create a more diverse and balanced training dataset. By artificially expanding the dataset through various transformations, the model is exposed to a broader range of scenarios, thereby improving its ability to generalize to unseen data. Importantly, the augmentation techniques are applied exclusively to the training data, after the dataset has been divided into training, validation, and testing sets. This ensures that the validation and test sets remain unaltered, serving as unbiased and reliable benchmarks for evaluating the model's performance.

By focusing the augmentation on the training set, the model is able to learn more robust and diverse features without compromising the integrity of the validation and test data. This approach helps the model avoid overfitting on the limited training data while preserving the generalizability of its performance metrics. The specific augmentation techniques used for different classes are summarized in Table 4, while Table 5 outlines the specific values and ranges applied for each augmentation. These augmentation techniques and their combinations were carefully selected to simulate real-world conditions that an embedded system, such as a robot or drone, may encounter during the deployment of the trained models. By incorporating these realistic

Table 4
Augmentation techniques applied to different classes.

Class	Count	Augmentations
Nematode	36	rotation_range, horizontal_flip, brightness, color_jitter, contrast, zoom_range, translate, small_noise, rotation_range + horizontal_flip, rotation_range + brightness, horizontal_flip + brightness, color_jitter + contrast, translate + brightness, zoom_range + color_jitter, zoom_range + contrast, horizontal_flip + small_noise, rotation_range + small_noise, translate + color_jitter, brightness + small_noise, horizontal_flip + zoom_range, brightness + contrast, horizontal_flip + color_jitter, translate + zoom_range, zoom_range + small_noise, horizontal_flip + translate, brightness + horizontal_flip, contrast + small_noise, color_jitter + small_noise, horizontal_flip + contrast, translate + small_noise, rotation_range + zoom_range, color_jitter + translate, rotation_range + contrast, vertical_flip + brightness, contrast + zoom_range, rotation_range + color_jitter
Healthy	13	rotation_range, zoom_range, horizontal_flip, brightness, color_jitter, contrast, small_noise, rotation_range + brightness, horizontal_flip + color_jitter, zoom_range + contrast, brightness + small_noise, horizontal_flip + small_noise, color_jitter + contrast
Phytophthora	8	rotation_range, horizontal_flip, brightness, color_jitter, contrast, zoom_range, translate, vertical_flip
Bacteria	4	rotation_range, horizontal_flip, translate, small_noise
Fungi	3	rotation_range, zoom_range, color_jitter
Pest	4	small_noise, rotation_range, horizontal_flip, color_jitter
Virus	4	color_jitter, horizontal_flip, rotation_range, translate

Table 5
Augmentation techniques with their respective values or ranges.

Augmentation type	Values/Range
Brightness	Random factor in the range [0.9, 1.1]
Color Jitter	Random factor in the range [0.8, 1.2]
Contrast	Random factor in the range [0.9, 1.1]
Small noise	Gaussian noise with mean 0 and standard deviation 0.02×255
Rotation range	Random rotation in the range [−10, 10] degrees
Zoom range	Random zoom factor in the range [0.85, 1.15]
Horizontal Flip	Mirroring of the image along the vertical axis
Vertical flip	Flipping of the image along the horizontal axis
Translate	Random horizontal and vertical shifts up to 10% of image width and height
Combined Augmentations	Combination of two augmentations (e.g., "rotation + brightness")

transformations, the model is better equipped to handle the variability and challenges it will face in practical applications, enhancing its robustness and adaptability.

In this work, we explore a wide range of augmentation techniques applied to leaf images to enhance the generalizability of our deep learning model. These augmentations can be categorized into various transformation types, each serving a specific purpose in simulating real-world conditions. Table 7 details basic transformations such as rotation, flipping, and translation, which are essential for mimicking changes in leaf orientation and positioning. These augmentations help the model generalize to different leaf appearances in the field. In addition to position changes, brightness adjustments play a crucial role in handling varying lighting conditions. As shown in Table 8, we use several brightness-based augmentations to simulate different light intensities, reflecting outdoor variations that occur naturally. Similarly, Table 9 presents color and contrast modifications that help the model handle shifts in color and contrast due to lighting changes and shadows.

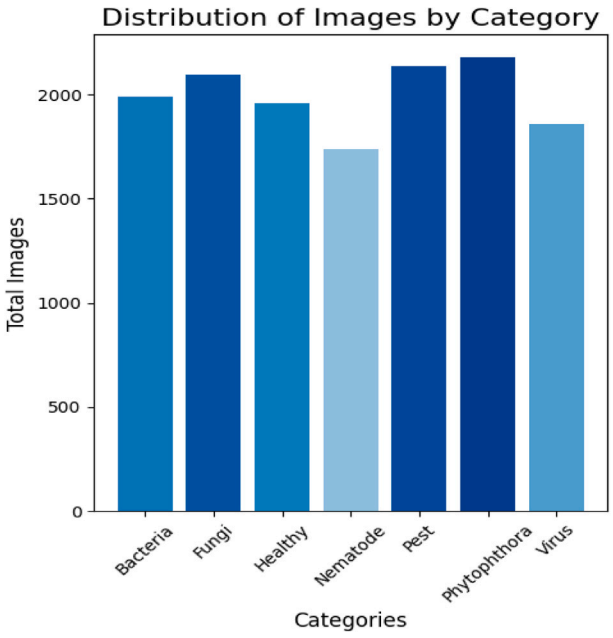


Fig. 4. Dataset distribution among various classes after augmentation.

Zoom modifications, discussed in Table 10, replicate real-world variations in leaf size and camera distance, further enhancing the dataset's diversity. Noise additions, highlighted in Table 11, simulate sensor noise or image artifacts commonly found in agricultural images, making the model robust to noisy inputs. Combined position and brightness modifications, shown in Table 12, represent scenarios where leaves are partially shifted and exposed to different lighting conditions, helping the model learn to detect leaves in uncontrolled environments. Lastly, Table 13 illustrates how combining rotation with lighting adjustments improves the model's ability to generalize under different angles and lighting conditions. Together, these augmentations provide a comprehensive strategy for enhancing the model's robustness and accuracy in detecting and classifying plant diseases.

Table 6 showcases the resulting number of augmented images obtained for the training data after the augmentation process was completed. Fig. 4 provides a visual representation of the class count distribution through a bar graph. This figure helps to illustrate the balance of data across different classes after augmentation, ensuring that no class is underrepresented, which is vital for preventing class imbalance issues during training.

The Nematode class images reveal distinctive patterns, including yellowing, patchy discoloration along the leaf veins, and irregular textures, often localized to specific areas of the leaf as showcased in Fig. 9. These visual cues make nematode infections unique, requiring carefully selected augmentations to capture and enhance these characteristics. To address the diverse visual features of nematode-infected leaves, a broad range of transformations was applied, such as rotation, horizontal and vertical flips, brightness adjustments, color jitter, contrast changes, zoom, translation, and small noise. Rotation and horizontal flips were used to account for different angles at which leaves might be viewed, ensuring that vein patterns and localized discolorations remain identifiable regardless of orientation. Brightness adjustments and color jitter simulated variations in lighting conditions, affecting

Table 6
Comparison of original and augmented image counts in training data.

Category (Disease)	Original images	Total images after augmentation (an training data)
Nematode	68	1739
Bacteria	569	1990
Healthy	201	1960
Phytoph-thora	347	2178
Fungi	748	2092
Pest	611	2135
Virus	532	1860
Total	**3076**	**13 954**

Table 7
Basic transformations and their purposes.

Basic transformations (Orientation and Position Changes)	Purpose
rotation_range	Simulates different angles at which an image might be captured to mimic real-world variations.
horizontal_flip	Accounts for different orientations of leaves in the field by mirroring the image.
vertical_flip	Similar to horizontal flip, it helps in simulating different leaf orientations by flipping the image vertically.
translate	Introduces shifts in image positioning to simulate minor camera movements or off-centered captures.
translate + zoom_range	Accounts for both translation and zoom, emulating a scenario where the leaf is partially shifted and either zoomed in or out.
horizontal_flip + translate	Mirrors the image and shifts it, resembling leaves in various mirrored positions within the camera's frame.
horizontal_flip + rotation_range	Simulates a leaf appearing at an angle with a mirrored orientation, as it might be photographed from different positions.

Table 8
Brightness adjustments and their purposes.

Brightness adjustments	Purpose
brightness	Addresses varying lighting conditions that occur in outdoor settings by adjusting the brightness of the image.
brightness + contrast	Reflects images captured under varying light intensities and contrast settings, providing a broader learning context for the model.
brightness + horizontal_flip	Adjusts brightness and flips the image, representing different light reflections on mirrored leaves.
brightness + rotation_range	Reflects conditions where a leaf is captured at an angle under varying sunlight, enhancing the model's ability to detect features in both low-light and bright environments.
brightness + small_noise	Adjusts brightness while introducing mild noise, representing varying light conditions in conjunction with sensor noise.
brightness + translate	Imitates images where the subject is slightly shifted within the frame and exposed to different brightness levels, as might happen in uncontrolled environments.
brightness + vertical_flip	Flips the image vertically while adjusting brightness, representing leaves in varied lighting from different perspectives.

the appearance of yellowing and dark spots while still preserving the infection's essential visual markers. Zoom and translation focused on specific sections of the leaf, emphasizing the patchy patterns central to nematode identification. Small noise, when combined with other transformations like contrast or brightness, reflected the natural variability in leaf texture that often accompanies infection. Additionally, combining multiple augmentations, such as rotation with brightness or color jitter with zoom, further enriched the dataset by capturing subtle

Table 9
Color and contrast adjustments and their purposes.

Color and contrast adjustments	Purpose
color_jitter	Simulates changes in color due to lighting variations, which helps in generalizing the model to different environmental conditions.
contrast	Imitates changes in image quality and lighting conditions by adjusting the contrast.
color_jitter + contrast	Represents changes in both color and contrast, simulating conditions where crops appear different due to lighting, shadows, or camera settings.
color_jitter + translate	Simulates a shifted subject with changes in color, addressing scenarios where the leaf is partially out of frame under different lighting.
contrast + horizontal_flip	Flips the image and adjusts contrast, reflecting mirrored images with varying light intensities.
contrast + zoom_range	Varies contrast in conjunction with zoom, replicating different distances and lighting conditions.
color_jitter + zoom_range	Replicates scenarios where the camera is zoomed in or out while the leaf color varies due to lighting or shadow effects.
color_jitter + horizontal_flip	Represents flipped images with altered color, simulating leaves under diverse lighting or reflection conditions.

Table 10
Zoom modifications and their purposes.

Zoom modifications	Purpose
zoom_range	Replicates the effect of the camera being closer to or further away from the subject, adding variability to the dataset.
horizontal_flip + zoom_range	Mirrors the image and changes its zoom level, mimicking real-life variations in crop size and positioning within the frame.
rotation_range + zoom_range	Rotates and changes zoom, addressing cases where leaves are captured from different angles and distances.
small_noise + zoom_range	Simulates zoom effects in conjunction with sensor noise, replicating realistic capture conditions.
brightness + zoom_range	Varies brightness with zoom, helping the model generalize across different illumination levels and leaf sizes.
color_jitter + zoom_range	Replicates scenarios where the camera is zoomed in or out while the leaf color varies due to lighting or shadow effects.

variations that enhance the model's ability to recognize nematode infections under different real-world conditions. This comprehensive set of augmentations was chosen to encapsulate the complex visual patterns characteristic of nematode damage, strengthening the model's ability to detect these unique indicators across diverse scenarios. These augmentations also contributed to class balancing, ensuring that the Nematode class had sufficient variety and representation in the dataset.

For the Healthy class, the selected augmentations — rotation range, zoom range, horizontal flip, brightness adjustment, color jitter, contrast adjustment, and small noise — were chosen to enhance the diversity of the dataset while preserving the key characteristics of healthy leaves, such as their uniform green color, smooth texture, and absence of visible disease markers as showcased in Fig. 11. Rotation and horizontal flip simulate different viewing angles, reflecting how healthy leaves may appear in varied orientations in natural settings. Adjustments to brightness, color jitter, and contrast were used to mimic different lighting conditions and slight natural variations in leaf color, capturing the potential range of greens seen in healthy foliage without altering the essential healthy appearance. The addition of small noise adds a subtle level of texture variation, helping the model learn to distinguish healthy leaves even when minor visual noise is present. These augmentations were intended to strengthen the model's recognition of healthy leaves across a range of realistic conditions, while also contributing to overall class balance by increasing the representation of this class.

Table 11
Noise additions and their purposes.

Noise additions	Purpose
small_noise	Represents sensor noise or artifacts commonly found in real-life agricultural images, improving the model's robustness to image noise.
horizontal_flip + small_noise	Mirrors the image and adds mild noise, simulating natural image variations caused by sensor artifacts or environmental factors.
color_jitter + small_noise	Varies color while adding noise, mimicking changes due to lighting fluctuations and sensor imperfections.
brightness + small_noise	Adjusts brightness while introducing mild noise, representing varying light conditions in conjunction with sensor noise.
small_noise + translate	Shifts the subject while adding noise, simulating leaves that are partially out of frame with slight image artifacts.
contrast + small_noise	Alters contrast while introducing noise, capturing realistic image conditions with varied quality.

Table 12
Combined position and brightness modifications and their purposes.

Combined position and brightness modifications	Purpose
translate + brightness	Imitates images where the subject is slightly shifted within the frame and exposed to different brightness levels, as might happen in uncontrolled environments.
translate + zoom_range	Accounts for both translation and zoom, emulating a scenario where the leaf is partially shifted and either zoomed in or out.
rotation_range + brightness	Reflects conditions where a leaf is captured at an angle under varying sunlight, enhancing the model's ability to detect features in both low-light and bright environments.

Table 13
Rotation and lighting combinations and their purposes.

Rotation and lighting combinations	Purpose
rotation_range + contrast	Adjusts rotation and contrast, enhancing the model's robustness against angle and light intensity changes.
rotation_range + color_jitter	Combines rotation with color changes, representing leaves captured at different angles under varying lighting conditions.
contrast + rotation_range	Adjusts rotation and contrast, enhancing the model's robustness against angle and light intensity changes.
color_jitter + rotation_range	Combines rotation with color changes, representing leaves captured at different angles under varying lighting conditions.

For the Phytophthora class, the selected augmentations — rotation range, horizontal flip, brightness adjustment, color jitter, contrast adjustment, zoom range, translation, and vertical flip — were applied to capture the diverse and distinctive features of Phytophthora infection on leaves. Phytophthora symptoms often include irregular dark spots, patches of brown or yellowing edges, and decaying tissue that can vary significantly in appearance based on factors like leaf angle and light exposure as showcased in Fig. 8. Rotation and flipping (both horizontal and vertical) were chosen to represent different orientations and viewpoints, ensuring that the model can recognize these infection patterns regardless of leaf positioning. Brightness, color jitter, and contrast adjustments simulate varying lighting conditions, capturing subtle variations in color and texture often seen in diseased areas. The zoom range and translation augmentations were applied to focus on specific parts of the infected region, ensuring the model can identify Phytophthora even in close-up views or slightly shifted perspectives. This combination of augmentations aimed to enhance the dataset's diversity, allowing the model to learn and generalize Phytophthora features under different environmental and positional contexts, while also increasing class representation.

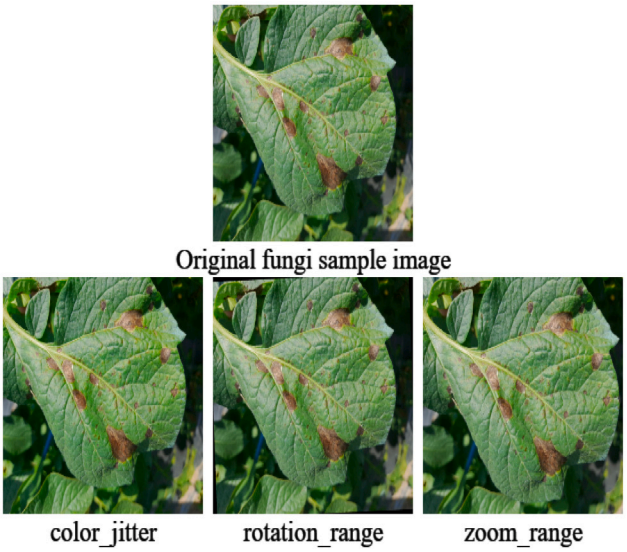


Fig. 5. Distribution of fungi class images after augmentation.

For the Bacteria class, the selected augmentations — rotation range, horizontal flip, translation, and small noise — were carefully chosen to capture the subtle and often dispersed patterns of bacterial infections on leaves. Bacterial spots and lesions can appear across different areas of the leaf surface, often affecting both edges and central parts as showcased in Fig. 10. Rotation and horizontal flip were applied to simulate varying orientations, ensuring the model could identify bacterial symptoms regardless of leaf positioning. Translation helped introduce minor shifts in the location of bacterial spots within the frame, reflecting real-life scenarios where infected areas may not always be centered. Finally, small noise was added to mimic slight variations in texture, which could result from natural environmental factors or image capture noise. This combination of augmentations aimed to enrich the dataset by incorporating realistic transformations, improving the model's ability to generalize bacterial leaf infections while enhancing class representation.

For the Fungi class, the selected augmentations — rotation range, zoom range, and color jitter — were specifically chosen to capture the distinctive characteristics of fungal infections on leaves, which often manifest as scattered spots, discolorations, and irregular patterns as showcased in Fig. 5. Rotation and zoom augmentations were applied to simulate different viewing angles and scales, ensuring that the model could recognize fungal infections regardless of the orientation or distance of the leaf within the frame. Color jitter was added to account for the variability in color caused by fungal growth, which can result in altered shades of green, yellow, or brown. This combination of augmentations enhanced the dataset's diversity by reflecting realistic variations in how fungal infections appear, improving the model's ability to generalize to different instances of fungi-infected leaves while increasing the representation of this class in the training dataset.

For the Pest class, the selected augmentations — small noise, rotation range, horizontal flip, and color jitter — were chosen to enhance the model's ability to recognize pest damage on leaves. Pest damage is often concentrated in specific sections of the leaf where pests prefer to feed, leading to irregular patterns of missing or damaged areas as showcased in Fig. 7. Applying rotation and horizontal flip ensures the model can detect these damage patterns regardless of leaf orientation. Color jitter was included to account for the discoloration that can occur around damaged areas, where affected tissue might change to shades of brown or yellow, contrasting with healthy green portions. Small

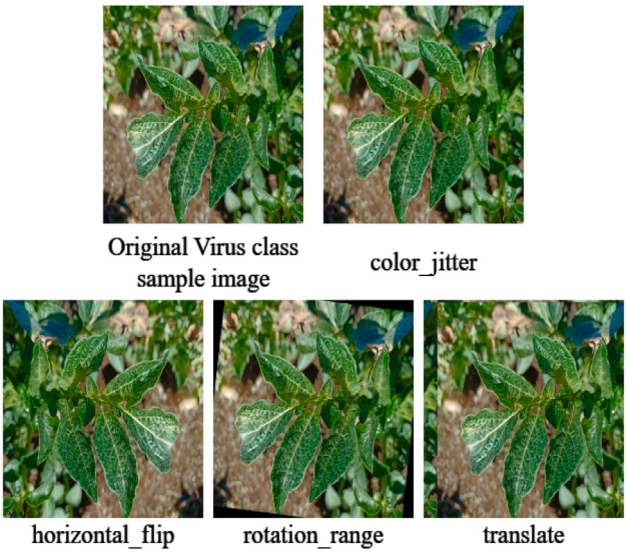


Fig. 6. Distribution of virus class images after augmentation.

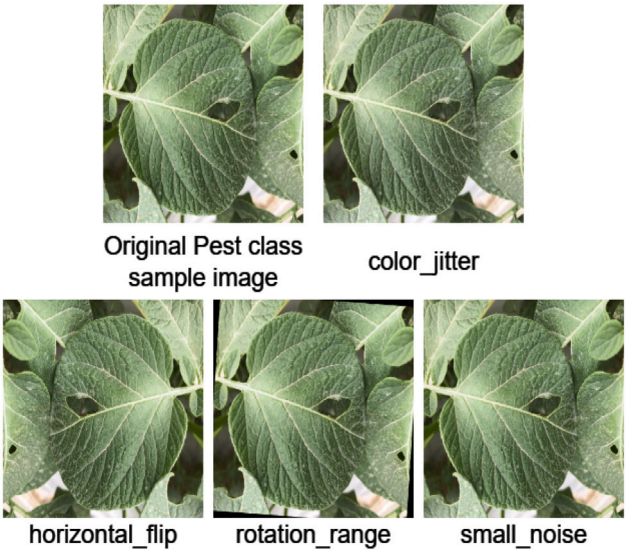


Fig. 7. Distribution of pest class images after augmentation.

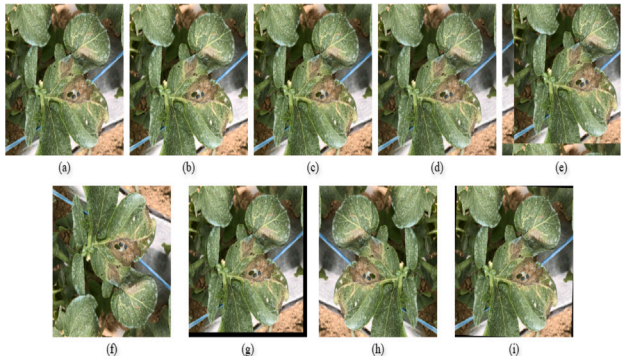


Fig. 8. Phytophthora class images after augmentation: (a) Phytophthora class original sample image, (b) brightness, (c) color jitter, (d) contrast, (e) translate, (f) vertical flip, (g) zoom range, (h) horizontal flip, (i) rotation range.

