Multimedia Tools and Applications (2024) 83:4465–4517
https://doi.org/10.1007/s11042-023-15809-9

PlaNet: a robust deep convolutional neural network model for plant leaves disease recognition

Munish Khanna[1] · Law Kumar Singh[2] · Shankar Thawkar[3] · Mayur Goyal[1]

Received: 16 February 2022 / Revised: 22 March 2023 / Accepted: 10 May 2023 /
Published online: 25 May 2023

Abstract

Researchers are looking for new ideas that can greatly increase the amount of food grown while also cutting costs. For precision agriculture to work, pests, weeds, and diseases must be easy to find and identify on plant leaves. Most plant diseases have symptoms that can be seen, and plant pathologists now agree that infected plant leaves are the best way to find them. This is a good use for computer-aided diagnostic systems because diagnosing diseases manually (hands and eyes) takes a long time, and the effectiveness of diagnostic treatment depends on how well the pathologist does (intra-observer variability). Based on what we need right now, we need a model that doesn't need much pre-processing and doesn't need manual functional (feature) extraction. So, in this study, methods for identifying and classifying plant diseases from leaf images taken at different resolutions are described that are based on deep learning. Using Deep Convolutional Neural Network (DCNN) image analysis, the main goal of this research is to help tell the difference between healthy and unhealthy leaves. We have also come up with a new model called **PlaNet**. Its performance has been compared to that of other common CNN models. We did a lot of testing and verification on 18 well-known CNN models based on deep learning, including one ensemble model made up of the five best models. We did this on four different combinations of three well-known standard benchmark datasets. The suggested **PlaNet** model has been tested, and the results show that it works in a highly efficient manner. It has been found that, among all the testing experiments, the best average-wise performance achieved was up to 97.95% accuracy, 0.9752 AUC, 0.9686 F1-score, 0.9707 sensitivity, 0.9576 precision, and 0.9456 specificity, respectively. So, the results of the tests show that the proposed system, which is based on deep learning, can classify different types of plant leaves quickly and accurately. When used alone or with other datasets (four different combinations of three datasets), the suggested model works very well. This shows that the proposed model is more flexible, general, and scalable than other approaches that are already in use. The fact that it must be able to accurately process a single image of a plant leaf in less than a second demonstrates its real-time capabilities.

Keywords Deep convolution neural networks · Deep learning · Ensemble model · Image classification · Leaf diseases identification · Transfer learning

✉ Munish Khanna
munishkhanna.official@rocketmail.com
Extended author information available on the last page of the article

1 Introduction

Plants are an integral element of the ecosystem and are thus essential for human life [93]. They are valuable for food and medicine, as well as raw materials for industrial use. Maintaining accurate plant species identification helps ensure the survival and prosperity of all of the flora and fauna. Recognition of plant types aids in Ayurveda and helps in identifying the various functions and applications of plants. Botanists use the method of plant leaf classification known as they go about identifying species [67].The agricultural industry is the economic backbone of India [10]. The emergence of industrial agriculture has had a noticeable influence on our climate. The ecosystem has been quite badly impacted by the high promotion of cultivation. The high use of chemical products in pesticides and fertilizers, which impact life on earth and raise the amount of soil, water, and air contamination, given the practicesadopted means farmers are consciously or unknowingly influenced by improved efficiency [91]. Toxic chemical use has resulted in massive chemical buildup in the land, water, climate, animals, and human bodies. Unnatural fertilizers and pollutants are short-term solutions with long-term environmental consequences. It has resulted in a global fall in agriculture. This is an issue that practically every farmer in the world faces today [9].

The scientific study of plant diseases is known as phytopathology. In precision agriculture, disease detection using pictures of plant leaves is a crucial area of research [29]. Nonetheless, growing and cultivating crops may be susceptible to several illnesses. The disease's symptoms and detection procedures are mostly discernible by observing infected plant leaves. When there are too many plant species and temperature fluctuations, plant diseases mutate and spread more rapidly, which may make them more hazardous. Observing plant disease spread by leaf appearance data is possible. Initially, farmers diagnosed these illnesses by observing the signs on plant leaves. To categorize these illnesses, human involvement may be required, either by analyzing the physical appearance of plants, i.e., the visual symptoms, or by conducting a laboratory examination. Manual identification is more difficult for a rookie or inexperienced farmer than for specialists dealing with plants [3]. For disease treatment, it is vital for a farmer to be aware of the disease status of their crop. By watching the leaves and symptoms, experts and seasoned farmers may spot a few types of plant problems with the naked eye. Additionally, the severity of the illness may be determined by practice and continuous monitoring. As crops and pathologies proliferate, farmers struggle to detect illnesses, prompting them to make wrong judgments and assumptions. Due to the diversity and quantity of crops and their psychopathological pests, even competent crop specialists and plant pathologists are sometimes prohibited from accurately identifying specific illnesses and are finally forced to rely on negative assumptions and recommendations [4]. To effectively identify plant diseases, the pathologist must have keen observational skills and the ability to appropriately interpret signals. The condition is undetectable, even by expert pathologists. For agronomists, the use of a sophisticated and clever professional approach to analyze plant diseases is necessary information. On the other hand, using a model like the one described is a good way to help farmers who don't know much about agriculture economics [47].

A computer-aided artificial intelligence-based automated system (or device) may be developed to help classify plant diseases by their presence and symptoms, which may be of great benefit to amateurs. This will prove to be a beneficial approach for farmers, as it will warn them when a disease epidemic is taking place [20]. More effective diagnosis of plant diseases is critical for sustainable agriculture and to avoid excessive wasting of financial and other capital. Plant diseases have a devastating effect on agricultural products and can lead directly to slow development, which can have a negative impact on returns. The planet

is expected to suffer an economic loss of up to $20 billion a year. Early identification of plant diseases may help mitigate these economic losses. Due to regional variations that can impede correct detection, complex environments are the most challenging obstacle for scientists. Moreover, orthodox approaches are primarily focused on professionals, consultants, and manuals, but they are often pricey, time-intensive, and hard-to-detect. So, for the sake of industry and the environment, it is critical to respond quickly and consistently to plant diseases [42]. Plant diseases may be predicted utilizing computer vision and deep learning techniques, which can reduce (or minimize) their impact on crops. Changes in agricultural supply and quality have an effect on national economic losses. Because of this, it is important to find diseases in plants when they are still young [58, 63].

Early-stage diagnosis of any illness can address many problems. Identified plants or leaves can be extracted from or isolated from the field or handled. Picture analysis may be a solution to plant disease identification [70]. The method we propose here is to diagnose diseased plants by extracting properties from the deeper layers of their leaves using plant photographs. During the preparation and testing, images of diverse conditions and sizes are considered. This approach lets farmers detect diseased plants and alert them until they disperse over a wider region in the early stages of the disease. Picture processing and data analysis are advanced deep learning methods that effectively and theoretically generate results from information. Numerous attempts have been made to date to create techniques for addressing pest and disease concerns and preventing crop losses [7]. Although several strategies have been employed and established to manage plant diseases, early detection and correct diagnosis of illnesses are critical in resolving this problem and minimizing disease spread and crop loss. Despite this, illness identification continues to be difficult because of a lack of critical infrastructure and a proliferation of computational resources and information. Predicting and identifying illnesses early, monitoring plant health, and implementing management measures are crucial for sustainable agriculture [74]. It is an ongoing struggle for researchers to keep ahead of these pests and diseases and to continue battling and supplying farmers and the agricultural industry with the most advanced instruments for plant disease control. Image recognition is used a lot more in plant disease detection and severity research now that computer systems have gotten better [86].

Innovations in artificial intelligence (AI) have paved the way for the development of faster and more accurate automated methods for illness diagnosis. A wide variety of disorders are identified using AI technology. AI, image processing, and graphics processing units (GPU) innovations have broadened their agricultural uses [58]. In addition, it has been noticed that artificial intelligence and machine learning are being developed in numerous agricultural domains, such as disease detection, soil tracking, weed management, bug diagnosis, crop analysis, drones, and weather forecasting. Agriculture and the food business are negatively impacted by crop diseases. The majority of plant diseases show several obvious signs. In order to identify the features of any plant disease, the visual model must have keen observational abilities. Theoretically, machine vision-based techniques may facilitate the early detection of plant diseases. Image detection with deep learning is important research because it could be used to keep an eye on huge fields of crops and find signs of disease on plant leaves in real time [70].

Serious leaf diseases have a devastating effect on plants and can result in total failure. Their condition is difficult todiagnose since it needs time and consistent monitoring. Due to the manual process, misdiagnosis rates are rather high. Numerous methods, such as preprocessing, the corresponding number of features, and recognition by machine learning algorithms, have been used in the development of recent computer vision techniques. Recent agricultural applications of data analytics include surveillance,

biometrics, medical imaging, and fruit categorization. Convolutional neural networks (CNN) are now regarded as the most effective tool for object identification and recognition, audio recognition, recommender systems, and natural language processing. The introduction of deep learning into the agricultural sector indicates the modernization of agriculture [7, 74]. The greatest advantage of deep learning is the framework it offers for testing on enormous datasets. Deep learning, a relatively new technique for picture identification and data analysis, aims to provide very promising outcomes. It may also be used for agricultural issues [62, 86]. We can create a machine to detect and identify plant ailments to increase productivity. In general, the leaves of plants serve as the first site of detection for the majority of plant illnesses, which may be diagnosed automatically via the use of efficient image recognition systems [21, 95]. Early detection of plant diseases is essential for applying effective disease prevention strategies, plant growth management strategies, and disease control measures that prevent disease transmission [11, 44, 79]. Frameworks for the detection and categorization of plant diseases have been proposed. CNN is often used for identifying artifacts. Form, colour, and texture are the fundamental characteristics used to identify objects in photographs. However, the output of several features is superior to a single feature type. It has been shown that leaf colour may predict some types of illnesses. Various CNN models have been proposed for this specific purpose. Nonetheless, growing and cultivating crops may be susceptible to several illnesses. For disease treatment, it is vital for a farmer to be aware of the disease status of their crop. The authors of [25] say that mapping leaves with different types of automated detection has become more useful over time.

This article attempts to use a Deep CNN based model to solve plant leaf disease identification problems. Its main advantage is that it eliminates the need for the feature engineering process in image classification. Deep learning extends classical methods of machine learning by adding more complex models and hierarchical representations to the learning process. Transfer learning, implemented in this work, uses a set amount of training data and reduces it, thereby cutting down on training time and costs. In order to build the best CNN model using insufficient training data, image augmentation is required, which is also implemented in this work. This technique of image enhancement will generate artificial images for training. In this research, in terms of novelty, we have provided two novel models. The first one is a lightweight CNN that was created from scratch (**PlaNet**), and the second one is the vote-based ensemble model of five top performing models. Fine tuning and transfer learning have also been applied. We have evaluated the performance of our suggested ones with 16 pre-existing CNNs. Three openly accessible datasets are selected for validation of the suggested technique. Four tests were performed to validate the proposed models' (s) adaptability, resilience, dataset-independent performance, and generality. Such a comprehensive multiclass experimentation procedure with such a large number of nets and three distinct benchmark datasets has rarely been achieved previously (typically, researchers perform one or two). Not only does our method require the least human input, but it also gives results in a fraction of a second that are better than those of traditional machine learning methods. The disease identification performance accuracy retrieved from this study indicates that deep CNN models are promising and may have a significant impact on disease identification efficiency, as well as potential for disease detection in real-time agricultural systems. Our recommended models acquire good performance on numerous criteria and can be tested to be deployed in the agriculture industry to properly and rapidly identify plant health. They are incredibly helpful for human society and particularly for the farming sector.

In short, the contributions are summarized as follows.

- We suggested a light-weighted deep learning network (PlaNet) for the automatic identification and categorization of plant leaf pictures, including illness. All models are trained via transfer learning, and the proposed model is compared to seventeen other cutting-edge deep learning models. The suggested CNN model does not require a complicated network structure of pre-trained models with a high number of hidden layers and parameters, and it has the benefits of less storage space and faster response while maintaining the same degree of accuracy as reported in the literature. According to a qualitative analysis of the data, the PlaNet model, which is non-destructive, automated, and quick with cheap computational costs, outperforms current approaches in terms of greater precision and improved classification accuracy. Thus, the suggested model proved effective for automated categorization.
- The performance of the suggested model has been thoroughly investigated using simulations. In totality 4 experiments are performed on the combinations of shortlisted datasets to identify the best performing, dataset independent (generalized), model. Extensive experiments are conducted using 3 benchmark publically available well-accepted datasets including Plant Village dataset containing 54,306 images with 38 classes.
- In our research, we have used an ensemble of pre-trained deep learning models. The advantage is that our proposed and implemented combines the predictions of five best models and performs well, in terms of all efficiency measuring parameters, in challenging situations. In complex settings including several sick leaves, ensembling has reduced forecast variance and increased accuracy.
- This work uses deep learning to diagnose plant illnesses using leaf images. We propose a CNN model that learns plant category characteristics without human interaction, removing the need for human feature development. Without agricultural expertise, biological data is extracted. Using deep models, this study enhances plant disease categorization. Deep learning models are better at categorizing [3]. Deep models can use unprocessed raw data. Deep models enable transfer learning by leveraging previously trained models on bigger datasets. To prevent overfitting, picture datasets were augmented. Our research shows that deep learning improves image processing accuracy. Deep learning's benefits are encouraging for smarter, more sustainable agriculture and food security. Farmers will be able to figure out what's wrong with their crops by taking pictures of the sick leaves instead of relying on expensive expert opinions.
- Symptom identification and visualization: after the classification of the condition, the user is able to see the areas of the leaf image that are indicative of the ailment (Symptoms of disease). This kind of symptom visualization benefits beginner users by offering more information on the disease process. In addition, symptom visualization allows the user to estimate the disease's spread to other plants.