Fig. 9. Nematode class images after augmentation: (1) Nematode class original sample image, (2) brightness + contrast, (3) color jitter + contrast, (4) horizontal flip + small noise, (5) rotation range + small noise, (6) rotation range + zoom range, (7) translate, (8) brightness, (9) brightness + horizontal flip, (10) brightness + small noise, (11) color jitter, (12) color jitter + small noise, (13) color jitter + translate, (14) contrast + small noise, (15) contrast + zoom range, (16) horizontal flip, (17) horizontal flip + brightness, (18) horizontal flip + color jitter, (19) horizontal flip + contrast, (20) horizontal flip + translate, (21) horizontal flip + zoom range, (22) rotation range, (23) rotation range + brightness, (24) rotation range + color jitter, (25) rotation range + contrast, (26) rotation range + horizontal flip, (27) small noise, (28) translate + brightness, (29) translate + color jitter, (30) translate + small noise, (31) translate + zoom range, (32) vertical flip + brightness, (33) zoom range, (34) zoom range + color jitter, (35) zoom range + contrast, (36) zoom range + small noise.

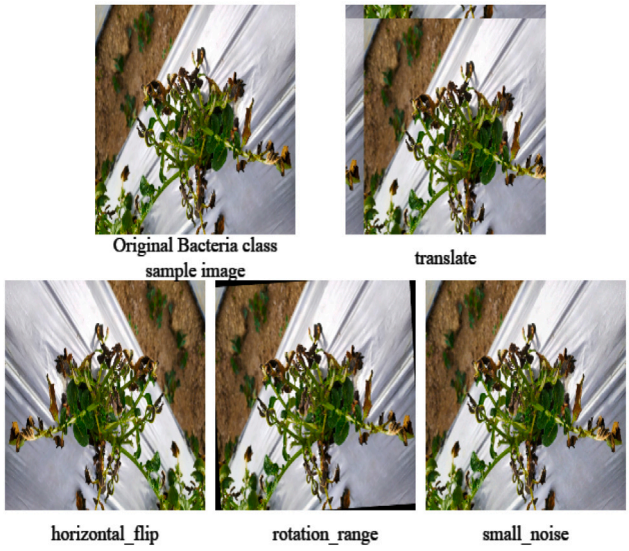


Fig. 10. Distribution of bacteria class images after augmentation.

noise was added to simulate minor environmental variations or texture differences caused by pest damage. This combination of augmentations helps the model generalize across diverse instances of pest-damaged leaves, capturing the variability in shape, color, and location of affected areas, while also increasing the dataset's representational balance for the Pest class.

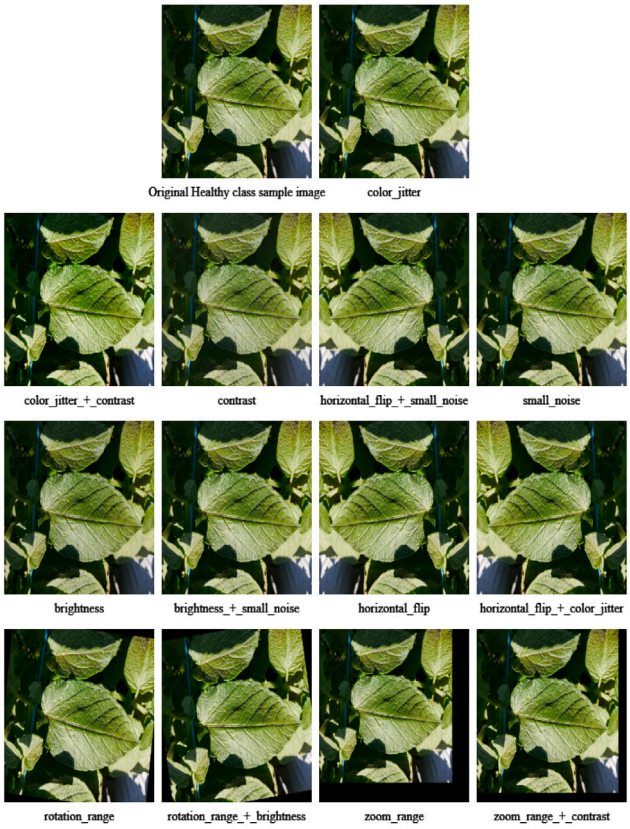


Fig. 11. Distribution of healthy class images after augmentation.

For the Virus class, the chosen augmentations — color jitter, horizontal flip, rotation range, and translation — were selected to address the distinctive visual traits that viral infections impart on leaves. Viral infections often lead to patchy discoloration, mosaic patterns, and irregular vein clearing as showcased in Fig. 6. Color jitter was applied to simulate the varied intensity and distribution of these discolorations, enabling the model to learn to detect viral patterns under different color shifts. Horizontal flip and rotation range ensure that the model can identify viral symptoms regardless of leaf orientation, as the symptoms can appear uniformly across the leaf's surface. Translation was included to mimic slight positional variations, which may occur in natural settings, especially when viewing leaves from different angles or distances. Together, these augmentations provide a diversified set of virus-infected leaf images, enhancing the model's ability to generalize and accurately identify viral infections across different visual scenarios.

The augmentations applied to each class are detailed in Table 4, which outlines the specific transformations such as rotations, flips, brightness adjustments, and noise addition used to increase dataset variability. These augmentations were applied with a range of values for each transformation, as specified in Table 5, ensuring that the model is exposed to diverse real-world conditions like different lighting, angles, and camera settings. The result of these augmentations is showcased in the figures, including "Fungi" (Fig. 5), "Virus" (Fig. 6), "Pest" (Fig. 7), "Phytophthora" (Fig. 8), "Nematode" (Fig. 9), "Bacteria" (Fig. 10), and "Healthy" (Fig. 11). These figures visually demonstrate how the augmentations simulate real-world variations in the dataset, ultimately improving the model's ability to generalize and detect plant diseases across diverse environmental conditions.

The augmentation process did not result in a perfectly balanced dataset because each augmentation technique, once selected for a particular class, was applied uniformly to all images within that class rather than selectively to a subset. To achieve a perfectly balanced dataset where all the class counts are exactly equal, the augmentation would have needed to apply certain techniques only to a smaller subset of class images. However, this research work focused on consistency and uniformity within classes, applying each selected augmentation to all images within a class ensured uniform enhancement across the dataset. This consistency was important for two reasons. First, by ensuring that all images within a class underwent the same augmentations, we maintained a similar quality and variation level across the dataset. This avoided the risk of overrepresenting certain images or patterns that could occur if only a subset of images received particular augmentations, which might introduce bias or redundant features within a class. Second, applying augmentations uniformly within each class helped prevent the model from focusing too heavily on a smaller subset with unique transformations. For instance, when an augmentation such as rotation_range was applied to the Virus class with 532 images, all 532 images underwent this transformation, rather than only a subset.

To quantify the imbalance, we calculated the Coefficient of Variation (CV), which measures the spread of class proportions relative to the mean proportion across classes. The formula for CV is given by:

$$\mathrm{CV} = \frac{\sigma}{\mu}$$

where:

- σ is the standard deviation of the class proportions, calculated as:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - \mu)^2}$$

- μ is the mean of the class proportions, calculated as:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} p_i$$

and p_i represents the proportion of each class i in the dataset.

The initial imbalance in the original dataset is reflected by a Coefficient of Variation (CV) of 0.552, indicating a high level of imbalance among the classes. After applying the augmentation process, the post-augmentation CV dropped to 0.079, showing a significant reduction in class imbalance. This substantial decrease in CV demonstrates the effectiveness of the augmentation process in improving class distribution, even though perfect balance was not achieved. The CV metric thus confirms that the dataset is more balanced post-augmentation, making it better suited for training purposes. This approach provided a streamlined augmentation process, ensuring data consistency across all images within each class while achieving a more balanced dataset, even if minor class count differences remain.

The augmentation techniques and their respective values are deliberately chosen within a controlled range to preserve the characteristics of the original training data. This careful selection ensures that the transformations, such as brightness, color jitter, or rotation, do not introduce excessive distortions that could negatively impact the model's ability to generalize.

3.5. Convolutional Neural Network (CNN)

Convolutional Neural Networks (CNNs) are specialized deep learning architectures primarily used for tasks such as image classification and object detection. CNNs are particularly effective for feature extraction, enabling them to capture essential details in images that are crucial for accurate classification. These models are typically trained on labeled datasets, which improves their ability to generalize to new, unseen data. CNNs consist of multiple layers, including convolutional layers and pooling layers, which progressively refine the input data to extract meaningful features. The learned features are then interpreted to output class predictions. Deep learning models, including CNNs, are often preferred over traditional machine learning algorithms due to their superior performance in complex tasks, as noted in the literature (Alzubaidi et al., 2021). Transfer learning plays a significant

role in this study by utilizing pre-trained models on large datasets like ImageNet and fine-tuning them on a specific image dataset. This approach enables the models to quickly adapt to new tasks and achieve better performance, which is particularly useful when the datasets share similar characteristics. In this research work, we have employed three CNN architectures: DenseNet201, ResNet152V2, and NasNetMobile. These models were selected based on their unique architectural advantages, proven effectiveness in similar tasks, and their suitability for the specific requirements of potato leaf disease detection in an uncontrolled environment.

The criteria used for selecting NasNetMobile, DenseNet201, and ResNet152V2 were primarily based on architectural strengths, proven effectiveness, and their suitability for uncontrolled environments. Each of these models has unique structural advantages that make them suitable for different aspects of image classification. For example, DenseNet201's dense connectivity pattern facilitates feature reuse, NasNetMobile's efficiency makes it ideal for mobile and resource-constrained environments, and ResNet152V2's deep architecture with residual connections enables it to capture intricate details. These models have demonstrated strong performance in similar tasks, particularly in the domain of plant disease detection, as highlighted by previous studies. For instance, DenseNet201 has been effectively used in maize disease identification, achieving a remarkable classification accuracy of 94.6% when combined with an optimized support vector machine, as demonstrated by Dash, Sethy, and Behera, (2023). Similarly, DenseNet201 has shown outstanding performance in robust weed and potato plant classification, achieving 100% accuracy when hybridized with SVM, as shown by Fauzi, Adhinata, Ramadhan, and Tanjung, (2022). Given these results, DenseNet201, despite being introduced several years ago, remains a strong contender for achieving high accuracy in various image classification tasks. The suitability of these models for handling the variability inherent in our dataset further justifies their selection.

3.5.1. NasNetMobile

NasNetMobile (Neural Architecture Search) (Zoph, Vasudevan, Shlens, and Le, 2018) was primarily designed for mobile and edge devices due to its lightweight architecture. The model's efficiency and adaptability make it an ideal choice for deployment in real-world agricultural settings where computational resources may be limited. NasNetMobile uses a Neural Architecture Search (NAS) to automatically design network architectures optimized for both accuracy and efficiency. This approach has demonstrated success in tasks such as plant disease detection, significantly improving detection accuracy while maintaining efficiency, as seen in Kamarudin and Ismail, (2022). The architecture of NasNetMobile is based on a repeating cell structure that includes reduction cells and normal cells, which together contribute to its high performance. The architecture's use of reinforcement learning to identify the most appropriate CNN structures ensures that it can adapt effectively to the diverse and challenging conditions of the potato leaf dataset used in this research. This model has been successfully used in the detection of plant diseases, further validating its selection for this study (Adedoja, Owolawi, Mapayi, and Tu, 2022). Fig. 12 displays an overview of the model's architecture.

3.5.2. ResNet152V2

ResNet152V2 (He, Zhang, Ren, and Sun, 2016) is part of the ResNet family, a neural network architecture developed by Microsoft. This model is built upon residual blocks, which allow for the training of very deep networks without suffering from the degradation problem, mainly because of the introduction of skip connections. These connections enable the model to learn identity mappings, effectively addressing the vanishing gradient problem that typically occurs in deep networks. ResNet152V2, an evolution of the original ResNet, incorporates a more refined residual learning approach, making it an excellent choice for tasks requiring high accuracy and robustness. This model has been successfully applied in various fields, including plant disease identification, as demonstrated in studies like Rachburee and Punlumjeak, (2022) and Chandra, Reddy, Sushanth, and Sujatha, (2022). Its depth and capability to capture fine-grained details make it particularly well-suited for distinguishing between the different diseases present in the potato leaf dataset. The architecture of ResNet152V2 includes a 7×7 convolutional layer with 64 filters at the beginning, which helps capture larger spatial features, followed by a series of bottleneck blocks that refine the feature maps. The use of skip connections ensures that the output of each block is added to the input, enhancing the model's ability to learn complex patterns. Fig. 13 displays an overview of the model's architecture.

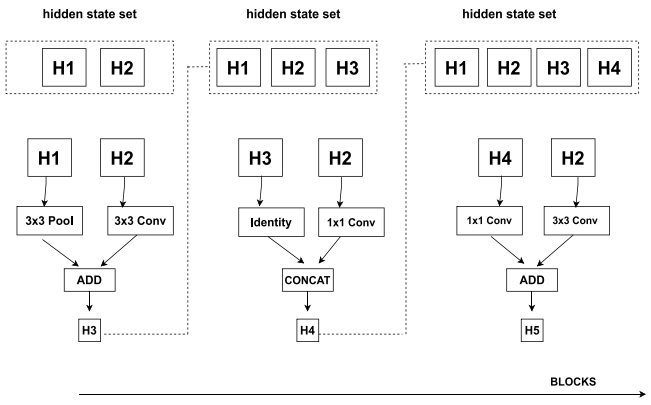


Fig. 12. NASNetMobile architecture (Tsang, 2019).

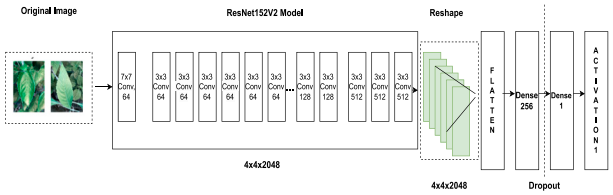


Fig. 13. ResNet152V2 architecture (Kittusamy, Krishnakumar, Aswath, Gowtham, and Vishal, 2021).

3.5.3. DenseNet201

DenseNet201 is a CNN architecture developed in 2017, known for its dense connectivity pattern, which facilitates feature reuse and enhances training efficiency. Despite being introduced several years ago, DenseNet201 remains a strong contender in various image classification tasks due to its innovative architecture. In DenseNet, each layer is connected to every other layer in a feed-forward fashion, which alleviates the vanishing gradient problem, strengthens feature propagation, and allows for the reuse of features, resulting in more compact and efficient models. This characteristic is particularly beneficial for our dataset, as it enables the model to leverage complex patterns and subtle features that are essential for distinguishing between different types of potato leaf diseases. Although DenseNet201 is not the most recent model, its architectural strengths and consistent performance justify its inclusion in this study. Fig. 14 displays an overview of the model's architecture.

The architecture of DenseNet201 includes dense blocks where the output from preceding layers is concatenated and used as input for subsequent layers, followed by a sequence of convolutional layers with a bottleneck structure and pooling layers to reduce spatial dimensions. This pattern is repeated throughout the architecture, making DenseNet201 highly efficient and accurate for the task at hand. In comparison to models used in previous studies, DenseNet201 has shown a significant improvement in accuracy, achieving 77.14% on our dataset,

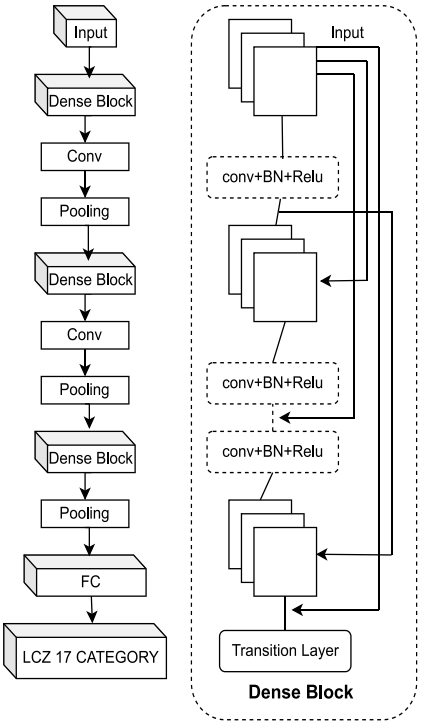


Fig. 14. DenseNet201 architecture (Kumar, Virmani, Tripathi, Agrawal, and Kumar, 2021).

compared to the 59.16% accuracy of DenseNet121 and 68.17% accuracy of ResNet50 in previous studies (Shabrina, Indarti, Maharani, Kristiyanti, Prastomo, et al., 2024). The increased accuracy can be attributed to DenseNet201's ability to effectively capture and reuse features across layers, which is particularly advantageous in datasets with complex and diverse images like ours. The model's dense connections help in reducing the number of parameters while maintaining high performance, making it an excellent choice for improving results on newer, broader datasets.

3.6. Fine-tuning

In this study, fine-tuning and regularization techniques played a crucial role in optimizing the performance of the deep learning models used for potato leaf disease classification. The transfer learning approach was employed, where pre-trained models on the ImageNet dataset (DenseNet201, ResNet152V2, and NasNetMobile) were fine-tuned on the potato leaf disease dataset. Fine-tuning allowed the models to adapt to the specific task of disease classification by leveraging previously learned features and adjusting them to the new dataset. Additionally, to prevent overfitting and improve the generalization ability of the models, early stopping and L2 regularization were applied.

Transfer learning was implemented using the DenseNet201, ResNet152V2, and NasNetMobile architectures, all pre-trained on the ImageNet dataset. These models were selected based on their well-established success in image classification tasks, particularly in the domain of plant disease detection. By leveraging their pre-trained weights, which contain valuable information learned from large-scale image datasets, the models were adapted to the potato leaf disease classification task through a process called fine-tuning. Fine-tuning enables the adjustment of these pre-trained weights to better suit the specific features and characteristics of the potato leaf dataset, thus improving model performance on the target task. The fine-tuning process began with model initialization, where the pre-trained ImageNet weights were loaded into each model. The existing fully connected (FC) layers were then replaced with new layers specifically designed for the potato leaf disease classification task. To better align with the dataset, the classification head was modified by removing the original final layers and adding a Global Average Pooling (GAP) layer, which helps in reducing the spatial dimensions of the feature maps and preventing overfitting. This was followed by a Dense layer with 1024 units, which included a ReLU activation function and L2 regularization to prevent the model from overfitting. The final output layer consisted of 7 units, each corresponding to one of the seven disease categories in the dataset, and a softmax activation was used to provide the class probabilities.

In the initial phase of fine-tuning, the pre-trained layers of the model were frozen. This means that the pre-trained weights remained unchanged, and only the newly added layers were trained. This approach allows the model to retain the general feature representations learned from the ImageNet dataset while gradually adapting to the new classification task. Since the early layers of the network typically capture more generic features (such as edges, textures, and colors), freezing these layers ensures that the model does not lose this valuable information, which is transferable to a wide range of visual tasks. After training the new layers for a few epochs, a portion of the pre-trained layers was unfrozen to allow for additional fine-tuning of the entire network. This step was essential in enabling the model to adjust the deeper learned representations from ImageNet to the specific characteristics of the potato leaf disease dataset. By unfreezing and fine-tuning the layers, the model could more effectively learn the subtle distinctions between different disease types while still benefiting from the original knowledge gained from large-scale data. This step ensured that the model was able to refine its ability to detect and classify diseases under varying conditions, such as different lighting or leaf orientations.

To optimize the training process, the Adam optimizer was employed with a learning rate of 0.00001, which provided an adaptive learning strategy that efficiently updated the model's weights during fine-tuning. Adam was chosen due to its ability to handle sparse gradients and its general robustness in deep learning applications. The categorical cross-entropy loss function was used to manage the multi-class classification problem, ensuring that the model's predictions were accurate across all seven disease categories. The combination of Adam and categorical cross-entropy helped the model converge more rapidly and stably during training, even on a limited dataset. Overall, fine-tuning the pre-trained models allowed for faster convergence and better performance on the potato leaf disease dataset. The process capitalized on the previously learned general representations while refining them to address the specific disease classification task, enhancing the model's ability to generalize to new and unseen data within the agricultural domain.

3.7. Hyperparameter selection and tuning

In pursuit of enhanced performance, deliberate refinements were applied to the model's hyperparameters. The hyperparameters were carefully selected based on prior knowledge, best practices in deep learning literature, and empirical testing through iterative experimentation. The primary objective was to optimize the model's performance while balancing computational resources and time constraints. Each value was chosen with a focus on achieving stable convergence and avoiding overfitting while ensuring efficient use of the available resources.

Initial experimentation began with commonly used batch sizes of 32 and 64. However, due to hardware resource constraints, particularly memory capacity, a smaller batch size of 16 was selected. This choice ensured that the training process remained manageable without risking memory overflows or exceeding the available resources during training. Selecting the learning rate is widely recognized as critical to the overall training process. Based on established practices in deep learning, starting with a lower learning rate, such as 0.00001, is commonly recommended to ensure smoother learning curves and stable convergence, particularly for complex architectures like DenseNet201, ResNet152V2,

and NasNetMobile. While higher learning rates (e.g., 0.001) can lead to faster convergence, they are also known to increase the risk of instability and overshooting the optimal loss. The chosen learning rate of 0.00001 aligns with best practices for these architectures, allowing the model to make incremental updates and achieve steady convergence.

The Adam Optimizer (Kingma, 2014) was selected as the most suitable optimizer for this problem due to its adaptive learning rate capabilities, making it an excellent choice for handling the variability and complexity present in our potato leaf disease dataset. Adam adjusts the learning rate for each parameter individually, depending on the magnitude of the gradients, allowing it to perform well across different layers and models. Although a base learning rate (0.00001) is provided, Adam fine-tunes the learning rate dynamically during training, enabling more efficient convergence. This adaptability, especially in handling complex and imbalanced datasets, makes Adam a better choice for this problem compared to traditional optimizers like stochastic gradient descent (SGD). Its ability to deliver faster convergence and more stable updates, even in variable data environments, further reinforces its selection for this task. Categorical Cross-Entropy was used as the loss function, as it is well-suited for multi-class classification problems like the one at hand, which involved different potato leaf diseases. No alternative loss functions were tested, as cross-entropy loss is widely accepted as the standard choice for such tasks.

In summary, the manual hyperparameter tuning process was guided by iterative experimentation and practical constraints, focusing on stability, convergence, and efficient resource utilization. The final hyperparameter configurations, outlined in Table 14, represent the best trade-offs between performance and resource efficiency.