The remaining paper is organized as follows: The extensive previews of literal works performed by other scholars are given in Sect. 2. Section 3 goes over the specifics and settings of various models (including the proposed one) for multi-class classification of plant disease(s) images. Results, graphs, and thorough discussion are described and analyzed in Sect. 4. In Section 5, a comparison is shown between the proposed work and a few prominent recently published relevant literatures. Section 6 is dedicated to the practical application of the work and the conclusion. Finally, section 7 deals with the limitations of the study, and section 8 deals with future assignments.

Table 1, presented below, depicts the abbreviations used in this paper and the corresponding description. Table 2 presents the symbols used in the mathematical equations and their corresponding meaning.

2 Related works

Several researchers in this area have been working on a model to predict leaf disease in plants. In this section, wehave performed an in-depth literature review exploring some relevant prominent studies published in last few years.

In Yang et al. [99], the authors suggested a model for classifying hyper spectral images based on tree diagram and logistic regressions, and the latter proved to be more reliable. Neural networks have been implemented in deep learning and artificial intelligence. In Babu et al. [8], author introduced and developed the ANN, back propagation paradigm for detecting a broad variety of pests and diseases on leaves. Back Propagation algorithm was applied to measure the gradient descent values for ANNswhichare important for minimizing errors in the first place. In Sankaran et al. [73], the researchers studied most influential traditional methods of plant disease identification. Spectroscopic related, imaging-based, and volatile profiling-based methods of plant disease detection both are existing methods of plants disease detection. The comparison paper discusses the advantages and disadvantages of these two practices. Another strategy was suggested in the analysis of (Wetterich et al. [97]) to cope with citrus greening by integrating the features derived from the SVM model. This approach increased the correctness of the device and result in average accuracy of 85% and 92.8%. The writers of (Rumpf et al. [70]; Calderón et al. [14]) used smart classifiers to evaluate spectral images for detections of plant diseases. To boost classification in hyperspectral imaging, (Guo et al. [32]) suggested the use of the K-NN algorithm with a directed filter technique. The Disease-related characteristics of Tomato leaves induced by yellow leaf curl were subject. The SVM with different kernels and 200 photos of safe and unhealthy tomato leaves was subject. The model correctly identified 90% of the instances. Deep learning related applications are useful to the practitioners for automatic diagnosis. Researchers have not addressed all the plant disease issues as yet. Newer methods are expected in

Table 1 Abbreviations Table

Abbreviation	Description	Abbreviation	Description
AI	Artificial Intelligence	GPU	Graphics Processing Unit
SVM	Support Vector Machine	A/CNN	Artificial/Convolutional Neural Network
DCNN	Deep Convolutional Neural Network	AUC	Area under the curve
ReLU	Rectified linear activation unit	PSO	Particle swarm optimization
GA	Genetic Algorithm	VGG	Visual Geometry Group
TP	True Positive	TN	True Negative
FP	False Positive	FN	False Negative
ROI	Region of interest	K-NN	K nearest neighbours
C-GAN	Conditional Generative Adversarial Network	DL	Deep Learning
FT	Fine Tuning	TL	Transfer Learning

Table 2 Symbols and meaning Table

Symbol	Meaning	Symbol	Meaning
F_j	Features Map	F_{j-1}	Convolution feature of the previous Layer
F_0	Original Image	W_i	Weight of the ith Layer
b_i	Offset vector of the ith layer	$\chi(.)$	rectified linear unit (ReLu) function
$\boldsymbol{down-sampling}(.)$	downsampling	y_n^{m-1}	feature vector in the previous layer
s	Pooling layer	$\boldsymbol{soft}\,max(.)$	Softmax function
${C_{onv}}_i^m$	Output of the convolved layer	$\boldsymbol{Bias}_i^m$	bias matrix with the *ith* iterative region of operation
w	Convolved window	C_i^m	Iterated convolved window
$\boldsymbol{windowsize}_{i,j}^{(m-1)}$	Window sized	∂	Standard deviation
B	batch	y_i'	output
$\boldsymbol{Diamention}_{\boldsymbol{output}}$	output dimension	y	Input instance
y_h	Height of the input instance	y_d	Color channel dimension of the input instance
y_w	Width of the input instance		

the field of performance. Author (Ferentinos [26]) merged CNNs with separate datasets to provide diagnosis for various plant diseases. For the identification mission, authors of (Lee et al. [53]; Grinblat et al. [31]) established a deep CNN approach on plant leaves for noisy photos. Deep learning was used to classify plants. This method was used to save tomato crop from diseases (Kamilaris and Prenafeta-Boldú [45]). The primary objective undertaken by [21] included forty separate researches that were extended to several agricultural problems. Batch normalization, dropout, and stochastic pooling are implemented to impact neural network to discriminate Multiple Sclerosis. The average classification accuracy achieved was 98.77%. In this study, a 13-layer neural network was built in (Zhang et al. [102]) and then it achieved a 93.25% rate of deep learning in the final experiment. CNNs may be used to identify multi-temporal crops of different elevations. This process was used in numerous crops such as maize. Using this form, the accuracy achieved was 85.54%. The proposed data model was implemented utilizing k-Nearest Neighbors, Decision Trees, SVM, and DCNN. This segment deals with the conceptual modeling analysis model and evaluation dataset.

Feature extraction is important in ML and model identification based on its popular uses, including entity classification, medical imaging and agriculture (Mittal et al. [56]; Mittal et al. [57]). A variety of methods to the diagnosis of the disease from the database are outlined in the literature. Plants Village (Mohanty et al. [58]), made up of thousands of photos, is the popular publicly accessible leaflet dataset. Extraction of color characteristics from multiple color areas is of significant significance in plant disease identification. The model CNN for the identification of tomato-leaf disease was proposed by Brahimi et al. [13]. They also used pre-trained CNN templates from AlexNet and GoogleNetImageNets and chosen tomato datasets are qualified in fine tuning. For a final classification, the extracted features are transferred to SVM. The best results and milestones were 99.86% and 99.18%.Moth-flame is a different form of disease detection. In (Hassanien et al. [35]), the authors picked most durable features utilizing moth-flame fitness function. In the next stage, the SVM is used as classificator and the input is robustly chosen. Benchmark datasets for testing this method are collected from the UCI learning data warehouse. Performing comparisons to PSO and GA demonstrate a better output of the mentioned moth-flame approach. A deep learning predictor for identification of tomatoes and pest attacks was introduced by Fuentes

et al. [27] in real time. Faster RCNN, FCN and multi-box single shot detectors were used. For deep function extraction, authors concatenate along VGG and ResNet. The data annotation and improve methodology have both been implemented to enhance the accuracy of identification. They established a data collection of pest and tomato diseases for laboratory research, which achieved more than 80% accuracy. Deep Network Refinement Filter Bank developed in Fuentes et al. [28] for the identification of tomato and other diseases. This method (Khan et al. [49]) comprises of three core measures, namely identifying the septic zone and identifying a wrong positive sample type. In the same paper techniques for the identification and recognition of fruit diseases, a correlation coefficient and a profound function extraction were developed. For the extraction of functionality, pre-trained deep models, such as AlexNet and VGG19 were used. Mohanty et al. [58] revealed a groundbreaking diagnostic system of smartphone-assisted plant leaf diseases. The datasets were made available and included 53,306 pictures of both good and poor groups in total. In their study authors Brahimi et al. [13], used HSI and colour rooms to prepare segmented rooms. In recent years, ML and DL are employed to identify and segment pictures.

Wang et al. [96] have demonstrated the use of pre-trained models VGG16, VGG19, ResNet50 and Inception-V3 to identify various varieties of Apple. They proved that test accuracy increased from around 87% to 93% due to transfer learning. Ma et al. [54] utilized a CNN model with three convolution layers to classify four distinct infections of cucumber pictures and got a classification accuracy of 93.4%. SVM and Random Forest were also utilized by researchers to classify infection, with accuracy rates of 81.9% and 84.8%, respectively. Zhang et al. [103] has used global pooling dilated CNN to compute94.65% accuracy in classifying the diseases of cucumber plants. The data rises to around 30,000 measurements. A new weighted count-based segmentation approach based on chi-square test and threshold value has been defined by Sharif et al. [75] Principal Component Analysis (PCA) was used to pick the three colour channels adaptively from red, green, blue, and hue range. Camargo and Smith [15] have established a method of segmenting diseased sections of banana leaflets using the Histogram of Intensities. Instead of the standard method of choosing the minima points of the local plateaus as the cut of the local maximums are placed and used to assess the cut of each plateau using its location in the histogram. Hamuda et al. [33] also addressed the different strategies of the plant segmentation. They discussed that the major pre-processing criteria must be fulfilled before conducting the image segmentation. Segmentation can be done with colour or using particularly red index. Arribas et al. [6] have developed an RGB dependent segmentation algorithm for sunflower crops. This seeds are quickly divided using pixels whose colours are green and black. Khan et al. [50] describe a novel classification of cucumber plant diseases. In this article, the writers have suggested a five-step method for features extraction and segmentation. Nevertheless, they have not provided any evidence that show the precise severity of the disorder. It is seen that this model achieves high accuracy, but due to the complexities of the design, it is computationally too intense to use. Rumpfet al. [70] also determined that there are three forms of diseases that can be extracted from sugar beet leaf. Mokhtar et al. [59] have also used SVM to identify two forms of tomato herbal diseases to ensure that 92% of the diseases are correct. In the method of detection of three diseases in wheat leaves, Johannes et al. [43] used the Naive Bayes strategy. Authors have tested other strategies in ML in (Yang & Guo [98]), which offer a summary of the machine-learning method, including Naive Bayes classifier, Support Vector Machine (SVM), K-means clustering, articulating ANN and Decision Trees. The researchers used computer training to classify plant pathogens (disease-causing bacteria) genes. Several scientists have recently suggested that deep learning is a safer way of achieving high precision in detecting plant diseases (Rangarajan et al. [69]; Ferentinos

[26]). Researchers have commonly used transfer learning in pre-trained models in other fields and recorded successful results in classical disease identification.

The most popular form of DL for image data processing is CNN. Mohanty et al. [58] have applied common CNN models such as AlexNet (Krizhevsky et al. [52]) and GoogleNet (Szegedy et al. [87]) in plant class and disease prediction in PlantVillage data collection (Yang &Guo [98]) photos. For plant classification and disease from 58 distinct crop and disease groups, Ferentinos [26] demonstrated the usage of pretrained models Alexnet and VGG. Authors using PlantVillage with other photos, registered a 99.53% accuracy with a pre-trained model of the VGG. In Zhang et al. [101], TL has been used in defining 9 forms of maize leaf disease using GoogleNet. Rangarajan et al. [69] who employed AlexNet and VGG16 to train and identify photos of tomato plant diseases from PlantVillage datasets are also documenting the usage of transference learning. They revealed that AlexNet provides around 97.29% accuracy and the estimate of VGG16 on 373 test images per class is about 97.49%.In (Fuentes et al. [27])authors proposed to classify the disease and bounding box of the disordered portion of the tomato-leaves photos a region-based CNN(R-CNN) and region-based FCN(R-FCN). The diseased portion have been manually annotated and the pre-trained VGG and ResNet(He at al. [36]) models used for their CNN prototypes to stop over-timing for any specific class. 83.06% was the highest mean accuracy for R-CNN and for R-FCN received 85.98%.The majorities of practitioners focusing on recognizing plants have utilized previously qualified models of other fields and have extended transition education for their unique tasks. Many scholars have concurrently attempted to classify several plant diseases (Rangarajan et al. [69]). It should however be remembered that pre-trained models for scenarios of classes have been created. The number of groups of plant diseases is limited, and overcrowding is the usage of these broad models. Although there have been several attempts to establish low order CNN models for recognizing plant diseases(Wang et al. [96]; Khamparia et al. [48]; Hu et al. [38]).Khamparia et al. [48] suggested a small model with two layers of convolution; checked with a precision of 93.7% in five epochs on 10 disease groups of mixed crops. In 4 Tea Leaf Disease groups of fundamental data raise, Hu et al. [38] have also built the CNN model from scratch and recorded 92.5% accuracy.

Plants are the main energy source, Cseke et al. [22], and play an important role in the resolution of global warming. Diseases threaten the lives of this important source of food, Strange and Scott [82]. Pomegranate disease detection and classification research was performed by the researchers, Pawar and Jadhav [66]. In this paper researchers discussed that it has been possible to classify different plant conditions in plants by using leaf pictures. Patterns which are noticeable and observable were collected and analyzed in determining a plant disease. Islam et al. [40] suggested picture segmentation study and the Multi-class SVM to diagnose potato disease. Authors began to use hybrid image processing and ML strategies to assess illness of leaf pictures. A classification method and a support vector machine assess the accuracy of 95% over 300 pictures. Majumdar et al. [55] provided an important way in which the picture of a wheat leaf disease can be studied. FCM is used on data points for the functionalities chosen from wheat leaves pictures. The first step was the amount of clusters to divide the input images into good and diseased pictures. Leaf marked as ill was split into four groups, which cope with the likelihood of 4 diseases arising with FCM. In this point, the number of clusters has been increased to four. The writers have proposed to pick a feature set dependent on interclass variance and intraclass variance. Singh and Misra [77] investigated many strategies for sunflower leaf disease classification. The findings of work on the different sunflower plant clustering and distinguishing strategies for diseases is discussed. In the

segmentation process of the sheet pictures, a particle swarm optimization algorithm was used. In Agarwal et al. [2], Grape Leaf Multi-Class SVM has been observed and graded. SVM was proposed in the binary classification strategy for the multi-class situation with easy manipulation. Research was carried out through the acquisition of RGB features translated into LAB. HSI picture reiterated that when the backdrop light was changed, the color had not changed. The properties of the HSI picture have also now been applied to the database. Gokulnath and Devi [30] proposed an effective loss-fused CNN model to identify the plants diseases. In (Tiwariet al. [92]), a DL-based approach was proposed for the plant infection recognition and categorization from leave images captured in different resolutions. Dense CNN architecture was trained on a large plant leaves image dataset. The aim of this published study (Tiwari [91]) was to compare the performances of Deep Neural Network (DNN) and Deep CNN on public leaf database. In the case of DNN, hybrid form and texture features were used as hand-crafted features, while non-handcrafted features were used for classification in the case of convolution. The simulation results confirm that the deep CNN-based deep learning architecture outperforms the handcrafted feature-based solution in terms of classification accuracy.