Table 14
Hyperparameters table.

Hyperparameters	Value
Batch size	16
Learning rate	0.00001
Epoch	20
Optimizer	Adam Optimizer
Loss	Categorical Crossentropy

3.8. Evaluation criteria

The study's outcomes hinge upon the evaluation of four pivotal assessment criteria: Accuracy, Precision, Recall, and F1-Score. Each criterion assumes a unique role in gauging the model's performance. Accuracy serves as a primary metric, denoting the proportion of correctly predicted data points within the entirety of the dataset. Precision scrutinizes the exactness of classification by measuring the precise identification of relevant samples within the dataset. Conversely, Recall focuses on the accurate identification of positive samples concerning the total positives present in the dataset, offering a comprehensive assessment of the model's performance. Furthermore, the F1 score, a critical metric, represents the harmonic mean of Precision and Recall, providing a well-balanced measure of the model's efficacy. The formulas for these metrics, utilizing True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN), are outlined below, elucidating the quantitative evaluation methodology employed in this study.

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

$$F1Score = 2\frac{PrecisionRecall}{Precision + Recall} \tag{3}$$

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{4}$$

3.9. Overfitting considerations

The dataset (Shabrina et al., 2023) used in this research has a significant class imbalance, which introduces the risk of overfitting when applying fine-tuned transfer learning models. This class imbalance can be clearly observed from the distribution shown in Fig. 2. Overfitting occurs when the model becomes too specialized in recognizing patterns within the training data, especially in classes with more samples, leading to poor generalization on unseen data. In a class-imbalanced dataset, the model tends to focus more on the over-represented classes, thereby neglecting the minority classes. Transfer learning models, which rely on pre-trained weights, can adapt too closely to the provided dataset, especially when fine-tuning on a limited number of images, leading to overfitting. This means the model learns both relevant and irrelevant details from the data, which ultimately affects its ability to perform well on new, unseen data. To mitigate the risk of overfitting, several strategic data augmentation techniques were applied, as explained in the data augmentation subsection. These techniques, specifically designed to address the class imbalance in the dataset, are outlined in Table 4, which provides a detailed overview of the transformations such as flipping, rotation, scaling, and adding noise. By introducing variability in the training set, these augmentations help the model generalize better by learning meaningful patterns rather than memorizing the training data. This is particularly beneficial for the minority classes, as it balances the dataset and prevents the model from overfitting to the majority class. As a result, these augmentations improve the model's robustness, enhancing its performance on both the validation and test sets. Another crucial technique used to combat overfitting is early stopping (Prechelt, 2002). Early stopping monitors the model's performance on the validation data during training and halts the process once performance starts to degrade, preventing further overfitting. In our research work, we applied early stopping to monitor the validation loss and halt training if no improvement was observed after 3 consecutive epochs. We chose this 'patience' value to give the model sufficient time to learn and improve while avoiding unnecessary prolonged training that could lead to overfitting. Additionally, by setting 'restore_best_weights=True', the model automatically reverts to the weights from the epoch with the best validation performance, ensuring that we retain the optimal configuration of the model. This approach was applied to ensure that the model was trained just enough to capture useful features without overfitting to the noise in the dataset, enhancing its generalization ability.

In our research, we applied L2 regularization (Lewkowycz and Gur-Ari, 2020) (also known as weight decay) to the network's weights during training to prevent them from becoming excessively large. This method encourages the model to learn smaller, more generalized weights, reducing the risk of overfitting to the training data. Specifically, we implemented L2 regularization in the fully connected layers of each transfer learning model used in our work. A regularization factor of 0.001 was applied to the dense layers, adding a penalty proportional to the square of the weights to the loss function. This strategy was used to discourage the models from learning overly complex solutions that might not generalize well to unseen data. L2 regularization was particularly important in fine-tuning transfer learning models, as it helped prevent the models from becoming overly specialized to the training set, ultimately improving their generalization performance on the test data.

To systematically evaluate the impact of fine-tuning and regularization techniques on reducing overfitting and improving generalization, we tested three distinct experimental settings. Each setting aimed to progressively enhance the model's ability to generalize from the training data while avoiding overfitting. In the first experimental setting, the models were trained on the original dataset without any additional techniques such as data augmentation, early stopping, or regularization. This baseline approach allowed us to assess the raw performance

Table 15
System configuration.

IDE Used	Kaggle
Processor	Intel Xeon @2.3 GHz
Cache	46 MB
RAM	16 GB
GPU	16 GB NVIDIA TESLA P100

of the models when directly applied to the unaltered dataset providing baseline results. However, in datasets with significant class imbalances, such as ours, training without augmentation or regularization can lead to overfitting, as the model tends to focus more on the majority classes, neglecting the minority ones. This setup highlighted the need for techniques to enhance generalization. The second setting introduced data augmentation techniques such as rotation, flipping, and brightness adjustments to the training set. By artificially increasing the diversity of the training data, these augmentations help the model learn to recognize a wider variety of patterns, thereby improving its ability to generalize to unseen data. Early stopping was also incorporated into this experiment to prevent the model from overfitting during training. Early stopping helps by monitoring the model's performance on a validation set and halting the training process once performance ceases to improve, thus avoiding overfitting on the training data. In the final experimental setting, we combined both data augmentation and early stopping with L2 regularization. L2 regularization adds a penalty to the model's weights during training, encouraging smaller and more generalized weights. This technique helps prevent the model from becoming overly complex, which can occur when the model learns patterns that are specific to the training data rather than generalizing well to new data. By applying this combination of techniques, we aimed to promote robust learning and reduce the likelihood of the model overfitting to the training set. These experimental settings were carefully designed to progressively build upon each other, with each step introducing additional mechanisms to enhance the model's ability to generalize effectively. The intention was to evaluate how the combination of data augmentation, early stopping, and L2 regularization could work together to reduce overfitting and improve the overall performance of the model.

3.10. Hardware and software setup

This research work included an examination of three distinct models. We used libraries that included Keras, Pandas and Scikit-learn in Python, which was the programming language that we used in this research. We noted that the compilation phase had seen advantages from the Kaggle cloud-based environment, which is well known for its smooth flexibility and absence of setup requisites. Kaggle provides a customizable ecosystem for Jupyter Notebooks. Kaggle also includes an impressive number of GPU services which include 16 GB NVIDIA TESLA P100 GPU and an Intel Xeon 2.3 GHz processor replete with a 46 MB cache and 16 GB of RAM. It amplifies the computational strength as illustrated by the hardware configuration detailed in Table 15.

3.11. Algorithmic framework

Algorithm 1 provides a concise overview of the fine-tuning process employed in this study.

Algorithm 1 Fine-Tuning Process for Potato Leaf Disease Classification

Input: Original dataset of potato leaf images with labels
Output: Trained CNN model for disease classification
Step 1: Set Hyperparameters
Set optimizer (Adam), learning rate (1×10^{-5}), batch size (16), epochs (up to 20), loss function (categorical cross-entropy), L2 regularization factor ($\lambda = 0.001$), early stopping patience (3 epochs).
Step 2: Data Preprocessing
Resize images to 128×128 pixels and normalize pixel values to [0, 1].
Step 3: Data Splitting
Split dataset into training (70%), validation (20%), and test (10%) sets.
Step 4: Data Augmentation
Apply augmentations (on training set) to address class imbalance:
foreach *class in training set* **do**
  if *class is underrepresented* **then**
    Apply augmentations from categories: Basic Transformations, Brightness Adjustments, Color and Contrast Adjustments, Zoom Modifications, Noise Additions, Combined Position and Brightness Modifications, Rotation and Lighting Combinations.
  else
    Apply augmentations from categories: Basic Transformations and Brightness Adjustments (if required).
  end
end
Step 5: Model Preparation
Select pre-trained model from [DenseNet201, ResNet152V2, NasNetMobile].
Replace top layers with:

- Global Average Pooling layer
- Dense layer (1024 units, ReLU activation, L2 regularization $\lambda = 0.001$)
- Output layer (7 units, softmax activation)

Freeze pre-trained layers.
Step 6: Model Compilation
Compile the model with optimizer (Adam), loss function (categorical cross-entropy), and metric (accuracy).
Step 7: Initial Training Phase
Train the model on training data with early stopping (patience = 3).
Step 8: Fine-Tuning Phase
Unfreeze some pre-trained layers, re-compile the model, and continue training with early stopping (patience = 3).
Step 9: Model Evaluation
Evaluate the model on test data; calculate accuracy, precision, recall, F1-score; generate confusion matrix and classification report.

4. Results

4.1. Performance analysis of proposed models

In this subsection, we present a detailed performance analysis of the proposed models using various evaluation metrics, including precision, recall, F1-score, accuracy, and test loss. Table 16 provides a comparative assessment of three models: NasNetMobile, ResNet152V2, and DenseNet201, evaluated on the original dataset, as well as on augmented datasets with early stopping and L2 regularization. The results demonstrate how each model's performance varies with augmentation techniques, highlighting the impact of strategic augmentation and regularization in reducing overfitting and improving generalization. The evaluation of NasNetMobile, ResNet152V2, and DenseNet201 was conducted across three configurations: on the original dataset, on the augmented dataset with early stopping, and on the augmented dataset with both early stopping and L2 regularization. These variations provide insights into the effectiveness of augmentation and regularization techniques in improving model performance and addressing overfitting.

The training accuracy results for DenseNet201, ResNet152V2, and NasNetMobile across the original dataset, augmented dataset with early stopping, and augmented dataset with both early stopping and L2 regularization are shown in Fig. 15. From the graph, we observe that the models trained on the original dataset showed slower improvement

Table 16
Performance evaluation of different models on the original dataset and augmented datasets.

Model name	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)	Test loss
NasNetMobile (original dataset)	70.55	69.21	69.44	69.21	0.8651
ResNet152V2 (original dataset)	66.03	66.03	66.88	66.03	1.3490
DenseNet201 (original dataset)	77.14	73.96	75.20	77.14	0.8793
NasNetMobile (Augmented + Early Stopping)	73.21	66.48	68.03	68.25	1.0541
ResNet152V2 (Augmented + Early Stopping)	63.06	64.71	63.40	64.13	1.0767
DenseNet201 (Augmented + Early Stopping)	75.26	79.14	76.83	77.14	0.6865
NasNetMobile (Augmented + L2 norm + Early Stopping)	60.87	63.24	61.73	62.86	2.3655
ResNet152V2 (Augmented + L2 norm + Early Stopping)	65.41	64.00	64.15	64.76	2.2572
DenseNet201 (Augmented + L2 norm + Early Stopping)	77.13	79.38	78.14	77.14	1.3507

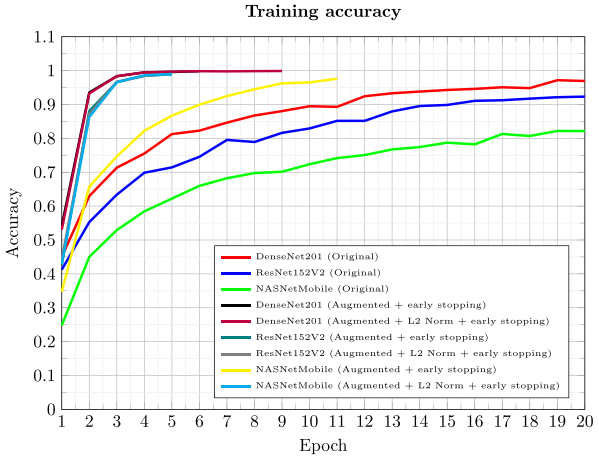


Fig. 15. Training accuracy for all 3 models over 20 epochs on the original dataset and augmented dataset with the three experimental settings each.

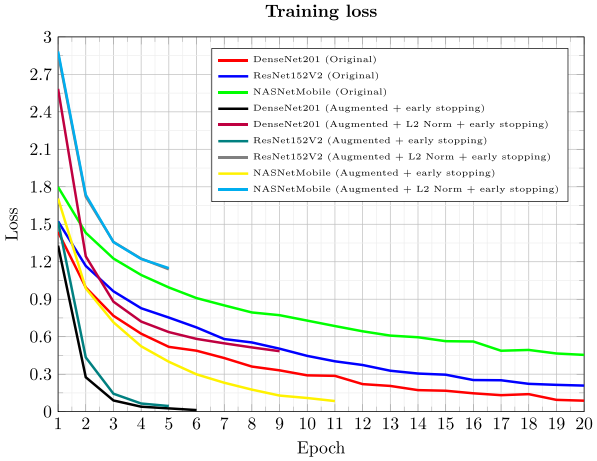


Fig. 16. Training loss for all 3 models over 20 epochs on the original dataset and augmented dataset with the three experimental settings each.

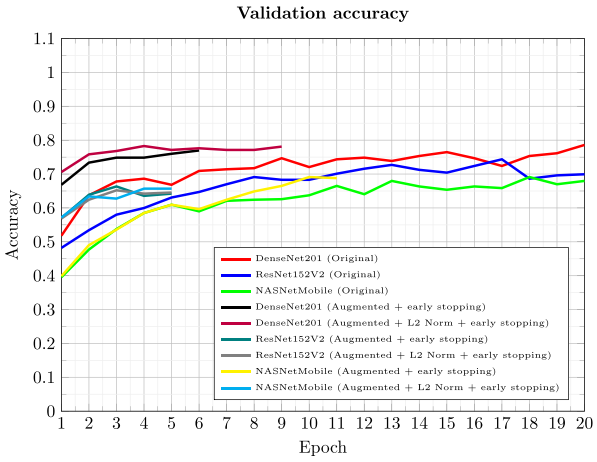


Fig. 17. Validation accuracy for all 3 models over 20 epochs on the original dataset and augmented dataset with the three experimental settings each.

in accuracy during the early epochs. Among these, DenseNet201 performed the best, with accuracy increasing steadily from around 45% in the first epoch to 94.28% by epoch 15. In contrast, NasNetMobile started with lower accuracy (around 24.8%) and reached a peak of approximately 82.13% by epoch 20. ResNet152V2 displayed a similar trend to NasNetMobile, but with slightly higher accuracy across the epochs. When data augmentation and early stopping were introduced, all models saw significant improvements in their training accuracy. DenseNet201 (Augmented + Early Stopping) quickly reached a very high accuracy, surpassing 99.5% by epoch 5, indicating that the model benefited greatly from the additional augmented data. NasNetMobile (Augmented + Early Stopping) and ResNet152V2 (Augmented + Early Stopping) also improved significantly, with NasNetMobile reaching around 96.51% by epoch 10, and ResNet152V2 reaching 97.6% by epoch 11, showing that augmentation and early stopping helped the models generalize better and learn faster. However, when L2 regularization was added along with early stopping, the training accuracy showed a slight decline in the later epochs, as L2 regularization penalized large weights to prevent overfitting. DenseNet201 (Augmented + L2 Norm + Early Stopping) maintained a high accuracy, stabilizing at around 98.88%, while ResNet152V2 (Augmented + L2 Norm + Early Stopping) reached 96.88% by epoch 5, and NasNetMobile (Augmented + L2 Norm + Early Stopping) reached approximately 96.24% by epoch 9.

The training loss results for DenseNet201, ResNet152V2, and NasNetMobile across the original dataset, augmented dataset with early stopping, and augmented dataset with both early stopping and L2 regularization are illustrated in Fig. 16. This graph tracks the reduction in loss over 20 epochs, allowing us to analyze how effectively each model minimizes error during training across the different experimental settings. In the original dataset configuration, all three models showed relatively higher initial loss values, which steadily decreased over the epochs. DenseNet201 began with a training loss of 1.4387 in the first epoch and achieved a low loss of around 0.0877 by epoch 20, demonstrating effective learning over time. Similarly, ResNet152V2 started with a higher initial loss of 1.5231, but by the final epoch, it reduced the loss to 0.2085. NasNetMobile, which started with a relatively high loss of 1.798 in the first epoch, showed a slower decline compared to the other models, ending with a loss of 0.4544 by epoch 20. This slower reduction suggests that NasNetMobile faced more difficulty in learning from the original dataset compared to the other models. When data augmentation and early stopping were applied, all models showed a faster and more consistent decline in training loss. DenseNet201 (Augmented + Early Stopping) saw a sharp reduction in loss, reaching as low as 0.026 by epoch 5 and stabilizing at even lower values in subsequent epochs. This suggests that data augmentation

significantly improved the model's ability to learn effectively from diverse training data. ResNet152V2 (Augmented + Early Stopping) and NasNetMobile (Augmented + Early Stopping) followed similar trends, with ResNet152V2 reducing its loss to 0.0847 by epoch 11 and NasNetMobile reducing its loss to 0.1091 by epoch 10. When L2 regularization was added alongside augmentation and early stopping, we observe a more controlled and gradual reduction in training loss. DenseNet201 (Augmented + L2 Norm + Early Stopping) maintained a steady decline, reaching a loss of 0.5141 by epoch 8 and continuing to drop, though not as sharply as in the configuration without L2 regularization. Similarly, ResNet152V2 (Augmented + L2 Norm + Early Stopping) and NasNetMobile (Augmented + L2 Norm + Early Stopping) showed more stable loss curves, with ResNet152V2 reaching 0.4839 by epoch 9 and NasNetMobile stabilizing at 0.1286 by epoch 9. The presence of L2 regularization helped prevent overfitting, ensuring that the models learned more generalizable patterns rather than memorizing the training data. Overall, the training loss analysis indicates that data augmentation and early stopping played significant roles in reducing loss more effectively and that L2 regularization provided further control over the learning process, preventing the models from overfitting. Each of these configurations demonstrated clear improvements in learning efficiency compared to training on the original dataset alone.

The validation accuracy results for DenseNet201, ResNet152V2, and NasNetMobile across the original dataset, augmented dataset with early stopping, and augmented dataset with both early stopping and L2 regularization are shown in Fig. 17. This graph provides an in-depth look into the generalization capability of each model, showcasing how well they perform on unseen validation data under various experimental settings. For the original dataset, the models showed moderate validation accuracy, with gradual improvements over time. DenseNet201 outperformed the other models, starting at around 51.8% in the first epoch and steadily increasing to 78.59% by epoch 20. This demonstrates that DenseNet201 was able to generalize fairly well to the validation data, even without the benefit of data augmentation or regularization. ResNet152V2 and NasNetMobile, in comparison, started at lower validation accuracies, with ResNet152V2 beginning at 48.2% and NasNetMobile at 39.54%. By epoch 20, ResNet152V2 reached a validation accuracy of 69.93%, while NasNetMobile improved to 67.97%, indicating that while they learned from the training data, they were more prone to overfitting compared to DenseNet201. When data augmentation and early stopping were introduced, all models showed a noticeable boost in validation accuracy. DenseNet201 (Augmented + Early Stopping) quickly achieved strong performance, reaching 75.98% by epoch 5 and maintaining high accuracy across subsequent epochs. ResNet152V2 (Augmented + Early Stopping) also improved significantly, reaching 71.57% by epoch 12, while NasNetMobile (Augmented + Early Stopping) saw more gradual improvement, reaching 69.12% by epoch 18. The augmented data helped the models generalize better by providing a more diverse set of training samples, and early stopping helped prevent overfitting, leading to more stable performance on the validation data. When L2 regularization was added alongside augmentation and early stopping, the models demonstrated more stable validation accuracy over the epochs. DenseNet201 (Augmented + L2 Norm + Early Stopping) achieved consistent validation accuracy, reaching around 77.12% by epoch 5 and stabilizing at that level. ResNet152V2 (Augmented + L2 Norm + Early Stopping) and NasNetMobile (Augmented + L2 Norm + Early Stopping) followed a similar trend, showing slight improvements but with a more controlled increase compared to their counterparts without L2 regularization. ResNet152V2 reached 74.35% validation accuracy by epoch 17, and NasNetMobile stabilized at 69.12% earlier than in the previous configurations. This suggests that the use of L2 regularization helped reduce overfitting, leading to better generalization on the validation set, especially for the models prone to overfitting, such as NasNetMobile.

The validation loss results for DenseNet201, ResNet152V2, and NasNetMobile across the original dataset, augmented dataset with early stopping, and augmented dataset with both early stopping and L2 regularization are displayed in Fig. 18. This graph illustrates how well each model generalizes to unseen data by tracking the loss over the validation set for the various experimental settings. In the original dataset, all models initially experienced relatively high validation loss values, with DenseNet201 starting at 1.2884, ResNet152V2 at 1.3491, and NasNetMobile at 1.591 in the first epoch. Over time, the loss decreased for all three models, with DenseNet201 showing the most consistent and effective loss reduction, achieving a final validation loss of 0.8048 by epoch 20. ResNet152V2 reduced its loss to 1.0816 by the final epoch, while NasNetMobile, despite showing improvement, maintained a slightly higher final loss of 0.9194. This indicates that while NasNetMobile was able to reduce its validation loss, it struggled more than the other models to generalize to unseen data using only the original dataset. When data augmentation and early stopping were applied, the validation loss decreased more consistently across all models. DenseNet201 (Augmented + Early Stopping) saw significant reductions in loss, starting at 0.8416 and reaching a low of 0.7051 by epoch 18. This improvement highlights the effectiveness of data augmentation and early stopping in providing the model with more diverse data, which helped it generalize better. ResNet152V2 (Augmented + Early Stopping) also showed improvement, with the validation loss reducing from 1.1441 in the first epoch to 0.8788 by epoch 12, indicating a more controlled learning process. NasNetMobile (Augmented + Early Stopping), although improving as well, showed a higher starting loss of 2.425, which gradually reduced to 1.142 by epoch 11. When L2 regularization was added alongside data augmentation and early stopping, the models demonstrated even more stable loss reduction. DenseNet201 (Augmented + L2 Norm + Early Stopping) achieved a steady loss curve, reducing from 1.8347 in the first epoch to 1.3183 by epoch 9, and stabilizing around that range. ResNet152V2 (Augmented + L2 Norm + Early Stopping) similarly maintained stability, with the loss reducing from 2.4559 to 1.3183 by epoch 9. NasNetMobile (Augmented + L2 Norm + Early Stopping), although showing improvement, had a higher initial loss (2.425) and maintained a higher final loss (1.142) compared to other models. The use of L2 regularization, in this case, helped control overfitting, particularly for the models that showed tendencies toward overfitting during training. Overall, the validation loss analysis shows that while data augmentation and early stopping were critical in improving generalization and reducing validation loss, the addition of L2 regularization helped stabilize the learning process, leading to more consistent results across models, especially for DenseNet201 and ResNet152V2. L2 regularization often leads to slightly higher validation loss because it penalizes large weights, effectively limiting the model's ability to fit the training data too closely. While this increases the validation loss, it helps prevent overfitting by encouraging the model to learn simpler, more generalizable patterns. As a result, the model is better equipped to perform well on unseen data, even though the immediate validation loss might appear higher.

4.1.1. Results for NasNetMobile (testing data)

The detailed classification report of all three experiments utilizing NASNetMobile is showcased in Table 17.

In the baseline evaluation on the original imbalanced dataset, NasNetMobile achieved an overall accuracy of 69.21%, with a macro-average precision of 68.30%, recall of 65.74%, and F1-score of 66.29%. These results suggest that the model performs reasonably well but faces challenges due to the inherent class imbalance in the dataset. For instance, Nematode, which had only 68 images in the original training set and 8 samples in the test set, showed a low recall of 37.5% and an F1-score of 46.15%, indicating the model's difficulty in identifying this minority class. In contrast, Bacteria, the largest class with 569 original images and 1990 images after augmentation, achieved perfect precision (100%) and a recall of 79.31%, resulting in an F1-score of 88.46%. This high precision indicates that the model tends to favor the majority class at the expense of minority classes. Other classes, such

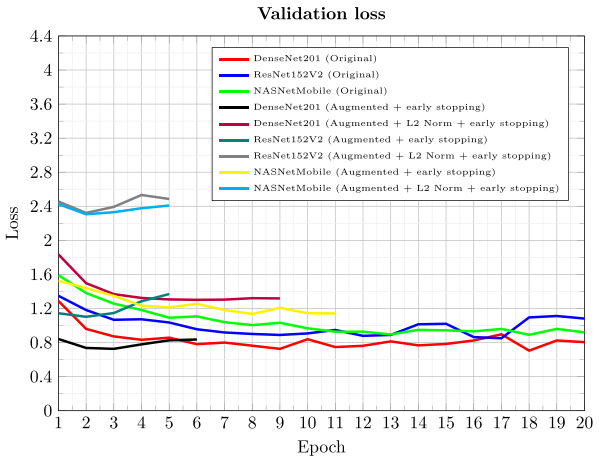


Fig. 18. Validation loss for all 3 models over 20 epochs on the original dataset and augmented dataset with the three experimental settings each.

as Fungi and Healthy, which have moderate representation, also show underperformance with F1-scores of 61.74% and 59.57%, respectively, reflecting the impact of class imbalance on model learning.

After applying augmentation techniques and early stopping, NasNetMobile's performance saw notable improvements, particularly for minority classes. The overall accuracy decreased slightly to 68.25%, but the macro-average precision increased to 73.21%, reflecting an improvement in the model's ability to detect minority classes. This improvement is evident in the Nematode class, where precision increased significantly from 60% to 80%, though recall remained at 50%, leading to a higher F1-score of 61.54%. For Bacteria, there was a slight drop in precision from 100% to 95.83%, but recall remained stable at 79.31%, and the F1-score decreased marginally to 86.79%, indicating the model's improved generalization across different classes. Healthy class performance also improved, with precision rising to 84.61% from 53.85%, although recall dropped slightly, resulting in an F1-score of 64.71%. However, the Fungi class, which had 748 original images and 2092 after augmentation, experienced a slight decline in performance, with precision dropping to 64.81% and recall to 46.05%, leading to an F1-score of 53.85%.

In the third experiment, which combined augmentation with both early stopping and L2 regularization, the overall accuracy declined further to 62.86%, but this setting improved the balance between precision and recall across most classes. For Bacteria, precision dropped to 87.04%, while recall increased to 81.03%, resulting in a slightly lower F1-score of 83.93%, suggesting a better balance between precision and recall. The Nematode class exhibited more consistent performance, with precision improving to 55.56% and recall increasing to 62.50%, leading to an F1-score of 58.82%. However, Fungi continued to struggle, with precision and recall dropping to 53.62% and 48.68%, respectively, resulting in an F1-score of 51.03%. Pest, another moderately represented class with 611 original images and 2135 after augmentation, also saw a decline in precision and recall, resulting in an F1-score of 47.37%. On the other hand, Healthy experienced a substantial drop in both precision and recall, bringing its F1-score down to 47.62%.

Overall, the augmentation techniques improved precision for several minority classes, as seen in the Nematode and Healthy classes. However, some classes, particularly Fungi and Pest, continued to underperform despite the adjustments. While augmentation improved the model's ability to generalize and detect minority classes, the application of L2 regularization helped mitigate overfitting, particularly in larger classes like Bacteria, which maintained relatively stable recall and a balanced F1-score. However, some smaller classes, like Fungi, continued to experience challenges in achieving optimal precision and recall

Table 17
Classification report for NasNetMobile (Original, Augmented with Early Stopping, and Augmented with Early Stopping + L2 regularization).

Class	Precision	Recall	F1-Score	Support
Original dataset				
Bacteria	1.000000	0.793103	0.884615	58
Fungi	0.630137	0.605263	0.617450	76
Healthy	0.538462	0.666667	0.595745	21
Nematode	0.600000	0.375000	0.461538	8
Pest	0.552239	0.596774	0.573643	62
Phytophthora	0.714286	0.694444	0.704225	36
Virus	0.746032	0.870370	0.803419	54
Accuracy			0.692063	
Macro avg	0.683022	0.657375	0.662948	315
Weighted avg	0.705514	0.692063	0.694411	315
Augmented dataset (Early Stopping)				
Bacteria	0.958333	0.793103	0.867925	58
Fungi	0.648148	0.460526	0.538462	76
Healthy	0.846154	0.523810	0.647059	21
Nematode	0.800000	0.500000	0.615385	8
Pest	0.523810	0.709677	0.602740	62
Phytophthora	0.666667	0.833333	0.740741	36
Virus	0.681818	0.833333	0.750000	54
Accuracy			0.682540	
Macro avg	0.732133	0.664826	0.680330	315
Weighted avg	0.705734	0.682540	0.680351	315
Augmented dataset (Early Stopping + L2 Regularization)				
Bacteria	0.870370	0.810345	0.839286	58
Fungi	0.536232	0.486842	0.510345	76
Healthy	0.476190	0.476190	0.476190	21
Nematode	0.555556	0.625000	0.588235	8
Pest	0.519231	0.435484	0.473684	62
Phytophthora	0.636364	0.777778	0.700000	36
Virus	0.666667	0.814815	0.733333	54
Accuracy			0.628571	
Macro avg	0.608658	0.632351	0.617296	315
Weighted avg	0.624701	0.628571	0.623299	315

due to the inherent complexity of detecting minority classes in an imbalanced dataset. The macro-average F1-score showed improvements after augmentation, but the final experiment with L2 regularization led to a slight decline in performance across most metrics, particularly for minority classes.

The confusion matrices for the NASNetMobile model across different conditions are illustrated in Figs. 19, 20, and 21. In Fig. 19, which represents the original dataset, we observe that the model performs well on the Bacteria class with 50 correct predictions but shows some confusion in distinguishing Fungi and Pest, as indicated by misclassifications in these classes. When early stopping is applied (Fig. 20), the Bacteria class slightly improves to 46 correct predictions, but Fungi class performance remains steady, with 35 correct predictions. One notable change is a slight improvement in the Healthy class, which shows a reduction in misclassifications in Fig. 20, while the Pest class gains notable improvement, with 44 correct predictions compared to 36 in Fig. 19. However, the application of L2 regularization along with early stopping (Fig. 21) introduces further refinements. The Fungi class sees a slight improvement to 37 correct predictions, while Bacteria remains consistent at 47 correct classifications. Misclassifications in the Nematode and Pest classes remain mostly stable, with a small decrease in errors. The Phytophthora class shows a reduction in confusion errors, with the model classifying 27 instances correctly. Overall, the changes from Figs. 19 to 21 reflect gradual improvement in certain classes, particularly Pest and Fungi, while other classes, such as Healthy and Bacteria, remain relatively stable. The class-wise accuracies for each experimental setting are showcased in Table 18 which are calculated by taking the ratio of the true positives to the support values of that particular class.

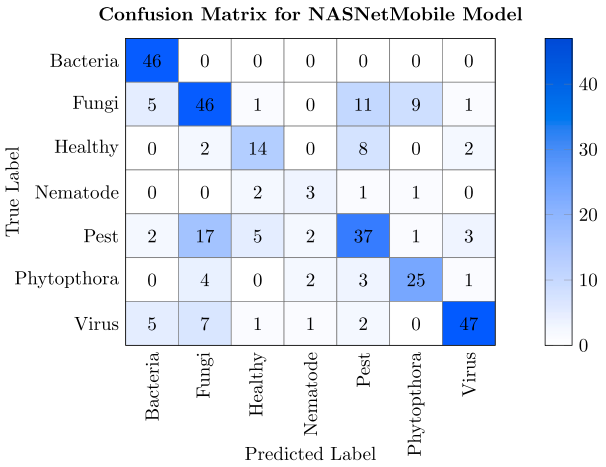


Fig. 19. Confusion matrix for NasNetMobile (original dataset).

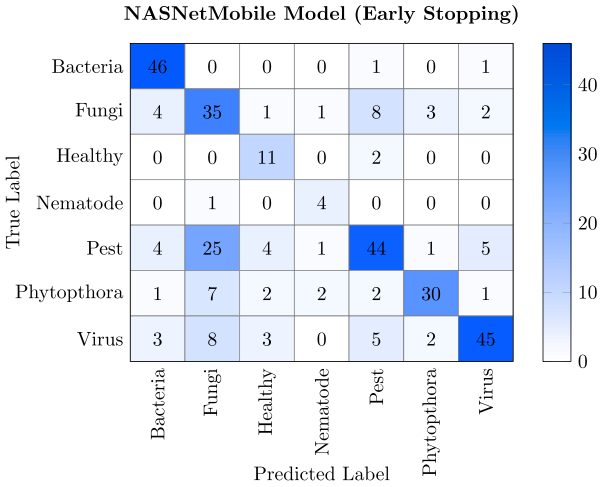


Fig. 20. Confusion matrix for NasNetMobile (Augmented + early stopping).

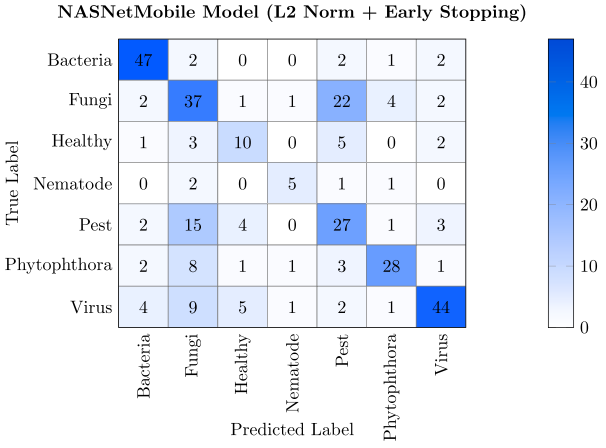


Fig. 21. Confusion matrix for NasNetMobile (Augmented + L2 regularization + early stopping).

4.1.2. Results for ResNet152V2 (Testing data)

The detailed classification report of all three experiments utilizing ResNet152V2 is showcased in Table 19. In the baseline evaluation on the original imbalanced dataset, ResNet152V2 achieved an overall accuracy of 66.03%, with a macro-average precision of 69.02%, recall of 69.32%, and F1-score of 67.13%. While the model performed decently overall, it struggled with class imbalance, particularly for minority classes such as Nematode, which had only 8 samples in the test set, 48 in the training set, and 14 in the validation set. For Nematode, precision was 71.43%, but recall was lower at 62.50%, resulting in an F1-score of 66.67%, indicating the difficulty in detecting this underrepresented class. In contrast, Bacteria, the largest class with 569 original images and 1990 images after augmentation, had a high precision (100%) but a much lower recall (68.97%), leading to an F1-score of 81.63%. This suggests that while the model is highly accurate in classifying Bacteria, it may be overfitting to this class at the expense of minority classes. Similarly, the Healthy class showed a high recall of 95.24%, but its precision was quite low at 42.55%, resulting in an F1-score of 58.82%, indicating the model's difficulty in balancing between false positives and false negatives for this class. Other classes, such as Fungi and Pest, further reflected this imbalance, with Fungi achieving an F1-score of 58.28%, while Pest managed 56.69%.

Table 18
Class-wise accuracy for each experimental setting (NASNetMobile).

Class	Experimental setting	Accuracy
Bacteria	Original data	0.8621
	Augmented + Early Stop	0.7931
	Augmented + L2 Norm + Early Stop	0.8103
Fungi	Original data	0.6053
	Augmented + Early Stop	0.4605
	Augmented + L2 Norm + Early Stop	0.4868
Healthy	Original data	0.5714
	Augmented + Early Stop	0.5238
	Augmented + L2 Norm + Early Stop	0.4762
Nematode	Original data	0.7500
	Augmented + Early Stop	0.5000
	Augmented + L2 Norm + Early Stop	0.6250
Pest	Original data	0.5806
	Augmented + Early Stop	0.7097
	Augmented + L2 Norm + Early Stop	0.4355
Phytophthora	Original data	0.8056
	Augmented + Early Stop	0.8333
	Augmented + L2 Norm + Early Stop	0.7778
Virus	Original data	0.8519
	Augmented + Early Stop	0.8333
	Augmented + L2 Norm + Early Stop	0.8148

After applying augmentation techniques and early stopping, the performance of ResNet152V2 showed marginal improvements in some areas, although the overall accuracy decreased slightly to 64.13%. The Nematode class saw a decrease in precision to 50%, but its recall improved to 62.50%, resulting in an F1-score of 55.56%, indicating a more balanced detection for this class compared to the baseline. The Bacteria class showed a drop in precision to 90%, though its recall remained consistent at 77.59%, leading to a slightly reduced F1-score of 83.33%, reflecting better generalization across classes post-augmentation. Healthy saw an improvement in precision to 63.16%, although recall dropped to 57.14%, resulting in a more balanced F1-score of 60%, suggesting that augmentation helped improve detection across minority classes at the expense of a slight decline in overall accuracy. Fungi showed a modest improvement in performance, with its F1-score improving slightly to 57.93%, while Pest experienced a decline, with its F1-score dropping to 55.64%.

In the third experiment, where augmentation was combined with early stopping and L2 regularization, the overall accuracy remained similar at 64.76%, but the balance between precision and recall improved for several classes. Bacteria maintained a relatively high performance, with a precision of 78.18% and recall of 74.14%, resulting in an F1-score of 76.11%, indicating that the model became less biased toward this majority class. The Nematode class saw its F1-score fluctuate further to 50%, as both precision and recall balanced out

Table 19
Classification report for ResNet152V2 (Original, Augmented with Early Stopping, and Augmented with Early Stopping + L2 Regularization).

Class	Precision	Recall	F1-Score	Support
Original dataset				
Bacteria	1.000000	0.689655	0.816327	58
Fungi	0.586667	0.578947	0.582781	76
Healthy	0.425532	0.952381	0.588235	21
Nematode	0.714286	0.625000	0.666667	8
Pest	0.553846	0.580645	0.566929	62
Phytophthora	0.736842	0.777778	0.756757	36
Virus	0.813953	0.648148	0.721649	54
Accuracy		0.660317		
Macro avg	0.690161	0.693222	0.671335	315
Weighted avg	0.704938	0.660317	0.668846	315
Augmented dataset (Early Stopping)				
Bacteria	0.900000	0.775862	0.833333	58
Fungi	0.593220	0.460526	0.518519	76
Healthy	0.631579	0.571429	0.600000	21
Nematode	0.500000	0.625000	0.555556	8
Pest	0.521127	0.596774	0.556391	62
Phytophthora	0.590909	0.722222	0.650000	36
Virus	0.677419	0.777778	0.724138	54
Accuracy		0.641270		
Macro avg	0.630608	0.647084	0.633991	315
Weighted avg	0.649877	0.641270	0.640587	315
Augmented dataset (Early Stopping + L2 Regularization)				
Bacteria	0.781818	0.741379	0.761062	58
Fungi	0.608696	0.552632	0.579310	76
Healthy	0.866667	0.619048	0.722222	21
Nematode	0.500000	0.500000	0.500000	8
Pest	0.526316	0.483871	0.504202	62
Phytophthora	0.642857	0.750000	0.692308	36
Virus	0.652174	0.833333	0.731707	54
Accuracy		0.647619		
Macro avg	0.654075	0.640038	0.641544	315
Weighted avg	0.650153	0.647619	0.644545	315

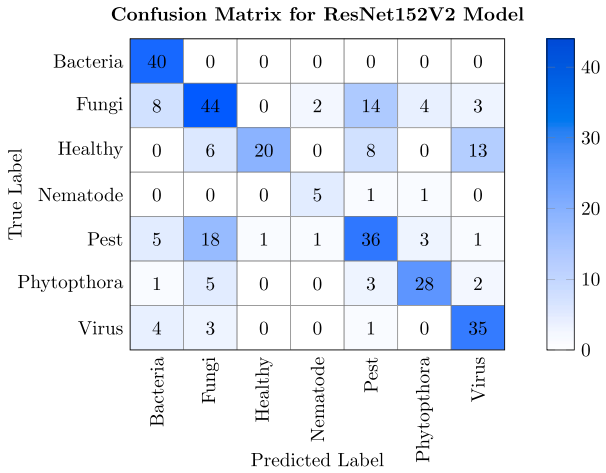


Fig. 22. Confusion matrix for ResNet152V2 (original dataset).

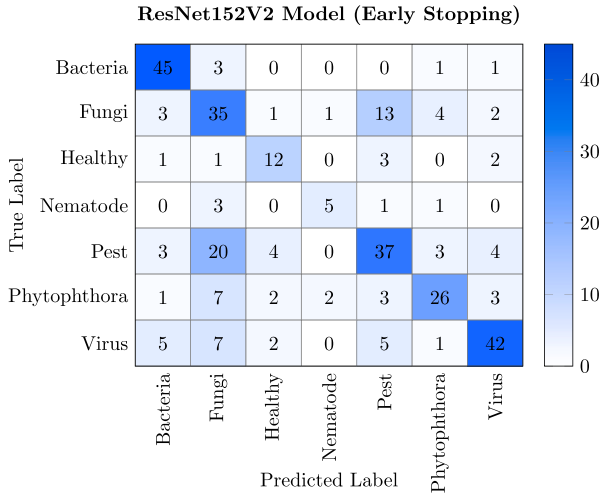


Fig. 23. Confusion matrix for ResNet152V2 (Augmented + early stopping).

at 50%. The Healthy class saw a notable improvement in precision, rising to 86.67%, though recall dropped to 61.90%, leading to an F1-score of 72.22%, the highest for this class across all experiments. Meanwhile, Fungi showed a slight decline in performance, with its F1-score decreasing slightly to 57.93%, indicating that L2 regularization helped stabilize its performance. Pest experienced consistent declines across all experiments, with its F1-score falling further to 50.42% after augmentation and regularization.

Overall, the macro-average F1-score fluctuated across experiments, starting at 67.13% in the baseline, slightly declining to 63.39% after augmentation, and finally stabilizing at 64.15% post-L2 regularization. This suggests that while L2 regularization helped reduce overfitting and balance precision and recall for some classes, minority classes such as Nematode and Pest continued to struggle with lower performance. The use of augmentation and L2 regularization helped balance the precision–recall trade-off for larger classes like Bacteria and Healthy, though certain minority classes continued to experience underperformance.

The confusion matrices for the ResNet152V2 model across different conditions are presented in Figs. 22, 23, and 24. In Fig. 22, which depicts the confusion matrix for the original dataset, the Bacteria class shows 49 correct predictions, with some misclassifications in the Fungi and Pest classes. The Fungi class performs well, with 48 correct predictions, but there are a few misclassifications in classes such as Healthy and Nematode. After applying early stopping in Fig. 23, the Bacteria class sees a slight drop to 45 correct predictions, while the Fungi class shows a slight decrease, with 35 correct classifications. However, there is a notable improvement in the Pest class, with 37 correct classifications compared to 35 in the original dataset. Moving to Fig. 24, where early stopping and L2 regularization are both applied, we observe further changes. The Bacteria class has 43 correct classifications, showing a minor decline from earlier stages, while the Fungi class shows a notable recovery to 42 correct predictions. The Healthy and Nematode classes see relatively consistent performance across the stages. The Pest class maintains its strong performance with 42 correct predictions, and the Phytophthora class also shows consistent improvement, reaching 30 correct classifications. Overall, the significant changes between Figs. 22 and 24 include improvement in the Pest and Phytophthora classes, while the Bacteria and Fungi classes see fluctuations in performance, possibly due to regularization impacts. The class-wise accuracies for each experimental setting are showcased in Table 20 which are calculated by taking the ratio of the true positives to the support values of that particular class.

4.1.3. Results for DenseNet201 (Testing data)

The detailed classification report of all three experiments utilizing DenseNet201 is showcased in Table 21. In the baseline performance evaluation on the original imbalanced dataset, DenseNet201 achieved an overall accuracy of 77.14%, with a macro-average precision of 77.14%, recall of 73.96%, and an F1-score of 75.20%. The model performed well for larger classes, such as Bacteria and Virus, but struggled with smaller classes like Nematode. For Bacteria, the model achieved a precision of 98.08% and a recall of 87.93%, indicating that

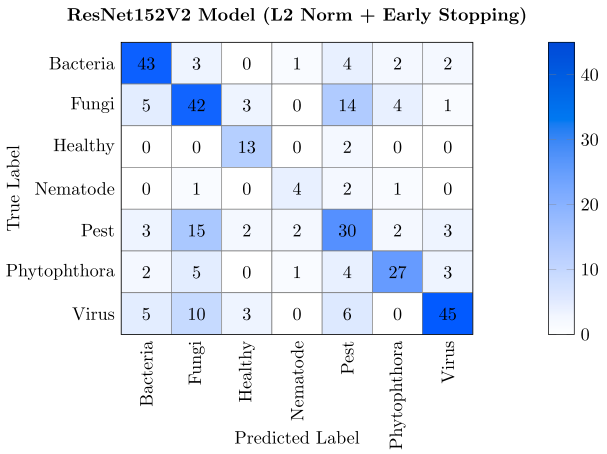


Fig. 24. Confusion matrix for ResNet152V2 (Augmented + L2 regularization + early stopping).