This paper introduces a DL-based method for recognizing and categorization of plant diseases from leaf images (Tiwari et al. [93]). For this investigation, a broad image collection of plant leaves, including 12 distinct species and 22 distinct categories, was utilized. Multiple intra-class and inter-class modifications to the training dataset make training a deep learning model more complicated and challenging. An exhaustive analysis of several deep neural networks with varying optimizers and learning rates was conducted. The recommended method had an average cross-validation accuracy of 98.68% and an average test accuracy of 97.72%. In Sujatha et al. [83] the accuracy of disease categorization gained through testing is rather remarkable, since DL techniques surpass ML methods in disease identification as follows: SGD-86.5% versus SVM-87% versus VGG-19–87.4% versus Inception-v3–89% versus VGG-16–89.5%.In this paper, practitioners (Abbas et al. [1]) presented a deep learning-based solution for tomato disease diagnostics that leverages the C-GAN to generate synthetic images of tomato plant leaves for data enhancement. The proposed data augmentation approach boosts the network's generalizability. Then, using TL, a DenseNet121 model was skilled on unreal and real data to categorize tomato leaf images into ten disease classes. The recommended method classified tomato leaf images into five, seven, and ten classes with 99.51%, 98.65%, and 97.11% accuracy, respectively. Several academics are merging computer vision and nature inspired approaches to automate the uncovering of plant diseases based on photographs of leaf surfaces. This study summarizes numerous aspects of this type of research, as well as their advantages and disadvantages (Vishnoi et al. [94]). Common infections and the research landscape at various levels of such detection systems are examined. Examining contemporary feature extraction strategies in an effort to identify those that appear to perform well across several crop kinds. This article presents an overview of various research initiatives that utilize computer vision and machine learning approaches to automate plant disease categorization and detection systems. The study examines a variety of appropriate techniques for image collection, preprocessing, lesion segmentation, feature extraction, and eventually classifiers. Several challenges that arose during the module's feature extraction have also been described. Nagaraju et al. [64] presented a more effective method for supplying a sufficient number of training images to a CNN model in order to address overfitting and classification issues. Two learning strategies are presented to overcome the problem of constrained datasets and CNN model overfitting during

classification: image preprocessing and transformation algorithm (IPTA) and image masking and REC-based hybrid segmentation algorithm (IMHSA). Subsequently, the histogram threshold method was utilized to generate all of the possible regions for subdividing the diseased leaf into identical portions. In order to evaluate the efficacy of the IPTA method, a novel CNN model was also given. The model is trained on two independent datasets, one before the IPTA application and one after. They justify that the validation accuracy was 65% prior to the use of IPTA. The proposed model achieved 73% validation accuracy using IPTA, thereby eliminating the overfitting problem. In this paper, Mukhopadhyay et al. [61] suggested a new technique for automatically diagnosing tea leaf diseases based on image processing technology. Picture clustering based on the Non-dominated Sorting Genetic Algorithm (NSGA-II) is presented for finding disease spots in tea leaves. Then, PCA and multi-class SVM were used on tea leaves to minimize feature size and identify disease, respectively. The results indicate that the recommended algorithm detects the type of disease present in tea leaves with an average accuracy of 83%. This study examined five different diseases of tea leaves. DCNN models are used in this paper (Hassan et al. [34]) to identify and diagnose plant diseases based on their leaves, as CNNs have generated exceptional results in the field of machine vision. This study's authors replaced conventional convolution with depth-separable convolution, which reduces the number of parameters and the calculation cost. The models were trained using a publicly available dataset containing 14 unique plant species, 38 discrete disease classifications, and healthy plant leaves. The highest accuracy through various CNNs was measured to be 99.56%.Using the optimized parameters, the MobileNetV2 architecture was also compatible with mobile devices.

This technique (Bansal et al., [10]) employs a pre-trained ensemble of three CNNs to classify apple tree leaves into one of four categories. The proposed model gained has a 96.25% accuracy rate. The anticipated approach can identify various diseases on leaves with an accuracy rate of 90%.This work presents a computer vision method for segmentation and classification of leaf diseases in order to describe the necessity and automate the disease diagnosis system (Chouhan et al. [20]). For disease area segmentation, a hybrid neural network paired with super pixel clustering is proposed. Various algorithms are utilized to analyse colour, shape, and texture characteristics. Jatrophacurcas L. and Pongamiapinnata L. were used as biofuel plants. In conclusion, seven distinct machine learning techniques were utilized to classify the pictures into three groups. This study by Jinag et al. [42] collects and enhances 40 photographs of each leaf disease, focusing on three types of rice leaf diseases and two types of wheat leaf diseases. The implemented work constructed the VGG16 model and then applied TL. The model has a 97.22% accuracy for rice leaf illnesses and a 98.75% accuracy for wheat leaf infections. In comparative research, this technique outperforms the single-task model, the reuse-model method in TL, the ResNet50 model, and the DenseNet121 model. The results of the experiments show that the enhanced VGG16 model and multi-task TL technique can both find diseases on rice and wheat leaves at the same time.

Sun et al. [85] offered a CNN model for real-time detection of apple leaf diseases on mobile devices. Using data augmentation and data annotation tools, AppleDisease5, a collection of apple leaf diseases consisting of basic backdrop shots and complex background images, is initially developed. Then, by rebuilding the common 33 convolution, a core module known as the MEAN block (Mobile End AppleNet block) is created to enhance exposure speed and lessen model dimension. Meanwhile, the Apple-Inception module is constructed by utilizing GoogleNet's Inception module and substituting each of Inception's 33 convolution kernels with MEAN blocks. Using the MEAN block and

the Apple-Inception module, a novel model for detecting apple leaf diseases, MEAN-SSD (Mobile End AppleNet based SSD algorithm), was compiled. In this work, the CNN model was constructed and trained using a very small dataset with a significant class imbalance (Sambasivam et al. [72]). They were able to obtain promising results for cassava mosaic and other cassava diseases detection by training CNNs from scratch on an imbalanced dataset. Class weight, focus loss, SMOTE, and various image dimensions were employed, with the input vector shape (448, 448, 3) yielding the best results. Computer vision-based monitoring is crucial for developing field guava plants. Mostafa et al. [60] identify various guava plant species by employing a DCNN-based data augmentation strategy based on color-histogram equalization and the unsharp masking procedure. To increase the number of changed plant photographs, nine out of 360 perspectives were utilized. The suggested study applies five neural network topologies to identify various guava plant species. ResNet-101 produced the mainly precise categorization outcome, with 97.74% accuracy, according to the test data. Tabular summary of few studies is also shown using Table 3.

3 Materials and models

In this section, we have a presented a thorough discussion on subject datasets and models implemented, including proposed one.

3.1 Dataset description

Three datasets of infected and healthy plant leaf images are gathered from kaggle and other resources. These datasets are freely available to the public and widely accepted by the research community. The details are provided in the below Tables 4 and 5. Moreover, we have combined the second and third datasets to create a customized fourth dataset in which the number of classes is 30 and the count of images is equivalent to the sum of images in the second and third datasets, respectively. The first dataset selected for the subject is the plant village dataset, which is a multinational general database used to research algorithms for investigating plant diseases using DL. To assess the proposed approach's performance, we conducted several experiments on this large database of plants and their diseases. The PlantVillage database contains a massive 54,306 photos of farm leaves; the collection comprises 38 distinct classes of 14 different plant species, each having healthy and diseased leaf pictures. All samples were obtained in a laboratory setting [58]. The Plantvillage database was used to download the 38 classes of images database, which included apples, potatoes, grapes, tomatoes, and maize. This dataset consists of RGB images of healthy and diseased crop leaves, which belongs to 38 diverse classes. Sample images of all datasets are shown below (refer to Fig. 1a, b and c). The second and third dataset sare obtained from the Kaggle website, the active link for which is listed in Table 4 below (last access: January 1, 2022). We then resized each image to 224×224 pixels for the next executing steps in the pipeline of the DL system. Then, when we did data augmentation, we used different flips like zoom, rescale, and rotation, because more images will help us train with the data we have, and at the same time, each class must have an equal number of images. The histogram equalization process is also employed to improve the quality of the pictures.

Table 3 Comparative table of previous studies

Year of Publication	Paper	Methods	Dataset	Plant/classes	Accuracy
2003	Yang et al. [99]	Regression Tree(C&RT) Approach	Hyperspectral images were taken	Three stages of crop development	89%
2009	Camargo and Smith [15]	Image Processing Techniques	The set of images used in this study taken from (INIBAP, http://bananas.biodiversityinternational.org) provided a set of pictures of banana and plantain crops	Two Classes (Healthy or Diseased)	Not Mentioned
2010	Rumpf et al. [70]	SVM	Up until 21 days after the first inoculation, daily records of the hyperspectral reflectance data acquired from infected and non-inoculated leaf samples were recorded. These leaf samples had either been inoculated or not.	Two class of healthy sugar beet leaves and diseased leaves	97%
2011	Arribas et al. [6]	Features Extraction and Features selection, Neural Network for training and classification	Created own datasets of 192 color images of sunflower crops	Classify two classes either sunflower leaf or non-sunflower leaf	Accuracy-Not mentioned AUROC-90%
2015	Calderon et al. [14]	Linear Discriminate analysis, SVM	High-resolution thermal and hyperspectral imagery were acquired with olive area	Sixclass Asymptomatic, Initial, Low, Moderate, severe, class recall	59.0%(LDA) 79.2%(SVM)
2015	Lee et al. [53]	CNN	MalayaKew (MK) Leaf Dataset consisting of 44 classes	44 different plant species	0.995(MLP)
2016	Grinblat et al. [31]	DCNN	866 leaf images provided by INTA (InstitutoNacional de TecnologíaAgropecuaria, Oliveros,Argentina)	Three class white bean,Red bean and Soyabean	Mean accuracy (92.6 ± 2)%
2017	Wetterich et al. [97]	FIS(Fluorescence imaging System) Technique, SVM, ANN	Samples taken from Florida, USA	Not Mentioned	92.8% (SVM) 92.2%(ANN)

Table 3 (continued)

Year of Publication	Paper	Methods	Dataset	Plant/classes	Accuracy
2017	Brahimi et al. [13]	CNN(AlexNet and GoogleNet)	www.PlantVillage.org (14,828 images) Taken	Nine Tomato diseases	(97.354 ± 0.290)%AlexNet without Pretraining (97.711 ± 0.149)%GoogleNet without Pretraining
2017	Hassanien et al. [35]	Improved moth-flame approach, SVM classifier	6 datasets from UCI machine learning data repository	Two classes: (Normal or Diseased)	90.5%
2017	Fuentes et al. [27]	Faster Region Based Convolution Neural network(Faster R-CNN)	Tomatao image were collected from several tomato farms located in Korea	10 Classes	90.60%
2017	Wang et al. [96]	VGG16,VGG19,Inception-V3 and ResNet50	Apple black rot images from Plant Village Dataset	4 classes	90.4%
2017	Zhang et al. [103]	13 Layer CNN	Fruit Datasets (i) Six month of onsite collecting via Digital Camera (ii) download from https://images.google.com (iii) download from https://images.baidu.com	18 Classes	94.94%
2017	Johannes et al. [43]	Image processing Techniques	Wheat 2014(w-2014) Wheat 2015(w-2015)	Three class (Rust, Septoria, Tan Spot)	78%(accuracy) 0.83(AUC)
2018	Khan, M. A. et al. [49]	VGG16,Caffe,AlexNet,Multi class SVM	6309 sample images of apple and banana fruits collected from Plant Village, CASC IFW and Plant Village for banana datasets	7 Classes	98.60%(plant village and CASC-IFW)
2018	Ma et al. [54]	DCNN	PlantVillage(https://plantvillage.org), ForestryImages(https://www.forestryimages.org/)	Four cucumber disease classes	93.4%
2018	Sharif et al. [75]	Different Features Extraction,Support Vector Classification	Create Own Datasets contained 580 images	Five Classes including four diseases and One Healthy	96.9%

Table 3 (continued)