Table 20
Class-wise accuracy for each experimental setting (ResNet152V2).

Class	Experimental setting	Accuracy
Bacteria	Original data	0.8448
	Augmented + Early Stop	0.7759
	Augmented + L2 Norm + Early Stop	0.7414
Fungi	Original data	0.6316
	Augmented + Early Stop	0.4605
	Augmented + L2 Norm + Early Stop	0.5526
Healthy	Original data	0.6667
	Augmented + Early Stop	0.5714
	Augmented + L2 Norm + Early Stop	0.6190
Nematode	Original data	0.5000
	Augmented + Early Stop	0.6250
	Augmented + L2 Norm + Early Stop	0.7500
Pest	Original data	0.5645
	Augmented + Early Stop	0.5968
	Augmented + L2 Norm + Early Stop	0.4839
Phytophthora	Original data	0.8056
	Augmented + Early Stop	0.8333
	Augmented + L2 Norm + Early Stop	0.7778
Virus	Original data	0.8333
	Augmented + Early Stop	0.7778
	Augmented + L2 Norm + Early Stop	0.8333

DenseNet201 effectively identified this class, likely due to the large number of samples. Similarly, Virus exhibited balanced performance with both precision and recall at 85.19%. However, the Nematode class, which had only 8 samples in the test set and 48 in the training set, showed lower performance, with a precision of 66.67% and recall of 50%, leading to an F1-score of 57.14%. This discrepancy highlights the challenges the model faces in detecting under-represented classes.

After applying augmentation techniques and early stopping, DenseNet201's performance remained strong, with overall accuracy holding steady at 77.14%. The macro-average precision decreased slightly to 75.26%, but the recall improved to 79.14%, indicating better generalization across all classes. The Nematode class saw significant improvement, with precision increasing to 63.64% and recall improving dramatically to 87.50%, yielding a higher F1-score of 73.68%. For Bacteria, the precision dropped slightly to 91.23%, while recall remained high at 89.66%, maintaining a balanced detection with an F1-score of 90.43%. Fungi, a moderately represented class, showed an increase in precision to 69.12%, though its recall dropped slightly to 61.84%, resulting in an F1-score of 65.28%. Overall, the application of augmentation and early stopping helped improve generalization and reduce overfitting, particularly for under-represented classes like Nematode, which benefited the most from these techniques.

Table 21
Classification Report for DenseNet201 (Original, Augmented with Early Stopping, and Augmented with Early Stopping + L2 Regularization).

Class	Precision	Recall	F1-Score	Support
Original dataset				
Bacteria	0.980769	0.879310	0.927273	58
Fungi	0.622222	0.736842	0.674699	76
Healthy	0.750000	0.714286	0.731707	21
Nematode	0.666667	0.500000	0.571429	8
Pest	0.759259	0.661290	0.706897	62
Phytophthora	0.769231	0.833333	0.800000	36
Virus	0.851852	0.851852	0.851852	54
Accuracy			0.771429	
Macro avg	0.771429	0.739559	0.751979	315
Weighted avg	0.781026	0.771429	0.773409	315
Augmented dataset (Early Stopping)				
Bacteria	0.912281	0.896552	0.904348	58
Fungi	0.691176	0.618421	0.652778	76
Healthy	0.727273	0.761905	0.744186	21
Nematode	0.636364	0.875000	0.736842	8
Pest	0.728814	0.693548	0.710744	62
Phytophthora	0.750000	0.750000	0.750000	36
Virus	0.822581	0.944444	0.879310	54
Accuracy			0.771429	
Macro avg	0.752641	0.791410	0.768315	315
Weighted avg	0.769559	0.771429	0.768682	315
Augmented dataset (Early Stopping + L2 Regularization)				
Bacteria	0.887097	0.948276	0.916667	58
Fungi	0.662162	0.644737	0.653333	76
Healthy	0.708333	0.809524	0.755556	21
Nematode	0.777778	0.875000	0.823529	8
Pest	0.724138	0.677419	0.700000	62
Phytophthora	0.771429	0.750000	0.760563	36
Virus	0.867925	0.851852	0.859813	54
Accuracy			0.771429	
Macro avg	0.771266	0.793830	0.781352	315
Weighted avg	0.769553	0.771429	0.769794	315

In the third experiment, where augmentation was combined with early stopping and L2 regularization, DenseNet201 continued to perform well, with overall accuracy again at 77.14%. The Bacteria class saw slightly improved performance, with precision increasing to 88.71% and recall rising to 94.83%, resulting in an F1-score of 91.67%. This indicates that L2 regularization contributed to further balancing the model, especially for this highly represented class. Nematode also improved further, with precision increasing to 77.78% and recall remaining high at 87.50%, giving an F1-score of 82.35%. Other classes, such as Fungi and Healthy, saw consistent performance, with Fungi achieving a precision of 66.22% and recall of 64.47%, while Healthy improved with a precision of 70.83% and a recall of 80.95%, leading to an F1-score of 75.56%. The use of L2 regularization helped control overfitting and ensured more balanced precision and recall, particularly in larger and moderately represented classes.

Across all experimental settings, the Bacteria class maintained relatively high performance, with its F1-score decreasing slightly from 92.73% in the baseline to 91.67% after augmentation and L2 regularization. The Nematode class showed the most significant improvement, with its F1-score rising from 57.14% in the baseline to 82.35% after L2 regularization. The Healthy class also saw improvements, with its F1-score increasing from 73.17% in the baseline to 75.56% after augmentation and regularization. Fungi experienced consistent performance, with a modest improvement in F1-score from 67.47% in the baseline to 65.33% after augmentation and L2 regularization. However, Pest showed a slight decline, with its F1-score dropping from 70.69% in the baseline to 70.00% after regularization, despite maintaining consistent accuracy across experiments.

Overall, the macro-average F1-score remained stable across all experiments, starting at 75.20% in the baseline, then slightly increasing

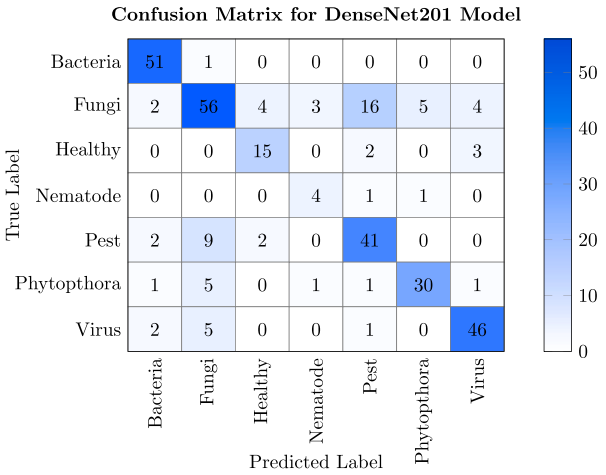


Fig. 25. Confusion matrix for DenseNet201 (original dataset).

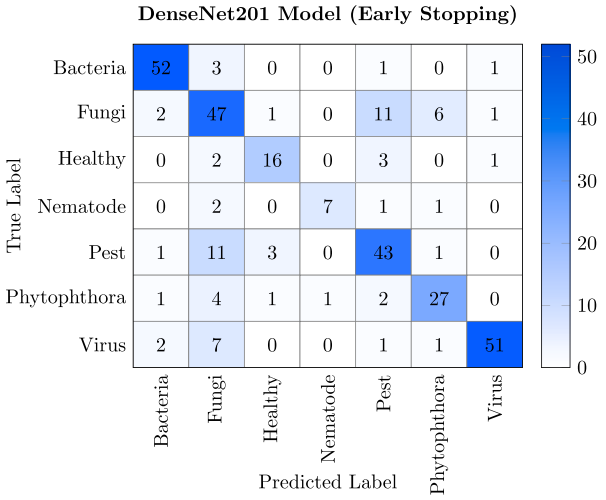


Fig. 26. Confusion matrix for DenseNet201 (Augmented + early stopping).

to 76.83% after augmentation, and finally stabilizing at 77.14% after L2 regularization. This demonstrates that DenseNet201 effectively maintained performance across all classes, with regularization and augmentation primarily benefiting the under-represented classes such as Nematode, while still preventing overfitting to the larger classes like Bacteria and Virus.

The confusion matrices for the DenseNet201 model across different conditions are shown in Figs. 25, 26, and 27. In Fig. 25, representing the original dataset, the model performs well for the Bacteria class, with 56 correct predictions. However, there is some confusion within the Fungi class, as 48 instances are correctly classified, but minor misclassifications exist in the Healthy and Pest classes. When early stopping is applied (Fig. 26), the Bacteria class slightly drops to 52 correct predictions, while Fungi stays consistent with 47 correct classifications. The Nematode class shows a noticeable improvement, with an increase in correctly classified instances (7), compared to the original dataset. There is also a slight improvement in the Pest class, which has 43 correct classifications in Fig. 26. In Fig. 27, where L2 regularization and early stopping are applied, the Bacteria class remains relatively stable with 55 correct classifications. The Fungi class improves slightly, reaching 49 correct classifications, while the Pest class continues to perform well with 42 correct predictions. The Healthy class remains consistent, showing only minor fluctuations across the three matrices. The main significant changes across the matrices are seen in the Bacteria and

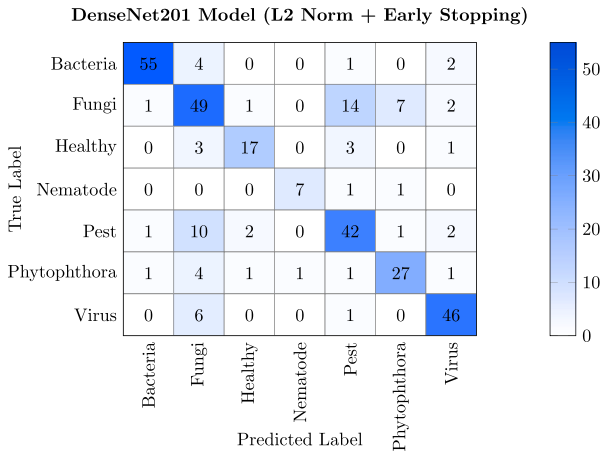


Fig. 27. Confusion matrix for DenseNet201 (Augmented + L2 regularization + early stopping).

Table 22
Class-wise accuracy for each experimental setting (DenseNet201).

Class	Experimental setting	Accuracy
Bacteria	Original data	0.9655
	Augmented + Early Stop	0.8966
	Augmented + L2 Norm + Early Stop	0.9483
Fungi	Original data	0.6316
	Augmented + Early Stop	0.6184
	Augmented + L2 Norm + Early Stop	0.6447
Healthy	Original data	0.7619
	Augmented + Early Stop	0.7619
	Augmented + L2 Norm + Early Stop	0.8095
Nematode	Original data	0.7500
	Augmented + Early Stop	0.8750
	Augmented + L2 Norm + Early Stop	0.8750
Pest	Original data	0.7258
	Augmented + Early Stop	0.6935
	Augmented + L2 Norm + Early Stop	0.6774
Phytophthora	Original data	0.8056
	Augmented + Early Stop	0.8333
	Augmented + L2 Norm + Early Stop	0.8333
Virus	Original data	0.9444
	Augmented + Early Stop	0.9444
	Augmented + L2 Norm + Early Stop	0.8519

Pest classes, where performance slightly fluctuates, but remains strong overall, while Fungi also shows consistent and gradual improvement. Overall, the combination of early stopping and L2 regularization brings subtle improvements in classification accuracy for the key classes. The class-wise accuracies for each experimental setting are showcased in Table 22 which are calculated by taking the ratio of the true positives to the support values of that particular class.

4.2. Analysis of overfitting and model selection

This section delves into the behavior of each model regarding overfitting and its generalization capability. We analyze how the models NasNetMobile, ResNet152V2, and DenseNet201 perform under different configurations: original dataset, augmented dataset with early stopping, and augmented dataset with both early stopping and L2 regularization. The goal of this analysis is to understand which models are more susceptible to overfitting and how regularization techniques help mitigate this issue, leading to optimal model selection for the task.

4.2.1. NasNetMobile

NasNetMobile, in its baseline configuration on the original dataset, demonstrates clear signs of overfitting. From the training and validation

loss curves (Figs. 16 and 18), it is evident that the model quickly reduces its training loss, stabilizing at lower values, while the validation loss declines at a much slower pace. This gap between the two loss curves suggests that NasNetMobile learns the training data relatively well but struggles to generalize to unseen data, particularly due to the class imbalance in the original dataset. In the original dataset configuration, NasNetMobile reaches a training accuracy of approximately 82.13% by epoch 20, yet the validation accuracy lags behind at 67.97%. This gap highlights the model's inclination toward overfitting, where it performs better on the training data than on the validation set.

When data augmentation and early stopping are applied, NasNetMobile's performance improves marginally. Training loss declines more steadily, and the validation loss curve begins to stabilize earlier. However, even with these adjustments, NasNetMobile remains prone to overfitting to a certain degree. Although the validation accuracy reaches 69.12%, the test loss (1.0541) is relatively high, indicating that the model continues to face challenges in generalizing beyond the training data. Notably, the inclusion of augmented data does help the model better handle minority classes, as seen by the improved precision for classes like Nematode and Healthy. The introduction of L2 regularization alongside data augmentation and early stopping offers additional control over overfitting, yet the validation loss still remains higher compared to other models like DenseNet201. NasNetMobile's final test loss (2.3655) in this configuration is the highest among the three models, reinforcing that while L2 regularization aids in penalizing large weights and preventing overfitting, NasNetMobile still struggles to achieve a balance between precision and recall, particularly for underrepresented classes. In terms of model selection, NasNetMobile, while useful in certain scenarios, may not be the ideal choice for this dataset given its susceptibility to overfitting. Its generalization is notably weaker than the other two models, and even with regularization, its performance on the test set lags behind. Therefore, it should be selected cautiously, particularly when handling imbalanced datasets without further advanced techniques like GAN-based augmentation.

4.2.2. ResNet152V2

ResNet152V2, like NasNetMobile, shows signs of overfitting in its baseline configuration on the original dataset. The training accuracy reaches 76.8% by epoch 20, but the validation accuracy falls behind, peaking at 66.03%. The disparity between training and validation loss curves (Figs. 16 and 18) highlights ResNet152V2's tendency to memorize training data without learning generalizable patterns for the test set. This overfitting is particularly noticeable in the higher validation loss (1.3490) compared to the training loss, suggesting the model's inefficacy in managing the class imbalance in the dataset. The addition of data augmentation and early stopping addresses overfitting to some extent. ResNet152V2 (Augmented + Early Stopping) shows improvement in validation accuracy, reaching 71.57% by epoch 12, with the training loss curve becoming more stable after initial epochs. However, despite these improvements, the model still exhibits overfitting tendencies. The test loss (1.0767) remains higher than DenseNet201, indicating that while ResNet152V2 benefits from the augmented data, it struggles to generalize effectively for all classes.

When L2 regularization is introduced alongside early stopping, the model's overfitting is further reduced. The validation accuracy stabilizes around 74.35%, and the training loss decreases more gradually, showing a more controlled learning process. However, the final test loss (2.2572) indicates that while the model benefits from L2 regularization, it may still not be fully equipped to handle the complexity and imbalance of the dataset. In terms of model selection, ResNet152V2 is a suitable choice when using augmentation and regularization techniques, as it demonstrates steady performance improvements. However, the model's susceptibility to overfitting without these techniques makes it less reliable on the original dataset. It may require additional techniques, such as more advanced augmentation or ensemble learning, to truly unlock its potential. In its current form, ResNet152V2 may be more appropriate when the dataset is sufficiently balanced or when overfitting is effectively controlled.

4.2.3. DenseNet201

DenseNet201 emerges as the most robust model in terms of both performance and resistance to overfitting across all configurations. On the original dataset, DenseNet201 achieves higher training accuracy (94.28% by epoch 15) compared to NasNetMobile and ResNet152V2, with the validation accuracy peaking at 77.14%. While the validation loss remains relatively low (0.8793), there is still a noticeable gap between training and validation loss, indicating some degree of overfitting. However, compared to the other models, DenseNet201 handles this better, particularly due to its superior architecture, which facilitates efficient feature reuse and reduces the risk of overfitting. When data augmentation and early stopping are applied, DenseNet201 exhibits significant improvements. The model quickly stabilizes at a high validation accuracy (75.98% by epoch 5), and the training loss reaches minimal levels early in the training process. This suggests that DenseNet201 benefits greatly from augmented data, allowing it to learn more generalizable patterns. The test loss of 0.6865 indicates that overfitting has been effectively controlled, and the model is able to perform consistently across both training and validation data.

The addition of L2 regularization further enhances DenseNet201's robustness against overfitting. The model maintains a stable validation accuracy (77.12%) while the training loss curve remains smooth, showing that L2 regularization helped control the model's complexity by penalizing large weights. The final test loss of 1.3507, while slightly higher than the augmented-only configuration, demonstrates that the model continues to generalize well even with regularization applied. In terms of model selection, DenseNet201 is clearly the best-performing model across all configurations. Its ability to handle both augmentation and regularization without significant loss in accuracy or increase in test loss makes it the ideal choice for this task. Furthermore, its consistent performance in handling the class imbalance and avoiding overfitting suggests that DenseNet201 is the most reliable option for practical applications, particularly in scenarios where generalization to unseen data is critical.

DenseNet201 emerges as the most suitable model for this task, given its ability to handle overfitting effectively across all configurations. NasNetMobile struggles with overfitting, even with regularization, making it a less ideal candidate unless further augmentation techniques are applied. ResNet152V2 performs reasonably well with augmentation and regularization but is prone to overfitting in its baseline configuration. Therefore, DenseNet201 should be the model of choice for further development and deployment, especially in real-world scenarios where data is imbalanced and generalization is essential.

4.3. K-Fold cross-validation

To ensure the robustness and reliability of the proposed DenseNet201 model, which demonstrated superior performance compared to other models, we applied 5-fold cross-validation with a stratified split. K-fold cross-validation is a well-established technique used to evaluate a model's performance by dividing the dataset into k equally sized subsets or "folds". In this study, we specifically focused on DenseNet201 for k-fold cross-validation because it was the only model among the three DenseNet201, ResNet152V2, and NasNetMobile that significantly outperformed previous studies in terms of accuracy. This additional validation was applied exclusively to DenseNet201 to confirm the robustness of its performance and to mitigate the potential influence of chance factors on the results. Stratified k-fold cross-validation was chosen to maintain the class distribution across all folds, which is crucial given the class imbalance in our dataset. By using a stratified split, each fold remains representative of the entire dataset, preserving the proportion of each class, which is essential for balanced evaluation. This method helps ensure that the evaluation is not biased by any particular subset of the data. The process began with the dataset being divided into 5 stratified folds using the 'StratifiedKFold' function

Table 23
5-Fold cross-validation results for DenseNet201.

Fold	Validation loss	Validation accuracy (%)
Fold 1	0.5938	78.90%
Fold 2	0.5092	82.44%
Fold 3	0.5398	81.30%
Fold 4	0.5189	81.46%
Fold 5	0.5270	82.44%
Average	**0.5378**	**81.31%**

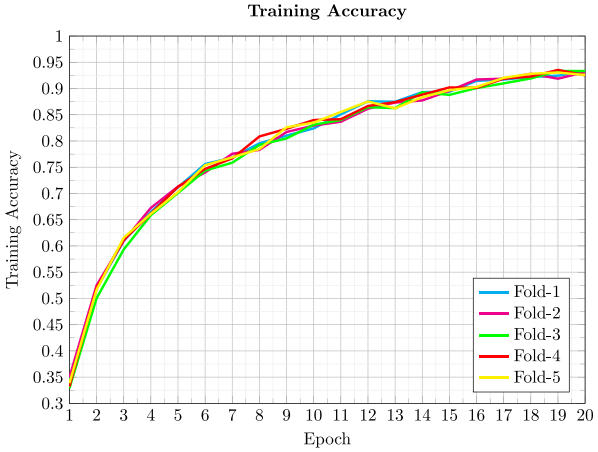


Fig. 28. Training accuracy graph for 5-fold cross-validation using the DenseNet201 architecture. This graph illustrates the training accuracy across each fold, demonstrating the model's learning performance and stability over different training subsets.

from the scikit-learn library. This division ensured that each fold maintained the class distribution, preventing any bias during model training and evaluation. For each fold, DenseNet201 was trained on 4 folds and validated on the remaining fold. This process was repeated for all 5 folds, ensuring that each fold served as the validation set exactly once. The model was trained using the previously specified hyperparameters: 20 epochs, a batch size of 16, and the Adam optimizer with a learning rate of 0.00001. After training on each fold, the model's performance was evaluated using validation loss and accuracy metrics. These metrics were recorded for each fold, and the results were averaged to provide a comprehensive assessment of DenseNet201's performance. The final evaluation was obtained by averaging the validation loss and accuracy across all 5 folds, reducing the likelihood of chance factors influencing the results and providing a more reliable estimate of the model's generalizability. The results of the 5-fold cross-validation for DenseNet201 are summarized in Table 23. The table presents the validation loss and accuracy for each fold, along with the average validation loss and accuracy across all folds. The application of 5-fold cross-validation confirmed the robustness of DenseNet201, with an average validation accuracy of 81.31% and an average validation loss of 0.5378 across the folds. This consistent performance across different subsets of the dataset underscores DenseNet201's effectiveness and reliability, further validating its superiority compared to previous studies. Figs. 28 and 29 illustrate the training accuracy and loss curves for the 5-folds, respectively, while Figs. 30 and 31 depict the corresponding validation accuracy and loss curves. These graphs provide a visual confirmation of the model's consistent performance across the folds.