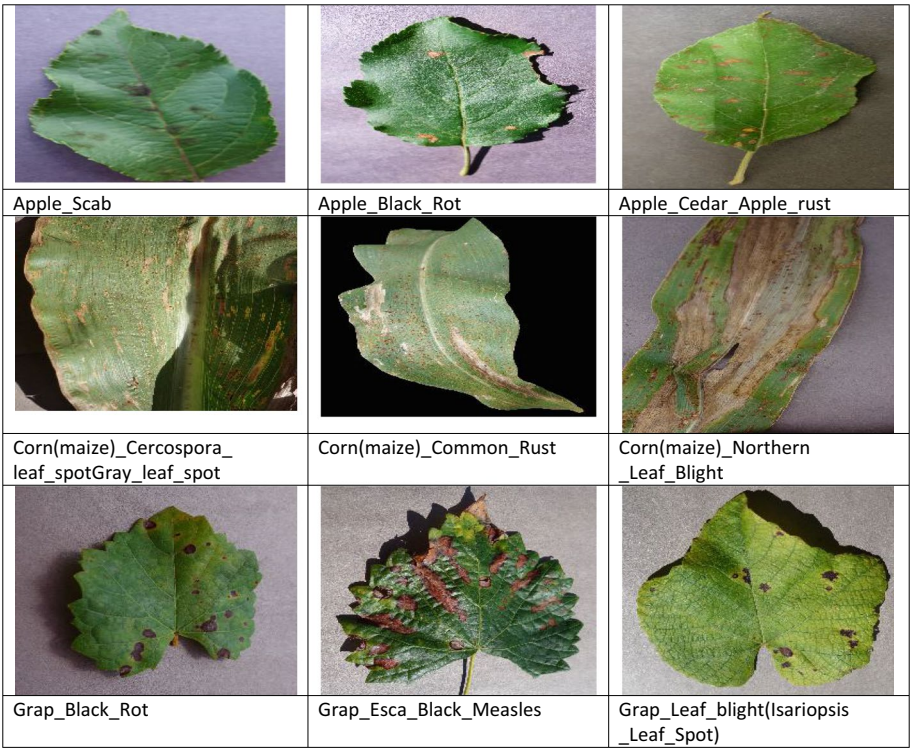
Year of Publication	Paper	Methods	Dataset	Plant/classes	Accuracy
2019	Zhang et al. [102]	Global pooling dilated CNN	600 cucumber disease leaf and 100 normal leaf	Two classes (normal or diseased)	94.65%
2019	Yang and Guo [98]	SVM,ANN	Own Data collection	Two classes: Diseased or healthy	92.8% for SVM and 92.2% ANN
2020	Geetharamani et al. [29]	Nine Layer DCNN	Plant Village Dataset	Healthy or disease for each datasets	96.46%
2020	Agarwal et al. [3]	CNN Model, Traditional ML method	Plant Village Dataset	Tomato crop disease or healthy	94.9%(KNN) 93.5%(VGG16)
2020	Agarwal et al. [4]	Conv2DModel	Cucumber plant Dataset taken from [100]	Two class cucumber plant disease or healthy	93.75%
2020	Khan, M. A. et al. [50]	VGG-19,Multi class SVM(M-SVM),cubic-SVM(C-SVM)	Cucumber disease leaf image	5 Classes	92.1%(M-SVM)
2020	Sambasivam and Opiyo [72]		Kaggle Dataset, Dataset consist of 5-fine grained cassava leaf disease categories	5 classes	93%
2021	Bansal et al. [10]	DenseNet121,EfficientNetB7	Plant Pathology and Plant –Micobe Biology Section of the Cornell Universtiy	Healthy or Diseased	96.25%
2021	Jiang et al. [42]	VGG16 Model	Collected 40 images of each leaf disease	Three kind of rice leaf and two kind of wheat leaf diseases	97.22%
2021	Abbas et al. [1]	C-GAN	Plant Village Dataset	5 Classes tomato disease	97.1%
2021	Hassan et al. [34]	CNN	Plant Village Dataset	Healthy or Diseased	96.56%
2021	Sun et al. [85]	CNN method proposed Mean Block(Mobile End AppleNet based SSD Algorithm)	Apple leaf Dataset (AppleDisease5)	5 classes of Apple Leaf Dataset	83.12%

Table 4 Description of the datasets used in the study

Dataset Serial Number	Total Number of Images (Before Augmentation)	Total Number of Images (After Augmentation)	Address (Last Access Date 01:01:2022)
1.	54,306 (38 Classes)	87,900	(https://github.com/spMohanty/PlantVillage-Dataset/tree/master/raw)
2.	19,708 (15 Classes)	20,738	(https://www.kaggle.com/emmarex/plantdisease)
3.	5208 (15 Classes)	5549	(https://www.kaggle.com/vbookshelf/v2-plant-seedlings-dataset)
4.	24,916 (30 Classes)	26,287	Combining above mentioned dataset 2 and 3; to create new, 4th dataset for fourth experiment.

Table 5 Hyperparameter values of the deep learning models implemented in this study

Parameters	Number of units	Number of Layers	Activation Function	Filter	Pool size	Padding	Depth	Learning Rate
VGG-16	16	16	ReLU	250	4×4	1	16	0.001
VGG-19	19	19	ReLU	250	4×4	1	19	0.001
Xception	14	14	ReLU	250	4×4	1	14	0.001
InceptionResnetV2	200	200	ReLU	224	4×4	1	20	0.001
NASNetLarge	24	24	SoftMax	224	4×4	1	24	0.001
Proposed Model	54	54	SoftMax	224	4×4	1	54	0.001
ResNet-101	101	101	SoftMax	224	4×4	1	101	0.001



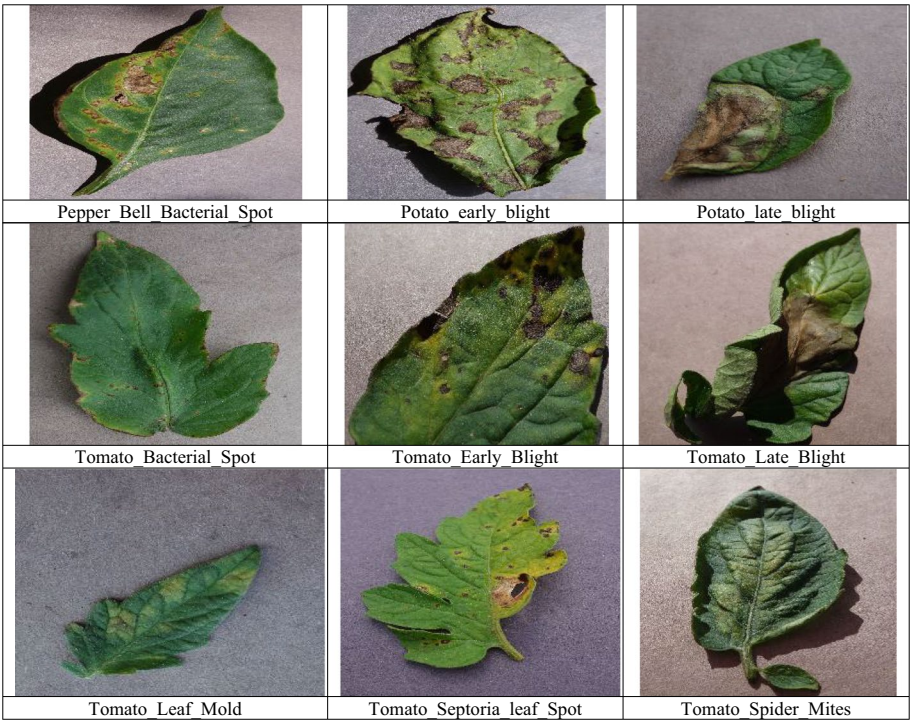
(a)

Fig. 1 **a** Sample images of leaves from Dataset #1 **b** Sample images of leaves from Dataset #2 **c** Sample images of leaves from Dataset #3

3.2 Models Implemented

Over the last several years, CNN has made great gains in entity identification and image categorization. Image categorizations tasks have traditionally depended on hand-engineered features such as SIFT, HoG, SURF, and others, followed by the application of a learning algorithm in these feature spaces. As a result, the effectiveness of these strategies was deeply reliant on the primary established qualities. Feature engineering is a hard and lengthy method that must be redone if the problem or the associated dataset changes significantly. This problem exists in all conventional attempts to diagnose plant diseases using computer vision since they rely mostly on hand-engineered features, picture augmentation techniques, and a number of other difficult and time-consuming procedures. Furthermore, conventional machine learning-based disease classification algorithms focus on a less amount of categories, which are frequently contained within a single crop.

DL models are the most often utilized architecture because they can learn significant attributes from input pictures at several convolutional layers. With high classification accuracy and a low mistake rate, DL can tackle difficult problems fast and efficiently. The DL model is composed of numerous parts (an input layer, convolution layer, max pooling, average pooling, activation functions, fully connected layers, and a softmax

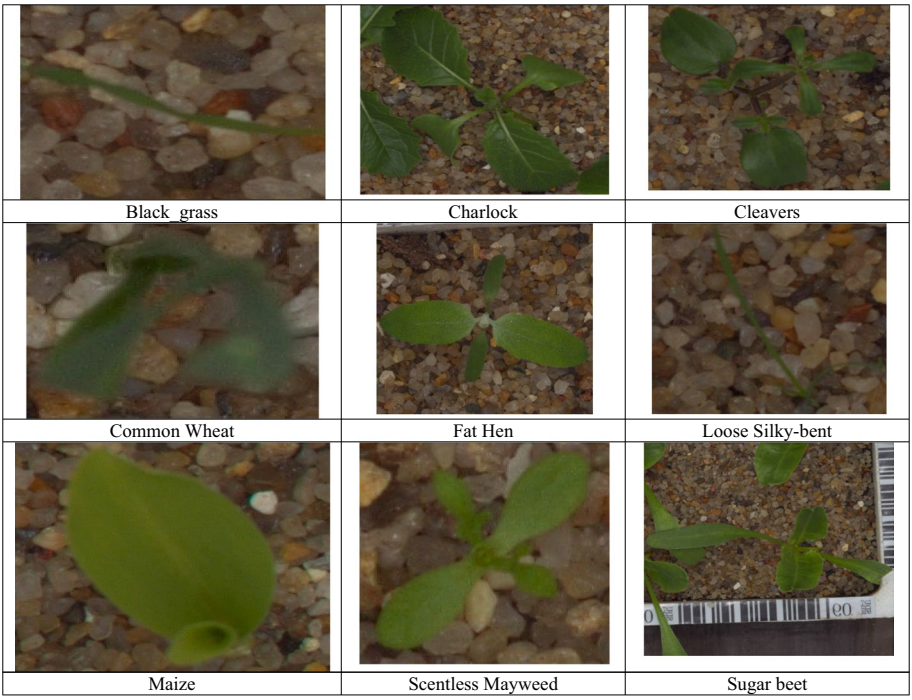


(b)

Fig. 1 (continued)

layer). CNN is a kind of neural network that utilizes fewer artificial neurons than typical feed-forward neural networks and is especially good at image recognition. The structure of CNN is normally made up of four main types of layers. Convolutional, activation, pooling, and fully linked layers comprise a CNN. Deep CNNs have excelled in text and non-text classification, person identification, picture classification, and ear recognition, among other tasks. In this situation, predictions are made on instances that do not originate from the same distribution as the training data. Deep learning and transfer learning, together with GPU improvements, have resulted in a robust plant disease classification and identification tool [26]. Plant diseases are now classified and identified using a deep learning technique. The excellent findings produced with this technique conceal a number of difficulties that are seldom investigated in comparable trials. This article looks at the main factors that affect how deep neural networks used in plant pathology are made and how well they work.

Our method, which is based on end-to-end supervised training with the DCNN architecture, outperforms existing techniques using hand-crafted features in typical benchmarks by a wide margin. The absence of a labor-intensive step of feature creation, as well as its generalizability, makes the proposed technique an especially intriguing choice for a practical and scalable approach to computer inference of plant diseases. We used the DCNN architecture to train a model on shots of plant leaves to identify crop species as well as the presence and kind of illness on images the computer had never seen before. The great



(c)

Fig. 1 (continued)

precision of the PlantVillage collection of 54,306 images enabled us to achieve this aim. Thus, without any feature engineering, the model appropriately categorizes crops and illnesses into 38 distinct dataset 1 groupings (PlantVillage dataset of 54,306 pictures). With datasets 2 and 3, the same great results and performance were found in the last three runs.

Transfer learning (TL) is the process of applying a previously trained network to a new problem. TL is widely used in DL because it enables a network to be trained with less input and more accuracy. A computer utilizes information from past work to increase the generality of a second in transfer learning. During transfer learning, the last few layers of the trained network are changed with new layers, such as a wholly connected layer and a softmax classification layer with the number of classes. Transfer learning, which requires just a few layers of pre-trained neural networks to be adapted to incoming input, has considerably decreased the demand for large datasets. Transfer learning is used to handle deep learning, which needs a large number of training instances. The pre-training model developed by training on a big data set may be used to train the model on a small data set, reducing the training time needed for deep learning substantially. FT is a kind of TL that maintains the starting parameters of the model's feature extraction layer [80]. TL is the process through which knowledge learned from training in one kind of topic is used to training in other related activities or domains. Given a source domain Ds and a source task Ts, and a target domain Dt and a learning task Tt, the goal of transfer learning is to improve the performance of the predictive function fT () for the learning task Tt by discovering and transferring latent knowledge from Ds and Ts, where $Ds \neq Dt$ or $Ts \neq Tt$. Deep

learning methods require a large quantity of labelled data to train models, but obtaining a large annotated dataset in a domain is unquestionably difficult. As a result, the transfer learning strategy has become the industry standard and is utilized in real-world applications such as solutions that employ a network that has already been trained on the big ImageNet dataset and simply uses the target dataset to determine the parameters of additional extended layers. The major steps of the transfer learning approach can be (1) Determine the primary network, (2) Create a completely new neural network, and (3) Model creation or improvement(They are frequently trained by minimizing prediction loss).

Convolutional Layers Let F_j represent the feature map of the $j-th$ layer in CNN, then the F_j can be generated as follows:

$$F_j = \chi\left(F_{j-1}W_j + b_j\right) \tag{1}$$

Pooling Layers The $m-th$ pooling layer,the output feature on the $n-th$ local receptive field can be calculated in Eq.(2)

$$y_n^m = down - sampling\left(y_n^{m-1}, s\right) \tag{2}$$

Fully Connected Layer Softmax function is written by

$$soft\max(p)_j = e^{p_j} / \sum_{k=1}^{k} e^{p_k} (for\, j = 1, \ldots . k) \tag{3}$$

Where k represents the dimension of the p vector.