4.4. Comparative study

This section analyzes the improvement in accuracy by comparing the results obtained in our study with those from previous work. The research paper used for comparison, authored by Shabrina et al., (2024), applied five different models to the same Potato Leaf Disease Dataset

Fig. 29. Training loss graph for 5-fold cross-validation using the DenseNet201 architecture. This graph shows the training loss for each fold, indicating the model's convergence behavior and helping to identify any overfitting trends across different training data splits.

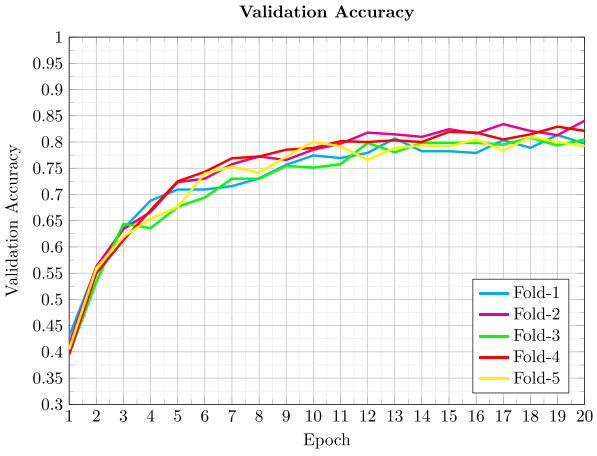


Fig. 30. Validation accuracy graph for 5-fold cross-validation using the DenseNet201 architecture. This graph presents the model's accuracy on validation sets across each fold, reflecting the model's generalization performance and robustness on unseen data.

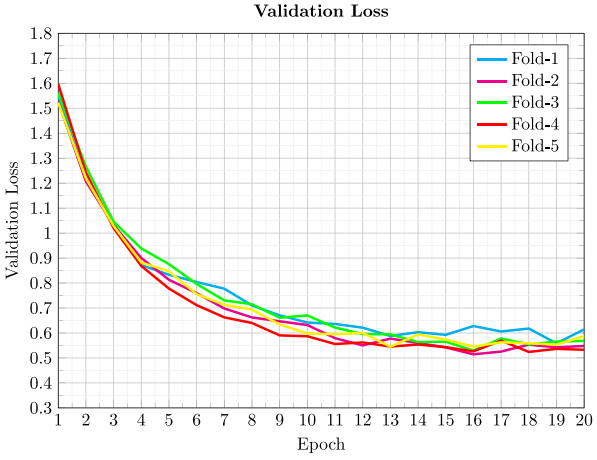


Fig. 31. Validation loss graph for 5-fold cross-validation using the DenseNet201 architecture. The validation loss across folds is displayed here, providing insights into the model's consistency on validation data and the effectiveness of training in minimizing error over multiple folds.

Table 24
Accuracy of different models on potato leaf disease dataset.

Model name	Accuracy (%)
EfficientNetV2B3 (Shabrina et al., 2024)	73.63
MobileNetV3-Large (Shabrina et al., 2024)	72.03
VGG-16 (Shabrina et al., 2024)	59.81
ResNet50 (Shabrina et al., 2024)	68.17
DenseNet121 (Shabrina et al., 2024)	59.16
NasNetMobile (Original) (Proposed)	69.21
ResNet152V2 (Original) (Proposed)	66.03
DenseNet201 (Original) (Proposed)	77.14
NasNetMobile (Augmented + Early Stopping) (Proposed)	68.25
ResNet152V2 (Augmented + Early Stopping) (Proposed)	64.13
DenseNet201 (Augmented + Early Stopping) (Proposed)	77.14
NasNetMobile (Augmented + L2 norm + Early Stopping) (Proposed)	62.86
ResNet152V2 (Augmented + L2 norm + Early Stopping) (Proposed)	64.76
DenseNet201 (Augmented + L2 norm + Early Stopping) (Proposed)	77.14
DenseNet201 K-fold cross-validation (stratified split) (Proposed)	**81.31**

in an Uncontrolled Environment (Shabrina et al., 2023). These models included EfficientNetV2B3, MobileNetV3-Large, VGG16, ResNet50, and DenseNet121. The highest accuracy reported in their study was 73.63%, achieved by the EfficientNetV2B3 model. In our research, we applied three distinct transfer learning models: NasNetMobile, ResNet152V2, and DenseNet201, on the same dataset. DenseNet201 initially achieved the highest accuracy of 77.14%, an improvement of 3.51% over the best results from the previous study. We further enhanced our models by applying early stopping and L2 regularization to mitigate overfitting and improve generalization. Specifically, DenseNet201 maintained its accuracy of 77.14% after early stopping, demonstrating its robustness. To evaluate the impact of additional regularization techniques, we applied L2 norm and early stopping to NasNetMobile, ResNet152V2, and DenseNet201. While NasNetMobile and ResNet152V2 exhibited slight variations in accuracy, DenseNet201's performance remained steady at 77.14%, reflecting its consistency across different training conditions. Finally, to ensure the robustness of DenseNet201, we applied 5-fold cross-validation (stratified split). This approach yielded an even higher average accuracy of 81.31%, representing an overall improvement of 7.68% over the previously reported best accuracy of 73.63%. This demonstrates that DenseNet201 not only surpasses other models in accuracy but also maintains consistent performance across different dataset subsets, confirming its reliability. The Table 24 summarizes the accuracy results of the different models from both the previous study and our work, highlighting the superior performance of DenseNet201 in multiple configurations.

4.5. Cross-dataset robustness validation

In real-world agricultural datasets, class imbalance is a common issue that can lead to overfitting and memorization of patterns when classification models are directly applied to smaller datasets. To address this challenge, strategic data augmentation, early stopping, and L2 regularization were employed to improve model generalization and prevent overfitting. While the proposed methodology has shown promising results on the original Potato Leaf Disease Dataset (Shabrina et al., 2023), it is essential to validate its robustness across different datasets. In this section, we test our approach on two additional datasets — the Modified PlantVillage dataset and the Modified Rose Leaf Disease dataset — to verify its effectiveness and generalizability. These two datasets were not originally highly imbalanced but have been made artificially imbalanced by removing images of certain classes to better test the approach mentioned in this research work. This cross-dataset validation was performed using the DenseNet201 model, which has proven to be the best choice in terms of accuracy, performance, and its ability to handle overfitting effectively. The dataset was split in the same ratio of 70/20/10 and the same setting of hyperparameters was used. The augmentations applied were only on the training data and not on validation and testing data.

Table 25
Image distribution across disease categories for original and augmented dataset for Modified PlantVillage dataset.

Category	Original images	Augmented images (Training data 70% split)
Potato early blight	700	1400
Potato late blight	140	1400
Potato healthy	106	1484
Total	**946**	**4284**

4.5.1. PlantVillage modified dataset

The PlantVillage dataset (Rex, 2019) is widely used for plant disease classification tasks and serves as a reliable benchmark for testing the performance of deep learning models in agriculture. In this research, we used a modified version of the PlantVillage dataset, specifically focusing on potato leaf diseases, including Early Blight, Late Blight, and Healthy categories. The original dataset contained 1000 images each for Early Blight and Late Blight, and 152 images for Healthy potato leaves. To create an imbalanced class distribution for this study, we randomly removed images from the Late Blight category using the random.sample() function from Python's random library, which selects a specified number of images randomly from the dataset without replacement, reducing it to 200 images, while retaining 1000 images of Early Blight and 152 images of Healthy leaves, resulting in a total of 1352 images. This imbalance was introduced deliberately to test and validate the methodology proposed in this research, which includes strategic augmentations, early stopping, and L2 regularization to handle imbalanced datasets and improve model generalization.

The augmentations applied were carefully chosen to simulate real-life variability in plant images, such as changes in brightness, contrast, and minor distortions like noise and rotation etc. These augmentations aimed to address the class imbalance while also mimicking real-world conditions that plants might be photographed under, such as different lighting, camera angles, and environmental conditions. The strategic augmentations helped the model learn more robust features, particularly in underrepresented classes, ensuring better generalization on unseen data. The image distribution for both the original and augmented datasets is represented in Table 25, while Table 26 showcases the strategic augmentations applied to each class to enhance model performance. Fig. 32 illustrates a sample image from each class along with the augmentations applied to those images. The resulting augmented dataset expanded to 4284 images, which was split into training (70%), validation (20%), and testing (10%) sets. By validating the model on this modified dataset, we aim to demonstrate its cross-dataset robustness and its ability to generalize effectively to new, unseen data.

The performance of the fine-tuned DenseNet201 model on the original and augmented versions of the modified PlantVillage dataset highlights the effectiveness of the proposed methodology — incorporating strategic augmentations, early stopping, and L2 regularization — particularly in addressing class imbalance. This approach significantly enhanced the model's ability to generalize, especially for underrepresented classes such as "Potato Late Blight" and "Potato Healthy", which were previously more difficult for the model to classify due to the imbalanced nature of the dataset.

Table 26
Augmentation techniques applied to different classes of potato diseases.

Class	Count	Augmentations
Potato healthy	13	rotation_range, zoom_range, horizontal_flip, brightness, color_jitter, contrast, small_noise, rotation_range + brightness, horizontal_flip + color_jitter, zoom_range + contrast, brightness + small_noise, horizontal_flip + small_noise, color_jitter + contrast
Potato Early blight	1	rotation_range
Potato Late blight	9	rotation_range, zoom_range, color_jitter, horizontal flip, brightness, small_noise, brightness + small_noise, horizontal flip + small_noise

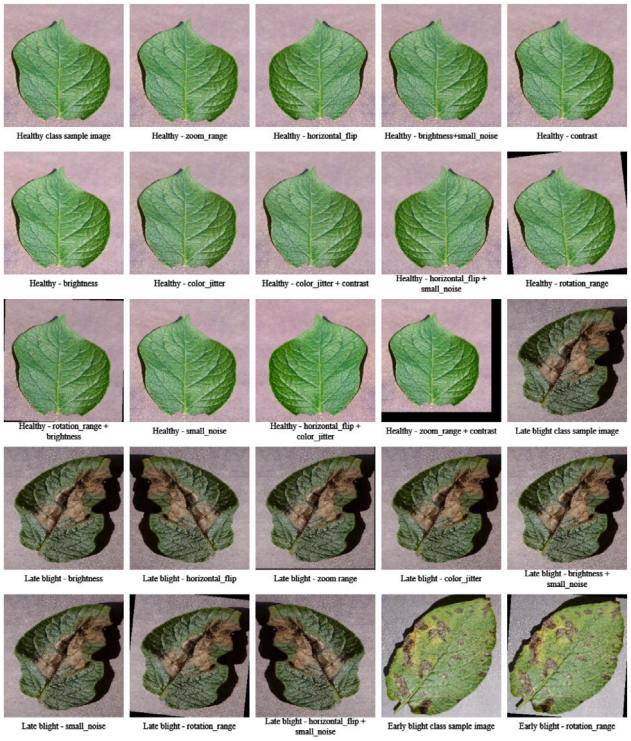


Fig. 32. Image of dataset distributed among various classes after augmentation.

Initially, when tested on the original dataset, DenseNet201 already performed well, achieving an accuracy of 98.53%. However, a closer look at the underrepresented classes reveals some limitations. For "Potato Late Blight", the model reached a high precision of 1.0000 but showed a recall of 0.9000, indicating it missed some true positives. Similarly, the "Potato Healthy" class achieved perfect recall (1.0000) but exhibited a lower precision (0.8889), leading to a reduced F1-score. These metrics suggest that while the model performed conservatively, it struggled to generalize well for minority classes, a typical issue in imbalanced datasets.

The introduction of data augmentation and early stopping in the second experimental setting addressed these concerns. The model's performance improved across all metrics, with accuracy rising to 99.26%. For "Potato Late Blight", recall improved to 0.9500, reducing the number of missed instances while maintaining a perfect precision of 1.0000. The F1-score for this class increased to 0.9744. Similarly, the "Potato Healthy" class saw improvements in precision, rising to 0.9412, while recall remained at 1.0000. These results demonstrate how augmentations, which introduce variability mimicking real-world conditions, helped the model learn from a broader range of input scenarios, particularly benefitting the minority classes.

Table 27
Performance evaluation of fine-tuned DenseNet201 on Modified PlantVillage Dataset (Original, Augmented with Early Stopping, and Augmented with Early Stopping + L2 Regularization).

Dataset type	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
Original dataset	96.30	96.67	96.28	98.53
Augmented dataset (Early Stopping)	98.04	98.33	98.14	99.26
Augmented dataset (Early Stopping + L2 Regularization)	100.00	100.00	100.00	100.00

In the third experimental setting, where both early stopping and L2 regularization were applied, the model achieved a perfect 100% accuracy across all metrics, including precision, recall, and F1-score for each class. This was especially significant for the underrepresented classes, where "Potato Late Blight" and "Potato Healthy" both achieved perfect precision, recall, and F1-scores (1.0000). The application of L2 regularization was instrumental in reducing overfitting, as evidenced by the reduced validation loss and stabilized validation accuracy. By penalizing model complexity, L2 regularization helped the model generalize better to unseen data, ensuring that the performance gains from augmentation and early stopping were fully realized.

These findings validate the research methodology, which effectively addresses class imbalance, reduces overfitting, and improves generalization. The model's performance on the modified PlantVillage dataset confirms its robustness and applicability to similar datasets in the agricultural domain, where class imbalance and real-world data variability are common challenges. The improvement in metrics across all experimental settings, especially for underrepresented classes, highlights the importance of the proposed methodology in handling imbalanced datasets. The strategic augmentations simulated real-life conditions, and the combination of early stopping and L2 regularization effectively mitigated overfitting, ensuring high precision and recall across all classes.

In conclusion, the methodology effectively improves the model's ability to handle class imbalance through strategic augmentations, early stopping, and L2 regularization. These techniques not only boosted overall performance but also ensured that minority classes like "Potato Late Blight" and "Potato Healthy" achieved perfect classification metrics. This confirms that the proposed approach is robust, generalizable, and well-suited for real-world agricultural datasets, where the challenges of data imbalance and variability are prevalent (see Figs. 33–36 and Table 28).

4.5.2. Rose leaf disease modified dataset

The Rose leaf disease dataset (Rajbongshi, Sazzad, Shakil, Akter, and Kaiser, 2022) was used in this study for cross-data validation due to the similarity of diseased leaf appearances on plants from different species. This made it an ideal candidate for validating the generalization capability of the fine-tuned DenseNet201 model. The original dataset contains a total of 917 images, spread across three classes: Black Spot, Downy Mildew, and Fresh Leaf. Specifically, the dataset included 313 images of Black Spot, 200 images of Downy Mildew, and 404 images of Fresh Leaf. To evaluate the robustness of the proposed methodology, we modified this dataset to introduce artificial class imbalance. In this modified dataset, we retained the 313 images for Black Spot but reduced the Downy Mildew class to 75 images, creating an imbalance, while keeping the Fresh Leaf category unchanged at 404 images. The reduction in the Downy Mildew class was achieved using the 'random.sample()' function from Python's random library, which selects a specified number of images randomly from the dataset without replacement. This deliberate introduction of class imbalance was designed to test our methodology's ability to handle such scenarios, improving model generalization through strategic augmentations, early stopping, and L2 regularization.

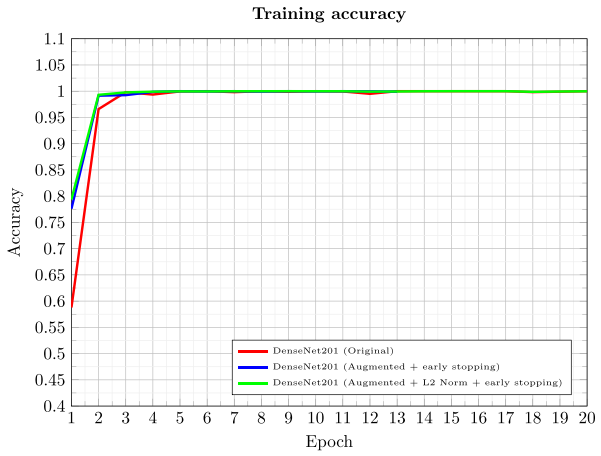


Fig. 33. Training accuracy for the original and augmented PlantVillage dataset using the DenseNet201 architecture. This graph compares the training accuracy across three settings: DenseNet201 (Original) indicated by the red line, DenseNet201 (Augmented + early stopping) by the blue line, and DenseNet201 (Augmented + L2 Norm + early stopping) by the green line. The comparison highlights how augmentation and regularization techniques impact model learning across different dataset conditions.

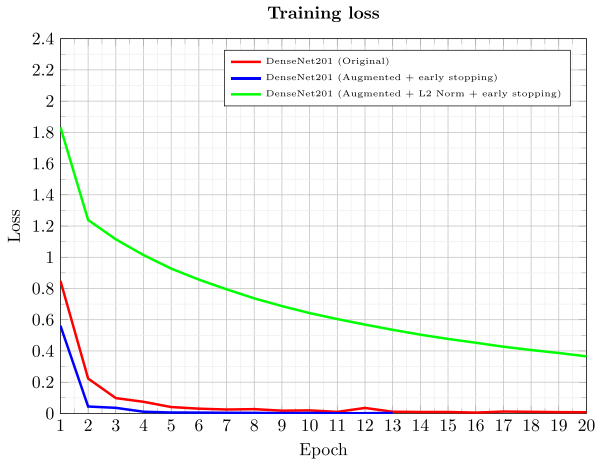


Fig. 34. Training loss for the original and augmented PlantVillage dataset using the DenseNet201 architecture. This plot shows the training loss across three settings: DenseNet201 (Original) in red, DenseNet201 (Augmented + early stopping) in blue, and DenseNet201 (Augmented + L2 Norm + early stopping) in green. The graph demonstrates the effect of augmentation and L2 regularization on model convergence and potential overfitting, with each setting revealing different loss patterns.

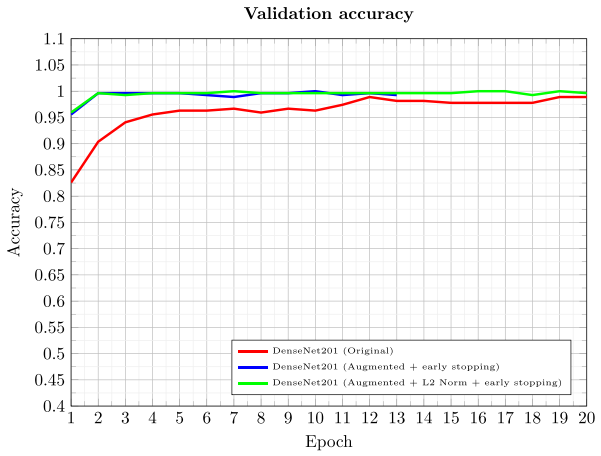


Fig. 35. Validation accuracy for the original and augmented PlantVillage dataset using the DenseNet201 architecture. This graph compares validation accuracy for three configurations: DenseNet201 (Original) in red, DenseNet201 (Augmented + early stopping) in blue, and DenseNet201 (Augmented + L2 Norm + early stopping) in green. The results provide insights into the model's generalization ability across different dataset settings, showing how augmentation and regularization contribute to performance consistency.

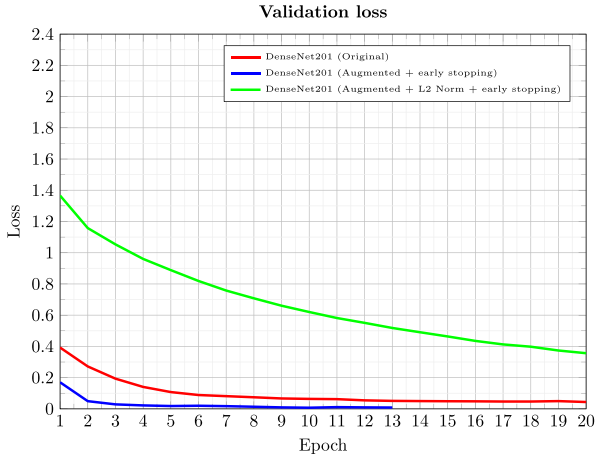


Fig. 36. Validation loss for the original and augmented PlantVillage dataset using the DenseNet201 architecture. The plot presents validation loss across three configurations: DenseNet201 (Original) in red, DenseNet201 (Augmented + early stopping) in blue, and DenseNet201 (Augmented + L2 Norm + early stopping) in green. This comparison highlights the model's robustness on unseen data, with the effects of early stopping and L2 regularization reflected in the validation loss across different dataset variations.

Table 29 presents the data distribution for both the original modified dataset and the augmented dataset, following the application of strategic augmentations. These augmentations were designed to simulate real-world variability, addressing the class imbalance while also enhancing the model's ability to generalize. The types of augmentations applied to each class are shown in Table 30, which include techniques like rotation, zoom, brightness, and color jitter, among others, to ensure diverse image representation. Additionally, Fig. 37 showcases sample images from each class, along with the images obtained after applying the augmentations. These augmented images were critical in enabling the model to learn robust features across different classes, improving its performance in real-world settings where variability in lighting, angles, and environmental conditions is common.

The performance evaluation of the fine-tuned DenseNet201 model on the Rose leaf disease dataset, as shown in Table 31, demonstrates clear improvements across the different experimental settings. Initially, when evaluated on the original dataset, the model achieved a respectable accuracy of 95.12%, with precision, recall, and F1-scores at 96.65%, 92.96%, and 94.64%, respectively. However, despite these high values, the model struggled with the class imbalance present in the dataset, particularly for the "Downy Mildew" class, which had a smaller number of images compared to the other classes.

With the application of data augmentation and early stopping in the second experimental setting, the model's performance saw a significant boost, achieving a perfect accuracy of 100% across all metrics. The strategic augmentations increased the variability within the training data, allowing the model to learn more robust features. Early stopping prevented the model from overfitting during training, leading to better generalization on the validation and test datasets. This improvement is notable because it shows how augmentations that simulate real-world variability, such as changes in lighting and angle, allow the model to generalize more effectively across all classes.

Finally, in the third setting, where both early stopping and L2 regularization were applied, the model once again achieved perfect performance across all metrics, with precision, recall, and F1-scores all

Table 28
Classification report for DenseNet201 on Modified PlantVillage Dataset (Original, Augmented with Early Stopping, and Augmented with Early Stopping + L2 Regularization).