Convolutional Layer The convolutional operation is summarized in the following equation.

$$C_{onv}{}_i^m = Bias_i^m + \sum_{j=1}^{a_i^{(m-1)}} windowsize_{i,j}^{(m-1)} * C_i^m \tag{4}$$

Batch Normalization Batch normalization is a normalization in which a batch of input data is normalized and it can be written as in equation

$$y_i' = \frac{y_i - \beta_B}{\partial_B^2} \tag{5}$$

Pooling Layer The pooling value is calculated as in Equation.

$$Diamention_{output} = y_h * y_w * y_d \tag{6}$$

$\boldsymbol{Diamention_{output}}$ is the notation for output dimension after performing pooling.

We implemented 16 well-known (excluding ensemble and proposed) popular and efficient CNN models to evaluate their performance when solving multi-class classification problems. We have also implemented one ensemble model (of fivebest performing models on the same problem), and to the best of our knowledge, we are the first to implement an ensemble model for the classification of plant leaf diseases. Along with this, we proposed and implemented a new CNN model "**PlaNet**", which is detailed in the following section. The names of implemented models are VGG-16 (Sukegawa et al. [84]), VGG-19 (Simonyan

and Zisserman, [76]), MobileNet (Howard et al. [37]), InceptionResNetV2 (Szegedy et al. [88]), InceptionV3 (Szegedy et al. [88]), EfficientNetB5 (Tan and Le, [89]), ResNet50V2 (Rahimzadeh & Attar, [68]), Xception (Chollet [19]; Simonyan and Zisserman [76]), EfficientNetB4 (Tan and Le, [89]), EfficientNetB6 (Tan and Le, [89]), Self Made-ensemble deep transfer learning model (VGG16+ InceptionResNet-V2 + EfficientNetB5+ EfficientNetB4+ EfficientNetB3) (Singh et al. [78]), DenseNet169 (Huang et al. [39]), EfficientNetB2 (Tan and Le, [89]), ResNet-101 (Ardakani et al. [5]), NASNetLarge (Zoph et al. [104]), EfficentNetB7 (Tan and Le, [89]) and EfficientNetB3 (Tan and Le, [89]).

To suit the models and permit speedier transfer, all photographs are shrunk to a set dimension of 224 × 224 pixels. Following picture scaling, image preprocessing is conducted to blacken the shorter sides of the images, if any, so that they are the same proportion, maintaining the information of the original photos and preventing visual distortion. Augmented data covers a variety of perspectives in order to provide more real-time data visualization and big data usage for deep learning; additionally, data augmentation is used to simulate interference in real-world situations and enlarge the data, which is critical during the model's training stage. To enrich our dataset and develop a more robust and general model, the suggested model employs picture augmentation methods such as random rotation, flipping, and scale shift; horizontal and vertical mirroring; sharpening; brightness and contrast adjustment; and others. Unsharp masking and the histogram equalization approach were used to sharpen the ROI and eliminate any existing noise in the enhanced data. To the best of our knowledge, the strategies suggested in this paper are novel and have the potential to considerably improve model performance by providing a more comprehensive training dataset. The final supplemented and upgraded data was put into a number of fine-tuned cutting-edge classification models, as well as the recommended ones. The suggested leaf disease detection models (PlaNet and Ensemble) are conceived and constructed on the concepts of easy operation, powerful real-time performance, low cost, and energetic promotion. These are end-to-end DL models that can automatically find distinguishing characteristics of leaf diseases (Figs. 2 and 3).

Transfer learning in deep learning is the use of neural network models that have already been trained to solve new problems or tasks. Transfer learning lets us improve the performance of a new model without having to train it from scratch. Instead, we can use what we learned from a model that was already trained on a similar task or domain. Transfer learning works by adapting a model that has already been trained for a different task. We freeze the weights of the pre-trained model for the new task without destroying the valuable knowledge that the pre-trained model has already learned. Transfer learning can be particularly useful in situations where we have limited data for a new task but a large amount of data is available for a related task. In this case, we can use a pre-trained model that has been trained on the related task to improve the performance of the new model.

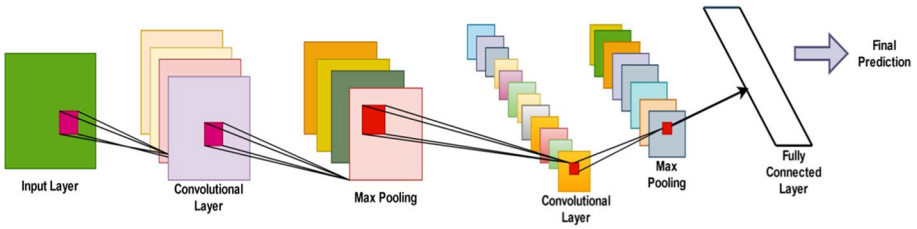


Fig. 2 Architecture of the CNN model

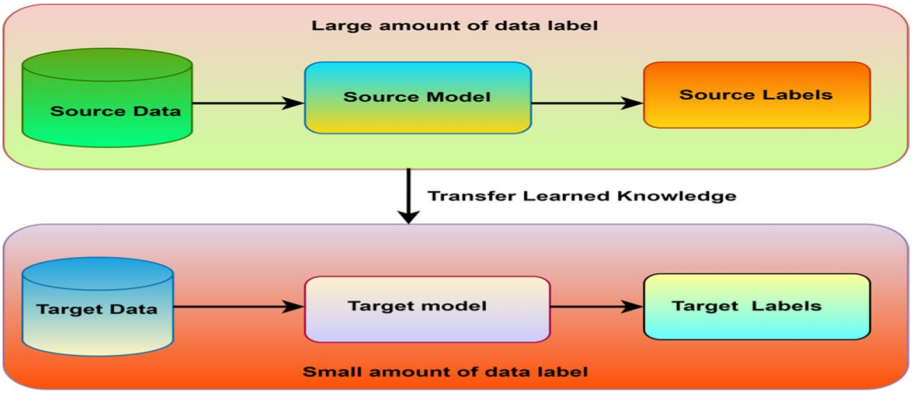


Fig. 3 Basic implementation of the Transfer Learning process

Utilizing predictions from several deep learning classifiers has the drawback of a significant degree of variance. Because each classifier is trained separately with different data in each epoch, each classifier changes its weight on its own and comes up with different results for the same image. Incorporating the predictions of all classifiers into some kind of integration is a simple solution to the aforementioned issue. This will reduce variance, and the ensemble model will generalize more effectively than the individual models. The ensemble approach utilized in this work is based on voting, and the final prediction incorporates the contribution of each classifier. This motivates us to present and implement an ensemble model in this study. We also proposed an ensemble deep transfer learning model (VGG16+ InceptionResNet-V2 + EfficientNetB5+ EfficientNetB4+ EfficientNetB3), as shown in Fig. 4, where these models performed differently during the categorization of comparable types of issues. Aside from that, there is a reason why these five models were selected to form this ensemble model. We began by training and testing all of the models that had been shortlisted. Then, depending on their performance, we selected the top

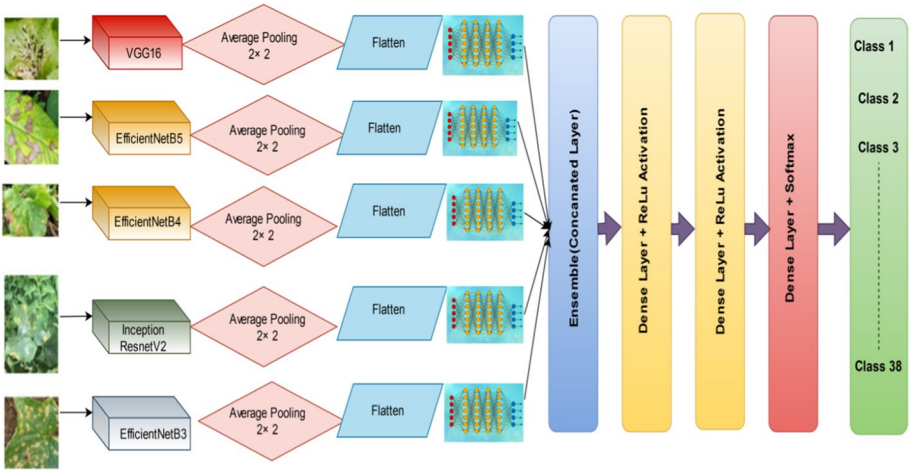


Fig. 4 Detailed representation of the implemented ensemble deep transfer learning model

five models to form an ensemble. We integrated all of the models and then applied two thick layers of 512 units each with a ReLU activation function. We then used a thick layer with a value of 38. We used the Softmax activation algorithm for this layer, and then we constructed the model. We trained one of the top models, calculated its weights, and then applied those weights to another model. We produced weights from another model and utilized those weights in another model. By doing so, we ultimately arrive at the conclusion that while in between; performance may suffer, passing those weights of the best model aids in sharpening slopes and increasing accuracy. We trained the model on a batch size of 16, utilizing Nadam as the optimizer for each step.

3.2.1 Proposed model

Even though many of the available methods (based on image processing or unsupervised learning) are great for figuring out how a disease is spreading in an area and figuring out where it is spreading, they aren't good at finding early signs because they need the illness to be in its later stages in order to see it in the image. Also, handcrafted feature-based algorithms produce accurate classification results, even though they have some problems, like the need for a lot of pre-processing and a long method. Nevertheless, the handcrafted-based approach limits feature extraction, and the resulting features may not be sufficient for effective identification, lowering accuracy. Deep learning, a family of machine learning algorithms, can tackle this difficulty, and various convolutions are performed in the Deep CNN's multiple layers [80]. They generate a variety of representations of the training outcomes, beginning with the most basic in the initial major layers and advancing to more complicated representations in the deeper layers. Firstly, the convolutional layers function as feature extractors on the training data, which is then decreased in dimension by the pooling layers. Convolutional layers synthesize extra discriminative features from a range of lower level features. The convolutional layers also serve as the foundation for the deep CNN. Feature architecture is a critical component of deep learning and represents a significant advancement over previous approaches to machine learning. Figure 5 depicts the suggested model's contributions to the various layers. The spatial component

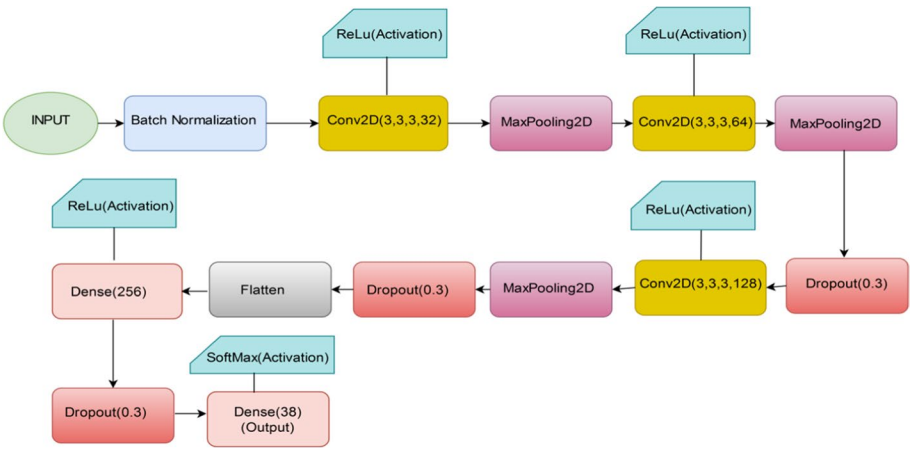


Fig. 5 Layered architecture of the proposed model (Built from Scratch)

of downsampling is performed by the pooling layer. It suggests lowering the amount of requirements. The suggested model's pooling layer was built utilizing the max pooling technique. Max pooling outperforms average pooling in the proposed deep CNN model, **PlaNet**. Another essential layer is dropout, which refers to the process of removing individuals from a network. It is a regularization approach for reducing overfitting. Ultimately, the thick layer classifies the data received from the convolutional and pooling layers.

A deep CNN will gradually extract higher level features from input images by using numerous layers in a model. As the proposed model, we created a 13-layer deep model PlaNet. This model consists mostly of three convolution layers, three Max Pooling layers, three dropouts, and two thick layers. Convolutional layers considerably improve the model's representational capability. Each convolution layer has strides of 3×3, with 32, 64, and 128 filters, respectively, whereas the MaxPooling layer has strides of 2×2 (refer to Fig. 5). The model consists of one batch normalization layer that normalizes the data, three Conv2D layers (the first with 32 filters, the second with 64 filters, and the third with 128 filters), three MaxPooling layers, and three Dropout layers with values of 0.3, 0.2, and 0.15, respectively. After that, we flattened the layer for full connectivity and then applied a dense layer with 256 neurons and an output dense layer with 38 neurons as our output classes.

The convolution layer makes a convolution kernel, which is then combined with the layer's input to make a tensor of outputs. Using a filter on an input to create activation is a simple way to do convolution. When the same filter is used on an object more than once, a "feature map" is made. This map shows where and how strong a detected feature is in an input, like a image. The suggested model contains three convolutional layers, C1, C2, and C3, with strides of 3×3, with C1 having 32 filters, C2 having 64 filters, and C3 having 128 filters. Each convolution layer has a 'ReLU' activation function. As the number of convolutional layers increases, so does the number of training parameters. PlaNet uses pooling layers to address this issue. The pooling layer reduces the total performance of the convolutional layer while keeping the image's features. Pooling layers enable down sampling of feature maps by enumerating the features present in patches of the feature map. Max pooling is used in the network, with strides of 2×2.