Class	Precision	Recall	F1-Score	Support
Original dataset				
Potato Early blight	1.0000	1.0000	1.0000	100
Potato Late blight	1.0000	0.9000	0.9474	20
Potato healthy	0.8889	1.0000	0.9412	16
Accuracy			0.9853	
Macro avg	0.9630	0.9667	0.9628	136
Weighted avg	0.9869	0.9853	0.9853	136
Augmented dataset (Early Stopping)				
Potato Early blight	1.0000	1.0000	1.0000	100
Potato Late blight	1.0000	0.9500	0.9744	20
Potato healthy	0.9412	1.0000	0.9697	16
Accuracy			0.9926	
Macro avg	0.9804	0.9833	0.9814	136
Weighted avg	0.9931	0.9926	0.9927	136
Augmented dataset (Early Stopping + L2 Regularization)				
Potato Early blight	1.0000	1.0000	1.0000	100
Potato Late blight	1.0000	1.0000	1.0000	20
Potato healthy	1.0000	1.0000	1.0000	16
Accuracy			1.0000	
Macro avg	1.0000	1.0000	1.0000	136
Weighted avg	1.0000	1.0000	1.0000	136

Table 29
Image distribution across disease categories for original and Augmented dataset for Rose leaf disease dataset.

Category	Original images	Augmented images (Training data 70% split)
Downy Mildew	75	520
Black Spot	313	438
Fresh Leaf	404	564
Total	**792**	**1522**

Table 30
Augmentation techniques applied to different classes of rose diseases.

Class	Count	Augmentations
Black_Spot	1	rotation_range
Downy_Mildew	9	rotation_range, zoom_range, horizontal_flip, brightness, rotation_range + brightness, horizontal_flip + color_jitter, zoom_range + contrast, brightness + small_noise, color_jitter + contrast
Fresh_Leaf	1	rotation_range

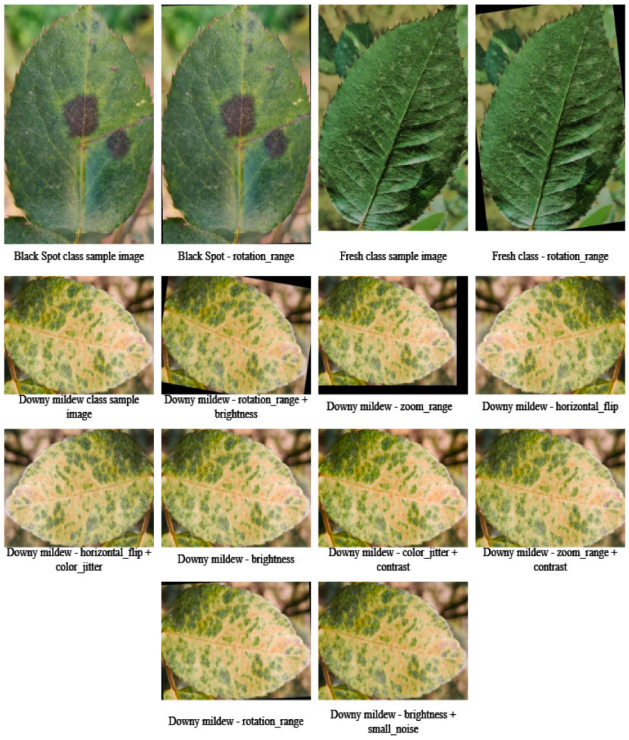


Fig. 37. Image of dataset distributed among various classes after augmentation.

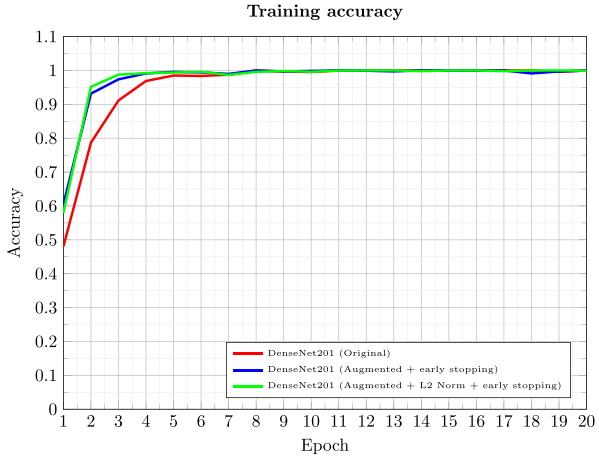


Fig. 38. Training accuracy for the original and augmented Rose leaf disease dataset using the DenseNet201 architecture. This graph illustrates the training accuracy across three configurations: DenseNet201 (Original) indicated by the red line, DenseNet201 (Augmented + early stopping) by the blue line, and DenseNet201 (Augmented + L2 Norm + early stopping) by the green line. The comparison highlights the impact of augmentation and regularization techniques on model learning with different dataset variations.

remaining at 100%. The introduction of L2 regularization was crucial in further reducing overfitting, as evidenced by the smoothness and stabilization seen in the graphical representations of training accuracy and loss over time, as shown in Figs. 38, 39, 40, and 41. These graphs provide clear evidence of the model's learning progression, with noticeable reductions in overfitting, thanks to the combination of augmentation, early stopping, and regularization. The smooth trends seen in the training and validation curves highlight how regularization helped control the model's complexity, ensuring it did not memorize the training data but learned patterns that generalized well to unseen data.

Table 32 provides a detailed classification report for each experimental setting, further confirming the model's overall improvement. In the original dataset configuration, the model struggled more with the underrepresented "Downy Mildew" class, achieving a precision of 1.0000 but a recall of 0.8750, leading to an F1-score of 0.9333. This was a clear indication of the class imbalance issue, as the model missed several instances of this class, failing to generalize adequately to the minority class.

However, once data augmentation and early stopping were applied, the performance for "Downy Mildew" improved dramatically, with both precision and recall reaching 1.0000, resulting in a perfect F1-score. This shows that the strategic augmentations effectively addressed the imbalance by providing the model with more diverse examples, allowing it to better distinguish between classes. Similarly, for the "Black Spot" and "Fresh Leaf" classes, which already had a larger number of images, the performance also improved to 100% in precision, recall, and F1-scores, indicating that the augmentations not only helped the

Table 31
Performance evaluation of fine-tuned DenseNet201 on Rose leaf disease dataset.

Dataset type	Precision (%)	Recall (%)	F1-Score (%)	Accuracy (%)
Original data	96.65	92.96	94.64	95.12
Augmented data (Early Stopping)	100.00	100.00	100.00	100.00
Augmented data (Early Stopping + L2 Regularization)	100.00	100.00	100.00	100.00

Fig. 39. Training loss for the original and augmented Rose leaf disease dataset using the DenseNet201 architecture. This plot shows the training loss across three settings: DenseNet201 (Original) in red, DenseNet201 (Augmented + early stopping) in blue, and DenseNet201 (Augmented + L2 Norm + early stopping) in green. The graph demonstrates how augmentation and L2 regularization affect model convergence and control overfitting across different dataset settings.

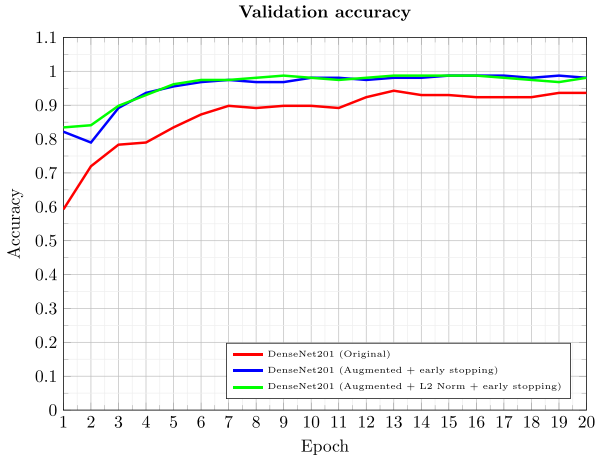


Fig. 40. Validation accuracy for the original and augmented Rose leaf disease dataset using the DenseNet201 architecture. This graph compares validation accuracy across three configurations: DenseNet201 (Original) in red, DenseNet201 (Augmented + early stopping) in blue, and DenseNet201 (Augmented + L2 Norm + early stopping) in green. The results provide insight into the model's generalization capacity, showing how augmentation and regularization contribute to consistent performance across different dataset conditions.

underrepresented class but also enhanced the model's overall ability to generalize.

In the final setting, where both L2 regularization and early stopping were employed, the model maintained perfect performance across all classes, further solidifying the effectiveness of the proposed methodology. L2 regularization played a critical role in penalizing large weight values, preventing the model from becoming too complex, while early stopping ensured that training was halted at the optimal point before overfitting could occur. The validation loss and accuracy curves (Figs. 40 and 41) provide additional graphical evidence of this, showing how the model's performance remained stable across training and validation datasets.

In conclusion, the combination of strategic augmentations, early stopping, and L2 regularization not only addressed the class imbalance issue effectively but also significantly improved the overall performance of the DenseNet201 model across all classes. The augmented dataset provided the model with more diverse training data, helping it to generalize better to minority classes like "Downy Mildew", while regularization techniques reduced overfitting and ensured stable performance. This study highlights the importance of using augmentation and regularization techniques in imbalanced datasets to achieve high performance and robust generalization.

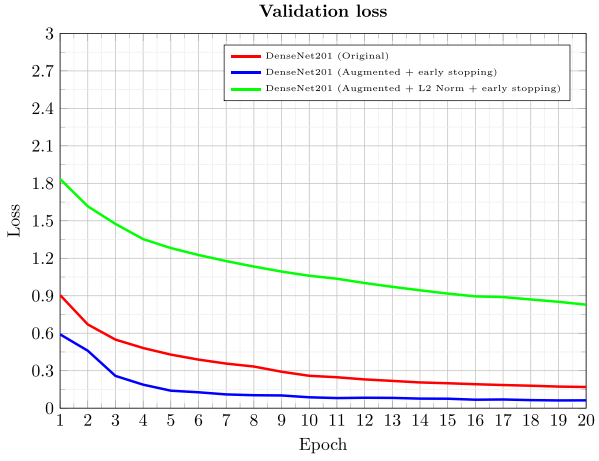


Fig. 41. Validation loss for the original and augmented Rose leaf disease dataset using the DenseNet201 architecture. The plot presents validation loss across three settings: DenseNet201 (Original) in red, DenseNet201 (Augmented + early stopping) in blue, and DenseNet201 (Augmented + L2 Norm + early stopping) in green. This comparison highlights the model's robustness on unseen data, with the effects of early stopping and L2 regularization visible in the validation loss across the different dataset variations.

4.6. Discussion

In this study, we aimed to enhance the accuracy and robustness of potato leaf disease classification using convolutional neural networks (CNNs) by addressing common challenges such as class imbalance and overfitting. We evaluated three deep learning models — NasNetMobile, ResNet152V2, and DenseNet201 — across different experimental settings involving data augmentation, early stopping, and L2 regularization. Our findings demonstrate that strategic data augmentation and regularization techniques significantly improve model performance, particularly in handling imbalanced datasets, and enhance the models' ability to generalize to new, unseen data. Initially, we trained and tested the models on the original imbalanced dataset without any augmentation or regularization. DenseNet201 achieved the highest accuracy of 77.14% (Table 16), outperforming NasNetMobile and ResNet152V2, which achieved accuracies of 69.21% and 66.03%, respectively. DenseNet201's superior performance can be attributed to its unique architecture characterized by dense connectivity, which promotes feature reuse and alleviates the vanishing gradient problem. This enables DenseNet201 to learn more complex and nuanced features essential for distinguishing between the seven classes of potato leaf

Table 32
Classification report for modified rose leaf disease dataset (Original, Augmented with Early Stopping, and Augmented with Early Stopping + L2 Regularization).

Class	Precision	Recall	F1-Score	Support
Original dataset				
Black Spot	0.967742	0.937500	0.952381	32
Downy Mildew	1.000000	0.875000	0.933333	8
Fresh Leaf	0.931818	0.976190	0.953488	42
Accuracy			0.951220	
Macro avg	0.966520	0.929563	0.946401	82
Weighted avg	0.952489	0.951220	0.951090	82
Augmented dataset (Early Stopping)				
Black Spot	1.000000	1.000000	1.000000	32
Downy Mildew	1.000000	1.000000	1.000000	8
Fresh Leaf	1.000000	1.000000	1.000000	42
Accuracy			1.000000	
Macro avg	1.000000	1.000000	1.000000	82
Weighted avg	1.000000	1.000000	1.000000	82
Augmented dataset (Early Stopping + L2 Regularization)				
Black Spot	1.000000	1.000000	1.000000	32
Downy Mildew	1.000000	1.000000	1.000000	8
Fresh Leaf	1.000000	1.000000	1.000000	42
Accuracy			1.000000	
Macro avg	1.000000	1.000000	1.000000	82
Weighted avg	1.000000	1.000000	1.000000	82

diseases, particularly in an imbalanced dataset where minority classes are underrepresented.

NasNetMobile and ResNet152V2, while powerful architectures, did not perform as well on the original dataset. NasNetMobile, optimized for efficiency, may lack the depth and complexity required to capture subtle differences between disease classes in an imbalanced setting. ResNet152V2 may have been more prone to overfitting due to its complexity and the limited representation of certain classes, as indicated by higher initial training losses and slower convergence (Fig. 16). To address class imbalance and improve generalization, we applied strategic data augmentation techniques and early stopping. Data augmentation involved applying various transformations to the images to increase the diversity of the training data, particularly for underrepresented classes. Early stopping was employed to prevent overfitting by halting training when the validation loss ceased to decrease. After training on the augmented dataset with early stopping, we observed that DenseNet201 maintained its accuracy at 77.14% (Table 16), demonstrating robustness and effective generalization. This consistent performance suggests that DenseNet201 effectively leveraged the augmented data to enhance its learning without introducing overfitting. The dense connectivity inherent in DenseNet201's architecture allows each layer to access feature maps from all preceding layers, promoting extensive feature reuse. This design is particularly advantageous when dealing with augmented data, as it enables the model to integrate a wide variety of features derived from the augmented images, improving its ability to generalize across both majority and minority classes.

In contrast, NasNetMobile and ResNet152V2 showed slight decreases in accuracy after augmentation and early stopping, dropping to 68.25% and 64.13%, respectively. One possible explanation for this decline is that the increased variability introduced by augmentation may have exceeded these models' capacity to generalize effectively. NasNetMobile, optimized for efficiency and designed for deployment on mobile devices, has a shallower architecture with fewer parameters compared to DenseNet201. While this makes NasNetMobile computationally efficient, it may lack the necessary depth and representational capacity to capture the complex patterns introduced by data augmentation, especially in an imbalanced dataset where minority classes require more nuanced feature extraction. ResNet152V2, although deeper than NasNetMobile, may have been more susceptible to overfitting due to its architectural characteristics. The residual connections in ResNet allow for the training of very deep networks by mitigating the vanishing gradient problem. However, in the context of an imbalanced dataset with augmented data, the model may have overfit to the augmented images of majority classes while still underrepresenting the minority classes. The fluctuations in validation accuracy observed for ResNet152V2 (Fig. 17) indicate that the model struggled to maintain stable generalization, possibly due to over-reliance on features specific to the training data. When we introduced L2 regularization alongside data augmentation and early stopping, DenseNet201 continued to achieve 77.14% accuracy. L2 regularization adds a penalty term proportional to the square of the magnitude of the weights, discouraging the model from assigning excessive importance to any single feature and promoting simpler, more generalizable models. DenseNet201's architecture, with its dense connections and feature reuse, may inherently benefit from L2 regularization, as it naturally avoids redundancy by sharing information across layers. This synergy between the architecture and regularization helps prevent overfitting, even with the additional complexity introduced by data augmentation.

On the other hand, NasNetMobile and ResNet152V2 experienced further decreases in accuracy with the addition of L2 regularization, dropping to 62.86% and 64.76%, respectively. This suggests that these models may have been over-regularized, leading to underfitting, where the models are too constrained to capture the underlying patterns in the data. For NasNetMobile, the combination of a lightweight architecture and strong regularization may have overly limited its capacity to learn from the augmented data. For ResNet152V2, the high depth combined with regularization might have disrupted the delicate balance between fitting the training data and generalizing, particularly in the context of class imbalance. Analyzing class-wise performance provides further insights into why DenseNet201 outperformed the other models after augmentation, regularization, and early stopping. DenseNet201 showed high precision, recall, and F1-scores across most classes, including minority classes like "Nematode" and "Healthy" (Tables 21 and 22). The dense connectivity allows the model to capture subtle features from the augmented images of minority classes, enhancing its ability to distinguish between classes with limited samples. The feature propagation and aggregation inherent in DenseNet201 ensure that information from minority classes is effectively integrated throughout the network. In contrast, NasNetMobile and ResNet152V2 exhibited significant fluctuations in class-wise metrics for minority classes, indicating challenges in generalizing from the augmented data. The architectural limitations of NasNetMobile may have prevented it from effectively learning the augmented features of minority classes, while ResNet152V2 may have overfit to the majority classes despite regularization, failing to generalize to underrepresented classes.

The training and validation loss curves further illustrate the models' learning behavior (Figs. 16 and 18). DenseNet201's training loss decreased steadily, and validation loss remained low and stable, indicating effective learning and generalization without overfitting. The model's ability to maintain low validation loss suggests that it successfully captured the underlying data distribution, including the augmented variations, and applied this knowledge to unseen data. NasNetMobile and ResNet152V2 exhibited higher validation losses and less stable curves, suggesting challenges with overfitting or underfitting. The less stable validation loss indicates that these models struggled to find an optimal balance between fitting the training data and generalizing to new data, particularly in the presence of augmented images and regularization constraints. To validate the robustness and generalizability of DenseNet201, we performed 5-fold cross-validation with stratified splits. DenseNet201 achieved an average validation accuracy of 81.31% (Table 23), indicating that its high accuracy is consistent across different subsets of the data. The stratification ensured that each fold maintained the same class distribution, which is crucial for evaluating performance on imbalanced datasets. The increased average accuracy compared to the initial 77.14% further highlights the model's ability to generalize well and suggests that DenseNet201 effectively

learned robust features applicable across various data partitions. We also assessed DenseNet201 on modified versions of the PlantVillage and Rose leaf disease datasets, which were intentionally imbalanced to mimic real-world conditions. After applying data augmentation, early stopping, and L2 regularization, DenseNet201 achieved perfect classification metrics, attaining 100% accuracy, precision, recall, and F1-score (Tables 27 and 31). These results confirm that our strategies effectively enhance the model's ability to handle class imbalance and generalize to different datasets. The consistent performance across different crops and disease types demonstrates the model's adaptability and the effectiveness of the training methodologies employed.

The superior performance of DenseNet201 after augmentation, regularization, and early stopping can be attributed to several factors. Firstly, its architectural strengths, characterized by dense connectivity, allow for efficient information and gradient flow throughout the network. This facilitates the learning of complex patterns from augmented data, enabling the network to capture both low-level and high-level features essential for distinguishing between similar disease classes. Secondly, the effective utilization of augmented data is enhanced by the model's capacity and architecture, which enable it to incorporate the variability introduced by data augmentation. By learning from a more diverse set of images, DenseNet201 improves its ability to generalize to unseen data, including underrepresented classes. Thirdly, there is a synergy with regularization techniques; DenseNet201 appears to work synergistically with L2 regularization and early stopping. Regularization prevents overfitting by discouraging overly complex models, while the architecture's inherent feature reuse reduces redundancy, making the model more robust to regularization constraints. Lastly, its resilience to class imbalance is notable. The model's ability to learn from augmented minority class samples helps mitigate the effects of class imbalance. By effectively recognizing patterns associated with underrepresented classes, DenseNet201 maintains high performance across all classes.

In contrast, NasNetMobile and ResNet152V2 faced challenges that hindered their performance. NasNetMobile's limited capacity, due to its efficiency-focused design, may lack sufficient depth to capture the complexity introduced by data augmentation, particularly for minority classes. This limitation could prevent the model from effectively learning the nuanced features required for accurate classification in an imbalanced dataset. ResNet152V2, despite its depth, was susceptible to overfitting. The model may have overfit to the training data, especially the majority classes, and struggled to generalize to augmented minority class samples. Additionally, both models may have been over-regularized with the addition of L2 regularization, leading to underfitting and a reduction in their ability to capture essential patterns in the data. This over-regularization could have constrained the models too tightly, preventing them from fitting the training data adequately and thus impairing their overall performance. Overall, DenseNet201's architecture and the training strategies employed allowed it to effectively handle the increased data variability and class imbalance, resulting in superior performance compared to NasNetMobile and ResNet152V2 after augmentation, regularization, and early stopping. The combination of these factors makes DenseNet201 a robust and reliable model for potato leaf disease classification in real-world scenarios where data imbalance and variability are common challenges.

In the previous study, models like EfficientNetV2B3, MobileNetV3-Large, VGG-16, ResNet50, and DenseNet121 were employed, and their performance was tested on both the original and augmented datasets. The augmentation strategy involved basic transformations such as brightness adjustment, flipping, rotating, zooming, and shifting, which resulted in an average of 990–1000 images per class. While this approach addressed the imbalance issue to some extent, it may not have been sufficient for handling the complexity and variability inherent in real-world datasets. Basic augmentations like these, while effective, may not introduce enough diversity in the data to significantly improve the models' ability to generalize. In contrast, our study employed not only basic augmentations but also strategic combinational augmentations, which incorporated complex transformations like brightness adjustments, color contrast modifications, zoom variations, and noise additions, along with combinations of position and lighting changes. These augmentations were applied with the specific goal of increasing diversity in the dataset, especially for minority classes, which helped DenseNet201 better capture subtle patterns across various conditions. By using a more comprehensive augmentation strategy, our approach allowed the model to learn more nuanced features and effectively generalize across imbalanced classes.

Additionally, we integrated early stopping and L2 regularization into our training process, which provided further advantages. Early stopping ensured that the model did not overfit to the training data by halting the training when the validation loss stopped improving. L2 regularization prevented the model from becoming overly complex by penalizing large weights, thus promoting simpler and more generalizable models. These techniques were critical in controlling overfitting, particularly in DenseNet201's case, which naturally benefits from the dense connections and feature reuse across layers. This architecture, combined with our strategic augmentations, helped DenseNet201 maintain balanced performance across both majority and minority classes. The DenseNet121 model from the previous study, despite being part of the DenseNet family, did not perform as well as our DenseNet201 implementation. DenseNet121 achieved a test accuracy of only 59.16% on the original dataset and 58.52% on the augmented dataset, indicating that it struggled with both the complexity and imbalance in the dataset. The primary reason for this lower performance is that DenseNet121 has fewer layers and feature maps compared to DenseNet201, which limits its ability to capture the fine-grained details necessary for effective disease classification. DenseNet201, with its deeper architecture and more extensive feature reuse, was better suited to handle the variability introduced by our comprehensive augmentation strategy. Moreover, the synergy between L2 regularization and DenseNet201's architecture further contributed to its superior performance, allowing it to effectively balance between underfitting and overfitting.

Compared to other models from the previous study, DenseNet201 also demonstrated superior performance. EfficientNetV2B3, while achieving a relatively high accuracy of 73.63% on the original dataset, is optimized for efficiency rather than depth, which limits its ability to capture complex patterns, particularly in an imbalanced dataset like ours. Similarly, MobileNetV3-Large achieved a lower accuracy of 72.03%, as its architecture, designed for lightweight applications, may not have the representational capacity to distinguish between the subtle differences in disease symptoms across classes. VGG-16, an older architecture, performed the worst with an accuracy of 59.81% due to its lack of modern innovations like residual connections or dense connectivity, which makes it less effective for more complex and imbalanced datasets. ResNet50, though deeper and more advanced than VGG-16, achieved only 68.17% accuracy, likely due to its residual connections being less effective in handling the wide variability introduced by our strategic augmentations. In contrast, DenseNet201's dense connectivity, which ensures efficient gradient flow and feature reuse, allowed it to outperform these models, especially when combined with advanced augmentation techniques and regularization. Its deeper network provided the necessary capacity to handle the complexity of the dataset, leading to its superior classification performance across all classes.

The findings of our study have significant implications for deploying deep-learning models in agricultural disease detection systems. Accurate and timely identification of plant diseases is crucial for implementing effective control measures, reducing crop losses, and ensuring food security. The demonstrated ability of DenseNet201 to accurately classify multiple potato leaf diseases, even in the presence of class imbalance and data variability, positions it as a strong candidate for real-world applications. Integrating DenseNet201 into mobile or edge devices can provide farmers with on-the-spot disease diagnosis

using images captured in the field. The model's robustness to variations in environmental conditions ensures reliable performance in real-world settings where controlled imaging conditions are not feasible. Techniques such as model pruning, quantization, and knowledge distillation can be employed to optimize DenseNet201 for deployment on devices with limited computational resources. The methodologies developed in this research are generalizable and can be adapted to other crops and diseases. By customizing augmentation strategies to simulate specific environmental conditions, models can be trained to recognize diseases in various contexts, enhancing scalability and versatility. While DenseNet201 demonstrated superior performance, exploring more modern architectures like EfficientNet, Vision Transformers (ViT), and Convolutional Vision Transformers (CvT) could potentially yield even better results. EfficientNet models achieve high accuracy with fewer parameters, making them attractive for deployment on devices with limited resources. Vision Transformers capture long-range dependencies in images, which could improve differentiation between diseases with subtle differences.

However, several challenges must be considered when adopting these architectures. Modern architectures often require large amounts of data to train effectively. Collecting extensive labeled datasets in agriculture can be challenging due to resource constraints and the rarity of certain diseases. Advanced models may have higher computational demands, making them less practical for deployment on mobile or edge devices common in agricultural settings. Complex models with many parameters are more susceptible to overfitting, especially with small or imbalanced datasets. Understanding the decision-making process of complex models is crucial for user acceptance in real-world applications.

5. Conclusion and future scope

In this study, we focused on enhancing the accuracy and robustness of potato leaf disease classification by addressing common challenges such as class imbalance and overfitting. We evaluated three deep learning models — NasNetMobile, ResNet152V2, and DenseNet201 — across various experimental settings that involved strategic data augmentation, early stopping, and L2 regularization. Our comprehensive experiments demonstrated that DenseNet201 consistently outperformed the other models, achieving superior accuracy and generalization, particularly in handling imbalanced datasets. Initially, DenseNet201 achieved an accuracy of 77.14% on the original imbalanced dataset, surpassing NasNetMobile and ResNet152V2, which achieved accuracies of 69.21% and 66.03%, respectively (Table 16). The dense connectivity inherent in DenseNet201's architecture facilitates efficient information flow and feature reuse, enabling it to learn complex patterns essential for distinguishing between the seven classes of potato leaf diseases, especially those that are underrepresented. NasNetMobile and ResNet152V2, while effective architectures, faced challenges due to their design constraints and susceptibility to overfitting or underfitting in the presence of class imbalance and augmented data.

The application of data augmentation and early stopping maintained DenseNet201's performance, while the other models experienced slight decreases in accuracy. Introducing L2 regularization further solidified DenseNet201's robustness, preventing overfitting without compromising its ability to learn from the augmented data. Class-wise performance analysis revealed that DenseNet201 effectively recognized patterns associated with minority classes, mitigating the effects of class imbalance. The training and validation loss curves illustrated that DenseNet201's learning remained stable and effective across epochs, indicating a strong generalization capability. Validation through 5-fold cross-validation with stratified splits resulted in an average validation accuracy of 81.31% for DenseNet201 (Table 23), confirming its consistent performance across different data subsets. Moreover, testing on modified versions of the PlantVillage and Rose leaf disease datasets, which were intentionally imbalanced to mimic real-world conditions, demonstrated DenseNet201's exceptional ability to generalize to new, unseen data. The model achieved perfect classification metrics — 100% accuracy, precision, recall, and F1-score — after applying data augmentation, early stopping, and L2 regularization (Tables 27 and 31).

These findings have significant implications for real-world agricultural applications. Accurate and timely identification of potato leaf diseases is crucial for implementing effective control measures, reducing crop losses, and ensuring food security. DenseNet201's robustness and high accuracy make it a strong candidate for deployment in practical settings. By integrating the model into mobile or edge devices, farmers and agricultural professionals can benefit from on-the-spot disease diagnosis using images captured in the field. Techniques such as model pruning, quantization, and knowledge distillation can optimize DenseNet201 for deployment on devices with limited computational resources, enhancing accessibility in remote or resource-constrained environments. The methodologies developed in this research are generalizable and can be adapted to other crops and diseases. Customizing augmentation strategies to simulate specific environmental conditions allows models to be trained to recognize diseases in various contexts, enhancing scalability and versatility. This adaptability is essential for building comprehensive plant disease detection systems capable of handling a wide range of agricultural challenges globally.

Despite the promising results, several challenges and opportunities for future research remain. One of the primary challenges is the limited size and diversity of the dataset. While our dataset was diverse, it may not capture the full variability of potato leaf appearances under different environmental conditions. Future work should focus on collecting a larger and more diverse dataset, including images from various potato species, growth stages, and environmental settings. This expansion will help the models generalize better and reduce the risk of overfitting. Addressing the similarity between certain disease classes is another critical area for future research. Some diseases, such as Phytophthora and Fungi, present visually similar symptoms, leading to misclassification. Developing methods for fine-grained image classification can help distinguish between visually similar diseases. Incorporating higher-resolution images or multi-spectral imaging could provide more detailed features for the models to learn from, improving their ability to differentiate between similar classes. Exploring more recent architectures like EfficientNet, Vision Transformers (ViT), and Convolutional Vision Transformers (CvT) could potentially enhance classification accuracy and efficiency. These models may offer better parameter efficiency and improved performance but require addressing challenges related to data requirements and computational complexity. Employing sophisticated data augmentation methods, such as Generative Adversarial Networks (GANs) to generate synthetic images, could further enhance the model's ability to handle class imbalance and improve robustness.

Optimizing models for faster computation without compromising accuracy is essential for facilitating real-time applications, especially in resource-constrained environments. Techniques such as model pruning, quantization, and knowledge distillation should be explored to create efficient models suitable for deployment on mobile devices or edge computing platforms. This optimization will make the technology more accessible to farmers in remote areas, where computational resources may be limited. Integrating the model into broader agricultural decision support systems can provide farmers with actionable recommendations beyond disease identification. By combining disease detection with information on management practices, weather forecasts, and crop growth stages, farmers can make informed decisions to optimize crop health. Developing a user-friendly interface that integrates model predictions into a practical tool for farmers is crucial. The interface should be intuitive, support multiple languages, and provide real-time feedback on disease identification. Utilizing prompt engineering can help create an adaptable system that meets the needs of diverse user groups. Ethical and privacy considerations are also important when deploying

such technology. Handling sensitive data requires compliance with data protection regulations and ethical considerations. Establishing clear guidelines for data collection, storage, and usage is important to maintain user trust and ensure the responsible deployment of technology. Collaborations with agricultural extension services, research institutions, and farmers can facilitate the gathering of diverse and representative data. Employing citizen science approaches, where farmers contribute images of diseased plants, can expand the dataset and improve model robustness. Involving stakeholders in the development process ensures that the solutions are grounded in practical needs and can lead to higher adoption rates. Addressing computational constraints is another area for future work. In regions with limited internet connectivity or electricity, solutions must be designed to function offline and be energy-efficient. Developing models and systems that can operate under such constraints will maximize accessibility and utility, ensuring that the benefits of the technology reach those who need it most.

Our comprehensive analysis demonstrates that DenseNet201, when combined with strategic data augmentation, early stopping, and L2 regularization, provides a robust solution for potato leaf disease classification in imbalanced and variable datasets. The model's consistent performance across different experimental settings and datasets underscores its potential for practical deployment in agricultural settings. By effectively addressing challenges such as class imbalance and overfitting, our methodologies offer effective strategies to enhance model performance. Future research focused on expanding datasets, exploring modern architectures, optimizing models for deployment, and integrating user-friendly interfaces will further enhance the utility and impact of deep learning applications in agriculture. By addressing these challenges, we can contribute to the development of effective and scalable solutions for agricultural disease management, ultimately supporting global efforts to improve food security and promote sustainable agricultural practices.

CRediT authorship contribution statement

Prit Mhala: Conceptualization, Software, Formal analysis, Investigation, Writing – original draft, Visualization. **Anushka Bilandani:** Methodology, Validation, Data curation, Writing – review & editing. **Sanjeev Sharma:** Project administration, Supervision, Methodology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

We have shared the dataset link in the manuscript.

References

Adedoja, A. O., Owolawi, P. A., Mapayi, T., & Tu, C. (2022). Intelligent mobile plant disease diagnostic system using nasnet-mobile deep learning. *IAENG International Journal of Computer Science*, *49*(1), 216–231.

Agarwal, M., Sinha, A., Gupta, S. K., Mishra, D., & Mishra, R. (2020). Potato crop disease classification using convolutional neural network. In *Smart systems and ioT: Innovations in computing: proceeding of SSIC 2019* (pp. 391–400). Springer.

Almanzor, E., Birell, S., & Iida, F. (2023). Rapid development and performance evaluation of a potato planting robot. In *Annual conference towards autonomous robotic systems* (pp. 15–25). Springer.

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., et al. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*, 1–74.

Andre, C. M., Legay, S., Iammarino, C., Ziebel, J., Guignard, C., Larondelle, Y., et al. (2014). The potato in the human diet: A complex matrix with potential health benefits. *Potato Research*, *57*, 201–214.

Arshad, F., Mateen, M., Hayat, S., Wardah, M., Al-Huda, Z., Gu, Y. H., et al. (2023). PLDPNet: End-to-end hybrid deep learning framework for potato leaf disease prediction. *Alexandria Engineering Journal*, *78*, 406–418.

Ashikuzzaman, M., Roy, K., Lamon, A., & Abedin, S. (2024). Potato leaf disease detection by deep learning: A comparative study. In *2024 6th international conference on electrical engineering and information & communication technology* (pp. 278–283). IEEE.

Asif, M. K. R., Rahman, M. A., & Hena, M. H. (2020). CNN based disease detection approach on potato leaves. In *2020 3rd international conference on intelligent sustainable systems* (pp. 428–432). IEEE.

Binnar, V., & Sharma, S. (2023). Plant leaf diseases detection using deep learning algorithms. In *Machine learning, image processing, network security and data sciences: Select proceedings of 3rd international conference on MIND 2021* (pp. 217–228). Springer.

Chakraborty, K. K., Mukherjee, R., Chakroborty, C., & Bora, K. (2022). Automated recognition of optical image based potato leaf blight diseases using deep learning. *Physiological and Molecular Plant Pathology*, *117*, Article 101781.

Chandra, N. M., Reddy, K. A., Sushanth, G., & Sujatha, S. (2022). A versatile approach based on convolutional neural networks for early identification of diseases in tomato plants. *International Journal of Wavelets, Multiresolution and Information Processing*, *20*(01), Article 2150043.

Chen, W., Chen, J., Zeb, A., Yang, S., & Zhang, D. (2022). Mobile convolution neural network for the recognition of potato leaf disease images. *Multimedia Tools and Applications*, *81*(15), 20797–20816.

Dash, A., Sethy, P. K., & Behera, S. K. (2023). Maize disease identification based on optimized support vector machine using deep feature of DenseNet201. *Journal of Agriculture and Food Research*, *14*, Article 100824.

Fauzi, M. D., Adhinata, F. D., Ramadhan, N. G., & Tanjung, N. A. F. (2022). A hybrid DenseNet201-SVM for robust weed and potato plant classification. *Jurnal Ilmiah Teknik Mesin, Elektro dan Komputer (JITEKI)*, *8*(2), 298–306.

Ghosh, H., Rahat, I. S., Shaik, K., Khasim, S., & Yesubabu, M. (2023). Potato leaf disease recognition and prediction using convolutional neural networks. *EAI Endorsed Transactions on Scalable Information Systems*, *10*(6).

Hasan, M. Z., Zahan, N., Zeba, N., Khatun, A., & Haque, M. R. (2021). A deep learning-based approach for potato disease classification. *Computer Vision and Machine Learning in Agriculture*, 113–126.

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).

Islam, M. A., & Sikder, M. H. (2022). A deep learning approach to classify the potato leaf disease. *Journal of Advances in Mathematics and Computer Science*, *37*(12), 143–155.

Jha, P., Dembla, D., & Dubey, W. (2024). Deep learning models for enhancing potato leaf disease prediction: Implementation of transfer learning based stacking ensemble model. *Multimedia Tools and Applications*, *83*(13), 37839–37858.

Kamarudin, M., & Ismail, Z. H. (2022). Lightweight deep CNN models for identifying drought stressed plant. *1091*, In *IOP conference series: Earth and environmental science*. (1), IOP Publishing, Article 012043.

Khalifa, N. E. M., Taha, M. H. N., Abou El-Maged, L. M., & Hassanien, A. E. (2021). Artificial intelligence in potato leaf disease classification: a deep learning approach. In *Machine learning and big data analytics paradigms: Analysis, applications and challenges* (pp. 63–79). Springer.

Kingma, D. P. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Kittusamy, K., Krishnakumar, B., Aswath, A., Gowtham, P., & Vishal, S. (2021). Terrain identification and land price estimation using deep learning. *2387*, Article 140030. http://dx.doi.org/10.1063/5.0068625,

Kumar, A., & Patel, V. K. (2023). Classification and identification of disease in potato leaf using hierarchical based deep learning convolutional neural network. *Multimedia Tools and Applications*, *82*(20), 31101–31127.

Kumar, H., Virmani, A., Tripathi, S., Agrawal, R., & Kumar, S. (2021). Transfer learning and supervised machine learning approach for detection of skin cancer: Performance analysis and comparison. *Drugs and Cell Therapies in Hematology*, *10*, 1845–1860.

Lanjewar, M. G., Morajkar, P., & P, P. (2024). Modified transfer learning frameworks to identify potato leaf diseases. *Multimedia Tools and Applications*, *83*(17), 50401–50423.

Lefebvre, M., Gil, S., Brunet, D., Natonek, E., Baur, C., Gugerli, P., et al. (1993). Computer vision and agricultural robotics for disease control: The potato operation. *Computers and Electronics in Agriculture*, *9*(1), 85–102.

Lewkowycz, A., & Gur-Ari, G. (2020). On the training dynamics of deep networks with L_2 regularization. *Advances in Neural Information Processing Systems*, *33*, 4790–4799.

Mahum, R., Munir, H., Mughal, Z.-U.-N., Awais, M., Sher Khan, F., Saqlain, M., et al. (2023). A novel framework for potato leaf disease detection using an efficient deep learning model. *Human and Ecological Risk Assessment: An International Journal*, *29*(2), 303–326.

Paul, H., Ghatak, S., Chakraborty, S., Pandey, S. K., Dey, L., Show, D., et al. (2024). A study and comparison of deep learning based potato leaf disease detection and classification techniques using explainable AI. *Multimedia Tools and Applications*, *83*(14), 42485–42518.

Prechelt, L. (2002). Early stopping-but when? In *Neural networks: Tricks of the trade* (pp. 55–69). Springer.

Rachburee, N., & Punlumjeak, W. (2022). Lotus species classification using transfer learning based on VGG16, ResNet152V2, and MobileNetV2. *IAES International Journal of Artificial Intelligence*, *11*(4), 1344.

Rajbongshi, A., Sazzad, S., Shakil, R., Akter, B., & Kaiser, M. S. (2022). FlowerNet: An extensive rose leaves dataset for disease recognition applying machine learning and deep learning models. http://dx.doi.org/10.17632/7z67nyc57w.2, (Version V2) Mendeley Data. https://doi.org/10.17632/7z67nyc57w.2.

Rashid, J., Khan, I., Ali, G., Almotiri, S. H., AlGhamdi, M. A., & Masood, K. (2021). Multi-level deep learning model for potato leaf disease recognition. *Electronics*, *10*(17), 2064.

Reddy, B., Mandal, R., Chakroborty, M., Hijam, L., & Dutta, P. (2018). A review on potato (Solanum tuberosum L.) and its genetic diversity. *International Journal of Genetics*, ISSN, 0975–2862.

Reis, H. C., & Turk, V. (2024). Potato leaf disease detection with a novel deep learning model based on depthwise separable convolution and transformer networks. *Engineering Applications of Artificial Intelligence*, *133*, Article 108307.

Rex, E. (2019). Plant disease.

Rozaqi, A. J., & Sunyoto, A. (2020). Identification of disease in potato leaves using convolutional neural network (CNN) algorithm. In *2020 3rd international conference on information and communications technology* (pp. 72–76). IEEE.

Saeed, Z., Khan, M. U., Raza, A., Sajjad, N., Naz, S., & Salal, A. (2021). Identification of leaf diseases in potato crop using deep convolutional neural networks (DCNNs). In *2021 16th international conference on emerging technologies* (pp. 1–6). IEEE.

Seitov, S. K., Saitov, S. R., & Rakintsev, D. S. (2022). Using the agricultural robot to control potato diseases. In *Agrarian perspectives XXXI* (p. 267).

Shabrina, N. H., Indarti, S., Maharani, R., Kristiyanti, D. A., Irmawati, I., Prastomo, N., et al. (2023). Potato leaf disease dataset in uncontrolled environment. [Mendeley Data, V1]. https://doi.org/10.17632/ptz377bwb8.1.

Shabrina, N. H., Indarti, S., Maharani, R., Kristiyanti, D. A., Prastomo, N., et al. (2024). A novel dataset of potato leaf disease in uncontrolled environment. *Data in Brief*, *52*, Article 109955.

Sharma, S., Anand, V., & Singh, S. (2021). Classification of diseased potato leaves using machine learning. In *2021 10th IEEE international conference on communication systems and network technologies* (pp. 554–559). IEEE.

Sharma, A., Azeem, N. A., & Sharma, S. (2022). Coffee leaf disease detection using transfer learning. In *International conference on advanced network technologies and intelligent computing* (pp. 227–238). Springer.

Sholihati, R. A., Sulistijono, I. A., Risnumawan, A., & Kusumawati, E. (2020). Potato leaf disease classification using deep learning approach. In *2020 international electronics symposium* (pp. 392–397). IEEE.

Singh, G., & Yogi, K. K. (2023). Comparison of RSNET model with existing models for potato leaf disease detection. *Biocatalysis and Agricultural Biotechnology*, *50*, Article 102726.

Spooner, D. (2013). Solanum tuberosum (potatoes). *Brenner's Encyclopedia of Genetics*, 481–483.

Suttapakti, U., & Bunpeng, A. (2019). Potato leaf disease classification based on distinct color and texture feature extraction. In *2019 19th international symposium on communications and information technologies* (pp. 82–85). IEEE.

Tewari, V., Azeem, N. A., & Sharma, S. (2023). Automatic guava disease detection using different deep learning approaches. *Multimedia Tools and Applications*, 1–24.

Tian, J., Chen, J., Ye, X., & Chen, S. (2016). Health benefits of the potato affected by domestic cooking: A review. *Food Chemistry*, *202*, 165–175.

Tiwari, D., Ashish, M., Gangwar, N., Sharma, A., Patel, S., & Bhardwaj, S. (2020). Potato leaf diseases detection using deep learning. In *2020 4th international conference on intelligent computing and control systems* (pp. 461–466). IEEE.

Tsang, S.-H. (2019). Review: NASNet — Neural architecture search network (image classification). URL https://sh-tsang.medium.com/review-nasnet-neural-architecture-search-network-image-classification-23139ea0425d.

Visvanathan, R., Jayathilake, C., Chaminda Jayawardana, B., & Liyanage, R. (2016). Health-beneficial properties of potato and compounds of interest. *Journal of the Science of Food and Agriculture*, *96*(15), 4850–4860.

Yang, C., Everitt, J. H., Du, Q., Luo, B., & Chanussot, J. (2012). Using high-resolution airborne and satellite imagery to assess crop growth and yield variability for precision agriculture. *Proceedings of the IEEE*, *101*(3), 582–592.

Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 8697–8710).