The neurons of a network layer connect the layers completely (densely). Each neuron in a layer receives feedback from all neurons in the preceding layer, suggesting that they are closely coupled. The dense layer is a completely interconnected layer, which means that every neuron in one layer is coupled to a neuron in the next one. The PlaNet features two dense layers, one with a 256-unit 'ReLU' activation function and another with a 38-unit 'Softmax' activation function. The activation functions ReLU and Softmax are used in the model's convolutional and fully connected layers in this work. The ReLU function can learn the parameters of a rectifier in an adaptive manner. The SoftMax function is used to convert CNN output into a probability distribution. To compute the difference between the prediction result and the label of the input samples, an average cross entropy loss function is used. The Adam algorithm is used to improve the loss function. ReLU is an activation unit that works in conjunction with other activation units such as tanh and sigmoid. The piecewise linear function, often known as ReLU, simply returns the same input value if it is greater than 0; otherwise, it returns 0. When an AI prediction model builds an output class, it removes as much of the misleading number as possible.

Dropout is the process of deactivating units in a neural network. When the network begins to practice, the learning phase is dominated by features with a higher representation in the outcomes, and the weights correlated with these features increase. During the training period, the Dropout layer randomly sets input units to 0 with a rate-dependent frequency, limiting overfitting. The PlaNet has three dropout layers with values of 0.3, 0.2,

and 0.15, respectively. This model PlaNet employs 'Adam' as an optimizer, with "sparse categorical cross entropy' as a loss, 'Accuracy' metrics, and an early stopping callback.

ReLU (Rectified linear unit) All convolutional network layers use the unsaturated ReLU activation since it outperforms other activation functions such as sigmoid and tanh. ReLU generates x as an output if the input, x, is positive; otherwise, it generates 0. Mathematically, the ReLU function is defined as follows:

$$R = \max(0, z) \tag{7}$$

$$R'(z) = \left\{ \begin{array}{l} 0 z < 0 \\ 1 \ z > 0 \end{array} \right\} \tag{8}$$

SoftMax The softmax function is used in the last Dense layer (the output layer with 38 neurons). It ranges from 0 to 1 giving the intermediate probability value. The SoftMax function is given by:

$$\sigma\left(\vec{z}\right) = \frac{e^{z_i}}{\sum_{j=1}^{k} e^{z_j}} \tag{9}$$

where, z = input vector; e = exponential function;K = number of classes. Thus we can conclude that we've built a large-scale neural network that is more succinct than others.

3.3 Hyperparameter tuning

This tuning is a useful machine learning technique to increase overall performance of classical neural networks. This optimization approach selects the optimal parameters representing the model architecture, also known as a hyperparameter. Hyperparameters are tuned by the full training phase, accuracy and required modifications (Tables 5, 6, 7, 8, 9 and 10). The model is updated to find the right mix to tackle the challenge. Classification techniques can be improved by tuning the hyperparameter [65]. Some typical hyperparameters are hidden layers number, learning rate, epochs, batch size, activation functions etc.

4 Experimental results and models evaluation

4.1 Experimental setup

We have designed tests to illustrate the resilience of each model, in which one combination of selected datasets will also be employed. Table 11 depicts the parameters of the experimental setting. During all the experimentation, we carried out three-fold cross-validation. The prime benefit of cross validation is that it makes use of all the data points in the dataset, which results in a low bias. The first dataset contains 38 classes, while second and third have 15 classes each. Because the 15 classes in each of the second and third datasets are disjoint, the fourth dataset, which is created by combining the second and third datasets, contains 30 classes (the combinations of datasets (1, 2) are already covered in the

Table 6 Hyperparameter values of the deep learning models implemented in this study

Parameters	Number of epoch	Optimizer	Number of Hidden Layers	Number of convolutional layers	Number of fully connected units	Dropout	Input Layer Size	Output Layer Size
VGG-16	30	Adam	5	13	10	0.25	250×250	38
VGG-19	30	Nadam	6	16	11	0.5	250×250	38
Xception	30	Nadam	5	36	26	0.25	250×250	38
InceptionResNetV2	30	Nadam	14	164	145	0.25	250×250	38
NASNetLarge	30	Softmax	13	142	134	0.5	224×224	38
Proposed model	30	Softmax	11	212	204	0.25	224×224	38
ResNet 101	30	Softmax	12	1000	1000	0.25	250×250	38

Table 7 Hyperparameter values of the deep learning models implemented in this study

Parameters	Number of units	Number of Layers	Activation Function	Filter	Pool size	Padding	Depth	Learning Rate
Ensemble	1000+	1000+	ReLU	250	4×4	1	1000+	0.001
MobileNet	28	28	ReLU	250	4×4	1	28	0.001

classes of dataset 1). Figure 6 shows process flow of the whole work and the Fig. 7 displays the complete layout and the step-by-step description of the methodology (in Experiment 1 and in other experiments too) to make the overall work more understandable. We have shown discrimination in 38 classes, but for experiments 2, 3, and 4, the figure can be altered according to classification classes (15, 15, and 30 respectively). The models of diverse CNNs are trained, and 4 experiments are conducted on 3 datasets; the results of various efficiency measuring parameters are obtained in Tables 12, 13, 14 and 15. Table 16 is compiled to depict the average performance of each model on each of the efficiency measuring metrics. From Table 16, Table 17 was made so that the average wise top five models for each metric could be found.

4.2 Performance indicators, evaluation metrics and generated results

Due to the fact that dataset 1 contains 38 types (other two with 15 classes each), multi-class classification isperformed. The metrics are measured using indices such as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). Traditionally, true positives (TP) refer to instances in which we predict that yes (subject has an infection) and they really do have the infection. True negatives (TN) indicate that we correctly predicted it and the subject is not infected. In the case of false positives (FP), we predict that the subject has the disease but does not currently have it. False negatives (misdetections): we anticipated them to be virus-free, yet they are. TP (correct detection) is the number of correctly classified assigned photos in each group, whereas TN is the count of correctly categorized images in all categories excluding the corresponding category. The value FN represents the number of misclassified photos in the chosen category. Except for the area in question, the FP contains the number of misclassified images in all other sections [51].

Accuracy is an often used metric, which is the degree to which a computer algorithm approaches the true value of an expected value. However, due to the accuracy fallacy, relying solely on accuracy may be deceptive. Accuracy is therefore used in conjunction with other performance measures, such as precision, recall, F1-score, and AUC, to test algorithms. All of the calculations above are dependent on the confusion matrix. These metrics are used to equate the proposed approach with the more advanced approaches to this situation [51]. Accuracy does not take into account the distribution of results. As a result, the F-measure is used to accurately manage delivery issues. It is advantageous when the dataset includes classes of imbalance. It is often introduced as a means of harmonic accuracy and recall. F1 scores indicate the proportion of instances classified correctly by the learning model [90].

Accuracy is used in conjunction with other performance indicators, such as precision, recall, F1-score, and AUC, to test deep learning algorithms. All of the calculations above are dependent on the uncertainty matrix. Precision is used to assess the classifiers'

Table 8 Hyperparameter values of the deep learning models implemented in this study

Parameters	Number of epoch	Optimizer	Number of Hidden Layers	Number of convolutional layers	Number of fully connected units	Dropout	Input Layer Size	Output Layer Size
Ensemble	30	Adam	1000+	1000+	1000+	0.25	250×250	38
MobileNet	30	Adam	25	25	24	0.25	224×224	38

Table 9 Hyperparameter values of the deep learning models implemented in this study

Parameters	Number of units	Number of Layers	Activation Function	Filter	Pool size	Padding	Depth	Optimizer
EfficentNetB7	813	813	ReLU	250	4×4	1	813	Nadam
InceptionV3	48	48	ReLU	224	4×4	1	48	Nadam

accuracy. A low precision value means that the classifier has a high number of false positives. Recall, also known as sensitivity, is a metric used to determine the completeness of a classifier. A lower recall value means that the classifier is having difficulty with higher FP values [81]. Specificity refers to the percentage of true negatives that are graded accurately as such.

$$Sensitivity(or\ Recall) = \frac{\text{True positives}}{\text{True Positives} + \text{False Negatives}} \tag{10}$$

$$Specificity = \frac{\text{True negative}}{\text{True Negatives} + \text{False positives}} \tag{11}$$

$$Precision = \frac{\text{True positives}}{\text{True Positives} + \text{False Positives}} \tag{12}$$

$$Accuracy = \frac{\text{True positives} + \text{True Negatives}}{\text{True Positives} + \text{False Positives} + \text{False Negatives} + \text{True Negatives}} \tag{13}$$

$$F1 - Score = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \tag{14}$$

The proposed model achieves a classification accuracy upto98.51%, and the effect of the false positive and false negative rates is shown in Appendix section using a confusion matrix. It demonstrates unequivocally that the suggested solution results in a lower incidence of false negatives and positives. Thus, the suggested model could be the strongest solution for identifying plant leaf infection. The bolded values in Tables 5, 6, 7, 8 and 9 indicate the situation in which the best value was obtained for the related success criterion.

In addition, a P-test was performed to confirm the computed results (refer to Tables 12, 13, 14 and 15). The P-Test is utilized to compare the test performance of one machine learning model to that of a second model. DeLong's test can be used to determine whether one model has a significantly different AUC than another. DeLong et al. [24] define a method for determining the empirical AUC. Consider that Efficient B5 has an AUC of 0.99956 for predicting glaucoma illness, whereas Efficient B4 has an AUC of 0.9725. With $p < 0.05$, DeLong's test may establish that Efficient B5 has a significantly different AUC from Efficient B4 [24]. The empirical AUC correlates to Mann-Whitney U-statistic. The Mann-Whitney statistics evaluate the probability that a randomly selected observation from the healthy population is less than or equal to a randomly selected observation from the plant-affected group. A nonparametric null test claims that a randomly picked value from one group is equally likely to be less than or greater than a randomly selected value from the other group.

Table 10 Hyperparameter values of the deep learning models implemented in this study

Parameters	Learning Rate	Number of epoch	Number of Hidden Layers	No. of convolutional layers	Number of fully connected units	Drop out	Input Layer Size	Output Layer Size
EfficentNetB7	0.001	30	813	813	813	0.25	224×224	38
InceptionV3	0.001	30	48	48	48	0.5	224×224	38

Table 11 Details of the requisite experimental setup

Experiment Number	Dataset Used to Train the model(s)	Dataset used for Testing	Total Number of Images (Training + Testing)	Number of Classes
1.	(1)	(1)	87,900	38
2.	(2)	(2)	20,738	15
3.	(3)	(3)	5549	15
4.	(2,3)	(2,3)	26,287	30

The results of the models used in Experiments 1, 2, 3, and 4 are shown in Tables 12 through 15. In experiment 1 (on dataset 1), the ensemble model and EfficientNetB5 get the best AUC and accuracy score, respectively, while PlaNet gets the best F1-score. Moreover, the Ensemble model performs the best for recall and specificity, and PlaNet generates the best precision value. The least accuracy generated in this experiment was 76%broadly (ignoring decimal values) by InceptionV3, but the remaining models' accuracy was greater than 78% (ignoring decimal values) and up to 98% (ignoring decimal values). In terms of AUC, the minimum score was 74% (Inception V3), and the overall performance range is 74% to 97%. For F1-scores, the minimum score was 75% (Inception V3), and the overall performance range is 75% to 98%. Ensemble model presents spectacular outcomes during experiment 2 (on dataset 2) and computes the best AUC and the F1 score while PlaNet (the proposed model) computes the best accuracy. The least accuracy generated in this experiment was 74% by InceptionV3, but the remaining models' accuracy was greater than 78% and up to 97%. In terms of AUC, the minimum score was 75% (InceptionV3), and the overall performance range was 75% to 98%. The minimum score for accuracy was 74% in the case of InceptionV3, and the range of performance was from 74% to 97%. For this experiment, the best recall value was

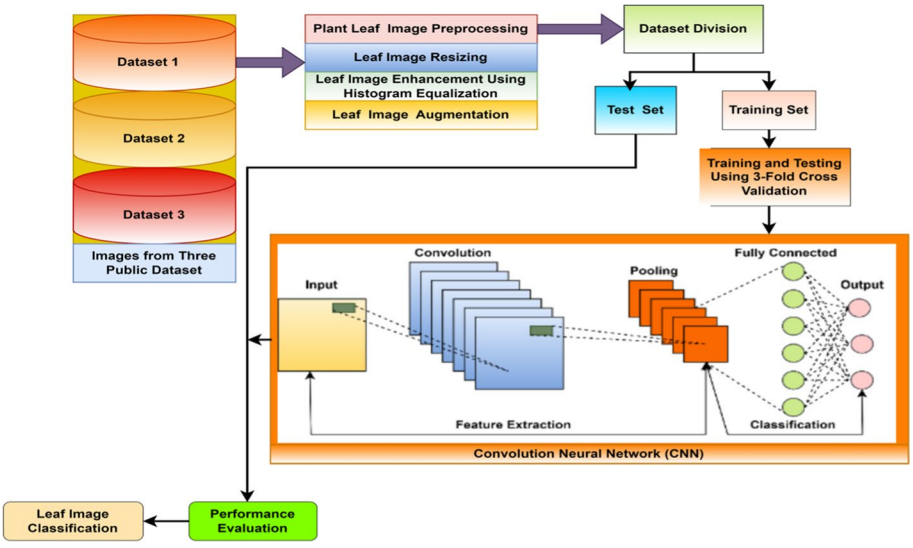


Fig. 6 The process flow diagram of the whole implemented work

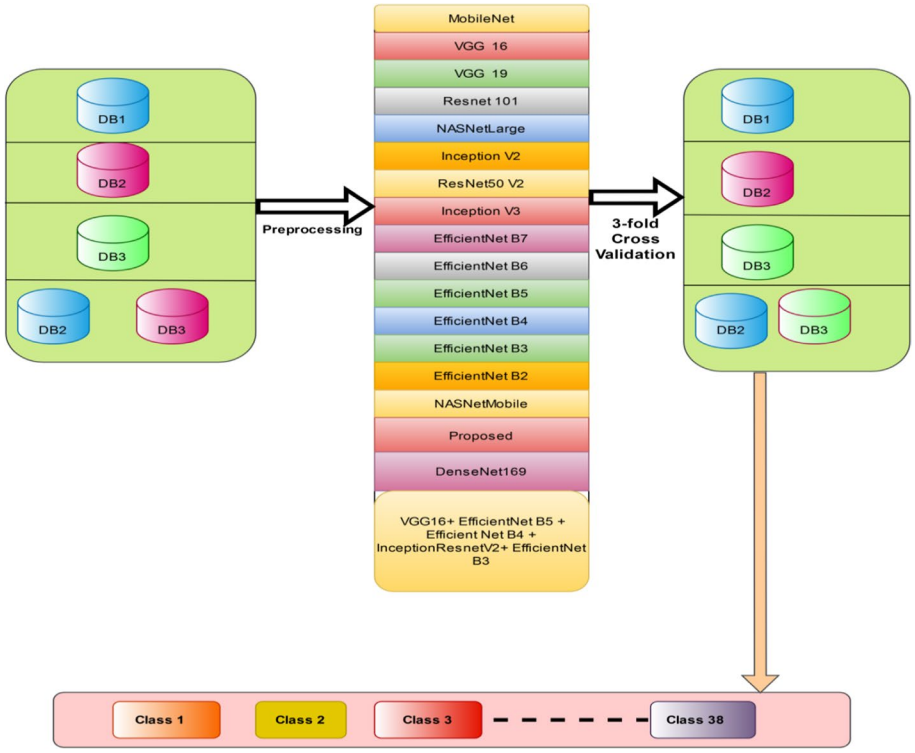


Fig. 7 Layout of the complete process executed in this study for Plant Disease recognition

generated by the PlaNet model and the least by the VGG-19 model. Similarly, the best performer for precision was EfficientNetB5 and the least show was presented by InceptionV3, and for specificity, the best performer was the PlaNet model and the weakest show was presented by InceptionV3.

In experiment 3 (performed on dataset 3), EfficientNetB2 computes the best accuracy, and the best AUC score is created by the Ensemble model. However, PlaNet performs the best in terms of F1-score. The least accuracy generated in this experiment was by InceptionV3 (73%), but the remaining models' accuracy was above 78% and up to 98%. In terms of AUC, the minimum score was 72% (Inception V3), and the overall performance range was 78% to 98%. The PlaNet model provided the best recall value for this trial, whereas the InceptionV3 model generated the lowest recall value. Ensemble was the greatest performer for precision, while InceptionV3 showed the least, and the EfficientNetB2 model was the best performer for specificity, while InceptionV3 showed the least.

During the last experiment (on datasets 2 and 3), the ensemble model computed the highest accuracy. The best AUC score was created by PlaNet, while the Ensemble model performed the best in terms of F1-score. The minimum accuracy generated in this experiment by VGG-19 is 79% broadly (ignoring decimal values), whereas the remaining models' accuracy was above 79%(ignoring decimal values) and up to 97%(ignoring decimal values). In terms of AUC, the lowest score was 79% (Inception V3), and the overall performance range was 79% to 95%. In terms of accuracy, VGG-19 got at least

Table 12 Results of different Models on the Dataset 1(**Experiment 1**)

Experiment −1 Image nets	Metric −1 F1 Score	Metric-2 AUC	Metric-3 Accuracy	Metric-4 Recall/sensitivity	Metric-5 Precision	Metric-6 Specificity	Metric-7 *P value*
VGG-16	0.8808	0.8498	0.8456	0.8510	0.8611	0.8531	0.00693
VGG-19	0.8395	0.7536	0.7856	0.7721	0.7602	0.7533	0.00673
MobileNet	0.8715	0.9054	0.8956	0.8865	0.8623	0.8732	0.36812
InceptionResNetV2	0.9001	0.8526	0.9125	0.8705	0.9052	0.8723	0.08913
InceptionV3	0.7541	0.7412	0.7623	0.7078	0.6999	0.7165	0.0927
ResNet-101	0.9045	0.9012	0.8535	0.8342	0.8498	0.8664	0.36812
ResNet50V2	0.8257	0.8158	0.8695	0.8534	0.8301	0.8263	0.00693
Xception	0.8923	0.8985	0.8562	0.8626	0.8442	0.8342	0.28914
VGG16+ InceptionResNet-V2 + EfficientNetB5+ EfficientNetB4+ EfficientNetB3 (Ensemble Model)	0.9701	**0.9799**	0.9856	**0.9823**	0.9545	**0.9612**	0.3747
EfficentNetB7	0.8424	0.8342	0.8669	0.8626	0.8709	0.8324	0.09727
EfficientNetB6	0.8545	0.8695	0.8456	0.8332	0.8571	0.8359	0.09103
EfficientNetB5	0.9545	0.9785	**0.9857**	0.9809	0.9490	0.9234	0.3737
EfficientNetB4	0.9574	0.9593	0.9725	0.9768	0.9779	0.9325	0.07186
EfficientNetB3	0.9658	0.8851	0.9256	0.9077	0.9156	0.8968	0.00693
EfficientNetB2	0.9658	0.9396	0.8847	0.8568	0.9023	0.9078	0.36812
DenseNet169	0.8795	0.8896	0.9045	0.8679	0.9061	0.8798	0.00695
Proposed Model	**0.9887**	0.9521	0.9851	0.9766	**0.9790**	0.9491	0.36819
NASNet Large	0.8565	0.8958	0.9145	0.9067	0.9185	0.8971	0.09492

The best value of the parameter is shown in bold

Table 13 Results of different Models on the Dataset 2(**Experiment 2**)

Experiment −2 Image nets	Metric −1 F1 Score	Metric-2 AUC	Metric-3 Accuracy	Metric-4 Recall/sensitivity	Metric-5 Precision	Metric-6 Specificity	Metric-7 *P value*
VGG-16	0.8432	0.9085	0.8659	0.8569	0.9032	0.8451	0.3747
VGG-19	0.8390	0.8432	0.7856	0.7523	0.7845	0.8427	0.08729
MobileNet	0.8415	0.8594	0.9025	0.8905	0.9123	0.8456	0.00691
InceptionResNetV2	0.9201	0.8797	0.9262	0.9451	0.8923	0.8761	0.07182
InceptionV3	0.7441	0.7515	0.7485	0.7589	0.7376	0.7612	0.00692
ResNet-101	0.7645	0.7896	0.8235	0.7858	0.7709	0.7654	0.00715
ResNet50V2	0.8957	0.9287	0.8956	0.8865	0.9345	0.8845	0.35757
Xception	0.9023	0.8929	0.9025	0.9207	0.8907	0.8989	0.00767
VGG16+ InceptionResNet-V2 + EfficientNetB5+ EfficientNetB4+ EfficientNetB3(**Ensemble Model**)	**0.9689**	**0.9812**	0.9687	0.9578	0.9612	0.9536	0.36809
EfficentNetB7	0.8524	0.9058	0.8596	0.8569	0.9190	0.9023	0.36801
EfficientNetB6	0.8948	0.8469	0.8956	0.8790	0.9045	0.8432	0.0721
EfficientNetB5	0.9259	0.9586	0.9658	0.9547	**0.9820**	0.9204	0.36828
EfficientNetB4	0.9623	0.9193	0.9475	0.9342	0.9789	0.9233	0.08627
EfficientNetB3	0.9147	0.9091	0.9256	0.9067	0.9235	0.8990	0.07184
EfficientNetB2	0.8647	0.8396	0.8741	0.8678	0.9056	0.8876	0.08821
DenseNet169	0.8896	0.9196	0.9369	0.9267	0.9123	0.9009	0.34812
Proposed Model	0.9645	0.9586	**0.9751**	**0.9689**	0.9768	**0.9234**	0.36828
NASNet Large	0.8964	0.9371	0.9025	0.8923	0.9429	0.8812	0.36728

The best value of the parameter is shown in bold

Table 14 Results of different Models on the Dataset 3(**Experiment 3**)

Experiment −3 Image nets	Metric −1 F1 Score	Metric-2 AUC	Metric-3 Accuracy	Metric-4 Recall/ sensitivity	Metric-5 Precision	Metric-6 Specificity	Metric-7 *P value*
VGG-16	0.8461	0.8468	0.8956	0.8456	0.8809	0.8345	0.00683
VGG-19	0.8759	0.8356	0.8597	0.8437	0.8124	0.8679	0.08527
MobileNet	0.8689	0.8945	0.8745	0.8654	0.8769	0.8354	0.09481
InceptionResNetV2	0.8755	0.8664	0.9025	0.8970	0.8812	0.9023	0.07394
InceptionV3	0.7732	0.7212	0.7369	0.7023	0.7342	0.7234	0.00613
ResNet-101	0.8564	0.8102	0.8025	0.8435	0.8154	0.8209	0.07101
ResNet50V2	0.8753	0.8583	0.9078	0.9034	0.8679	0.8567	0.00623
Xception	0.9154	0.9098	0.8894	0.8823	0.8934	0.8812	0.36821
VGG16+ InceptionResNet-V2 + EfficientNetB5+ EfficientNetB4+ EfficientNetB3 (**Ensemble Model**)	0.9654	**0.9899**	0.9397	0.9276	**0.9576**	0.9234	0.36823
EfficentNetB7	0.8433	0.8493	0.8956	0.8954	0.8456	0.8341	0.08627
EfficientNetB6	0.8433	0.8097	0.8456	0.8343	0.8109	0.7980	0.00692
EfficientNetB5	0.8563	0.8957	0.8569	0.8600	0.9150	0.8932	0.36721
EfficientNetB4	0.9625	0.9653	0.8995	0.8935	0.9453	0.9436	0.36741
EfficientNetB3	0.8487	0.8231	0.8759	0.8265	0.8669	0.8112	0.0066
EfficientNetB2	0.9447	0.9783	**0.9885**	0.9546	0.9431	**0.9781**	0.37886
DenseNet169	0.8125	0.7961	0.7859	0.7721	0.8254	0.7904	0.00763
Proposed Model	**0.9656**	0.9878	0.9872	**0.9740**	0.9328	0.9235	0.36812
NASNet Large	0.8826	0.8993	0.9256	0.9359	0.9012	0.9331	0.08709

The best value of the parameter is shown in bold

Table 15 Results of different Models on the on the Combination of Dataset (2+3)(**Experiment 4**)

Experiment −4 Image nets	Metric −1 F1- Score	Metric-2 AUC	Metric-3 Accuracy	Metric-4 Recall/sensitivity	Metric-5 Precision	Metric-6 Specificity	Metric-4 *P value*
VGG-16	0.8889	0.8585	0.8695	0.8512	0.8708	0.8643	0.007186
VGG-19	0.8145	0.7956	0.7958	0.8185	0.8067	0.8145	0.00697
MobileNet	0.9124	0.9149	0.9582	0.9511	0.9603	0.9262	0.08809
InceptionResNetV2	0.9278	0.8565	0.9125	0.8565	0.9132	0.8576	0.07394
InceptionV3	0.8025	0.7912	0.8498	0.7988	0.8563	0.7957	0.00691
ResNet-101	0.8875	0.9075	0.8887	0.9075	0.8934	0.8965	**0.3647**
ResNet50V2	0.7825	0.7883	0.7969	0.7891	0.7864	0.7942	0.00678
Xception	0.8654	0.8589	0.8668	0.8454	0.8612	0.8534	0.08943
VGG16+ InceptionResNet-V2+Efficient-NetB5+ EfficientNetB4+ EfficientNetB3 (**Ensemble Model**)	**0.9589**	0.9499	**0.9776**	**0.9654**	0.9321	0.9442	**0.34722**
EfficentNetB7	0.8041	0.8423	0.8234	0.8361	0.8512	0.8121	0.08609
EfficientNetB6	0.9036	0.8946	0.8943	0.8864	0.8534	0.8581	0.09482
EfficientNetB5	0.9502	0.9487	0.9703	0.9634	**0.9643**	0.9574	0.08819
EfficientNetB4	0.9105	0.8931	0.8899	0.8756	0.8834	0.9078	0.0947
EfficientNetB3	0.8956	0.9012	0.9455	0.9309	0.9488	0.9151	0.00692
EfficientNetB2	0.9457	0.9439	0.8948	0.8901	0.9109	0.8831	0.3562
DenseNet169	0.8768	0.8699	0.8745	0.8612	0.8653	0.8621	0.007125
Proposed Model	0.9556	**0.9526**	0.9704	0.9636	0.9421	**0.9547**	0.36521
NASNet Large	0.8554	0.8691	0.8545	0.8536	0.8621	0.8424	0.09125

The best value of the parameter is shown in bold

Table 16 Average-wise performance comparison of all the models

ALL EXPERIMENTS AVERAGE	F1	AUC	Accuracy	Recall/sensitivity	Precision	Specificity	*P value*
VGG-16	0.8647	0.8659	0.8692	0.8512	0.8791	0.8493	0.09470
VGG-19	0.8422	0.8070	0.8066	0.7966	0.7909	0.8196	0.00767
MobileNet	0.8735	0.8935	0.9077	0.9059	0.9029	0.8701	0.09736
InceptionResNetV2	0.9058	0.8738	0.9134	0.8922	0.8979	0.8771	0.08427
InceptionV3	0.7684	0.7512	0.7742	0.7419	0.7575	0.7492	0.00691
ResNet-101	0.8532	0.8521	0.8430	0.8427	0.8323	0.8273	0.090124
ResNet50V2	0.8450	0.8477	0.8674	0.8581	0.8547	0.8529	0.07028
Xception	0.8938	0.8900	0.8787	0.8775	0.8723	0.8669	0.08481
VGG16 + InceptionResNet-V2 + EfficientNetB5+ EfficientNetB4 + EfficientNetB3(**Ensemble Model**)	0.9658	**0.9752**	0.9679	0.9582	0.9513	**0.9456**	0.3731
EfficentNetB7	0.8305	0.8579	0.8564	0.8553	0.8741	0.8452	0.08834
EfficientNetB6	0.8740	0.8551	0.8702	0.8582	0.8564	0.8338	0.07304
EfficientNetB5	0.9217	0.9453	0.9446	0.9347	0.9514	0.9112	0.35157
EfficientNetB4	0.9482	0.9342	0.9273	0.9200	0.9463	0.9268	0.08405
EfficientNetB3	0.9062	0.8796	0.9182	0.8929	0.9137	0.8805	0.00681
EfficientNetB2	0.9302	0.9253	0.9105	0.8923	0.8946	0.9141	0.36701
Proposed Model	**0.9686**	0.9627	**0.9795**	**0.9707**	**0.9576**	0.9376	0.3512
DenseNet169	0.8646	0.8688	0.8754	0.8569	0.8772	0.8583	0.06702
NASNet Large	0.8727	0.9055	0.8992	0.9022	0.9061	0.8859	0.09381

The best value of the parameter is shown in bold

Table 17 Best 5 performing models (on average basis)

Ranking of Models	On the basis of Accuracy	On the basis of AUC	On the basis of F1-SCORE	On the basis of Recall/sensitivity	On the basis of Precision	On the basis of Specificity
First	**Proposed Model(PlaNet)**	(**Ensemble Model**)	**Proposed Model(PlaNet)**	**Proposed Model(PlaNet)**	Proposed Model(**PlaNet**)	(**Ensemble Model**)
Second	(**Ensemble Model**)	Proposed Model(**PlaNet**)	(**Ensemble Model**)	(**Ensemble Model**)	EfficientNetB5	Proposed Model(**PlaNet**)
Third	EfficientNetB5	EfficientNetB5	EfficientNetB4	EfficientNetB5	(Ensemble Model)	EfficientNetB4
Fourth	EfficientNetB4	EfficientNetB2	EfficientNetB2	EfficientNetB4	EfficientNetB4	EfficientNetB2
Fifth	EfficientNetB3	EfficientNetB4	EfficientNetB5	MobileNet	NASNet Large	EfficientNetB5

The best value of the parameter is shown in bold

a 79% score, and performance ranged from 79% to 97% overall. In this experiment, the Ensemble model produced the best recall value, whereas Inception V3 produced the worst. Similarly, the best performer for precision was EfficientNetB5, and the least show was presented by ResNet50V2, and for specificity, the best performer was the PlaNet model, and the weakest show was presented by ResNet50V2.

Finally, we created Table 16 (by extracting information from Tables 12, 13, 14 and 15) to illustrate the average performance of all the models on all of the relevant parameters over all experiments. PlaNet computes the highest average accuracy in this situation. The best AUC score was obtained by the ensemble model. However, in the case of the F1-score, PlaNet does the best average-wise. The average minimum accuracy obtained by Inception V3 is 77%. However, the remaining models' accuracy was above 80% and up to 97%. In terms of AUC, the minimum score was 75% (Inception V3), and the overall performance range is 75% to 97%. In terms of the F1-score, InceptionV3 generated the lowest possible score, and the whole range of performance is between 76% and 96%. For precision computation, range varies from 75% (InceptionV3) to 95% (PlaNet) Next, the ranges of computed values for the recall metrics vary from 74% (InceptionV3) to 97% (PlaNet). Finally, for specificity, the range lies between 74% (InceptionV3) and 94% (Ensemble). Based on all of these experiments and a lot of talk, we can say that the deep learning-based technique works well for classifying plant diseases into multiple groups. The results are amazing, with the exception of two or three models that don't do so well.

4.3 Computation time and environment

In the proposed effort, Google Colaboratory is utilized to train and evaluate the models. The GPU's runtime has been selected. The average computation time for VGG-16 was 47 minutes, for VGG-19 it was 3 hours and 29 minutes, for MobileNet it was 2 hours and 30 minutes, for InceptionResNetV2 it was 1 hour and 35 minutes, for InceptionV3 it was 1 hour and 10 minutes, for Xception it was 2 hours and 20 minutes, for EfficientNetB3 it was 1 hour and 20 minutes, for EfficientNetB2 it was 1 hour and 35 minutes, for DenseNetLarg Locally, the experiments were carried out on a machine equipped with an Intel core i5-8520U (8Gen) processor running at 1.60-I.80GHz, 8GB RAM, and a 64-bit Windows operating system. The programming language used for the implementation is Python-3.6.

Few graphs, displaying validation_loss, training_loss, validation_accuracy, and training_accuracy, for six models (out of 17 implemented models) implemented during the first experiment are demonstrated using Fig. 8. Remaining models for the first experiment and all models' graphical performance during the second, third, and fourth experiments are not shown due to space constraints. As we increase the number of epochs in the validation phase, the training and testing losses become almost identical, indicating that the model is sufficiently well suited. We found that the accuracy of training and validation are almost the same, which shows that there is no overfitting. Also, the accuracy is getting better with each epoch, which shows that the model is working well.

Based on intense trials on a large dataset, the results, and in-depth data analysis, we discovered that our suggested "**PlaNet**" model worked well in Plant Disease identification, producing superior performance metrics such as accuracy and F1-score than existing state-of-the-art approaches.

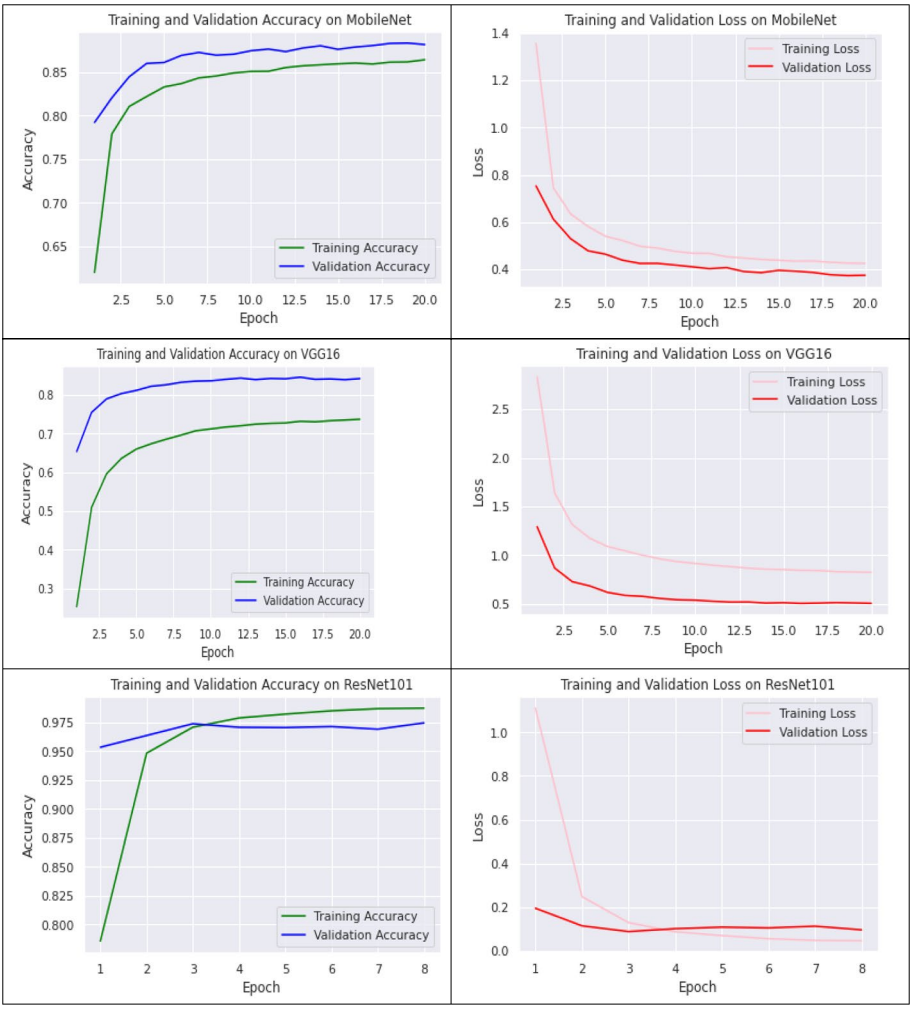


Fig. 8 Generated Training loss, Validation loss, Training accuracy and Validation accuracy curves of 7 models during first experiment

5 Comparison with prior works

In order to compare the work performed in this study with the 21 most recent and relevant studies published in Scopus and/or SCI indexed journals during the last 30 months (2019 (July)-to date). Relevance is measured in terms of studies where leaf disease classification is implemented using deep learning-based architectures. The brief comparative discussion is hereby presented, but a detailed comparative Table 18 is also drawn for ready reference, shown below after the discussion. In the case of Geetharamani and Pandian [29], the authors have selected the same dataset that we have worked upon. The number of classes and the generated results are comparable in both cases. The number of models implemented and tested in our work is far larger than in [29]. Moreover, we

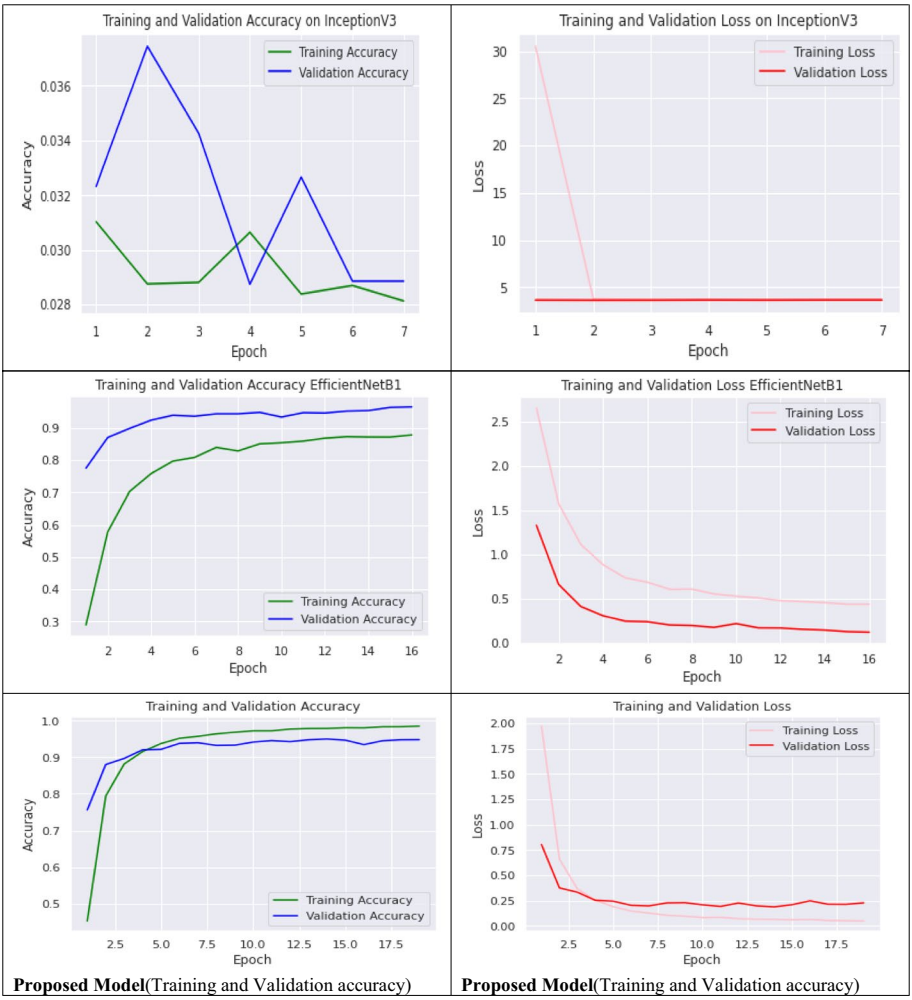


Fig. 8 (continued)

have worked on two more datasets and their combinations, and we have also performed well in these other three cases too, which proves the superiority of our model not only in one experiment but also in the other three experiments too. The same argument is also applicable to Atila et al. [7]. We admit that the accuracy of [7] is better than ours, but other equivalent efficacy measuring parameters, AUC and F-Score, are not mentioned in [7], so a fair comparison cannot be made on these two parameters. In the case of Agarwal et al. [3], practitioners selected leaves of one specific crop from a public dataset for disease identification and classification. When we compare our work with that of [3], we deduce that our work is implemented in a more detailed manner because the count of subject images, the number of classes for image classification, and the number of models implemented, the number of experiments conducted, and finally, the number of subject datasets are larger, so our results are more secure and generic and our models are more useful for the agriculture sector. The same argument is also applicable when we

Table 18 Comparison of the results with state-of-the-are CNN methods

Year of Publication	Reference	Num. Of Experiments Performed	Whether subject is Public dataset	Number of classes for classification	Max. Accuracy (%)	Max. AUC score	Max. F1-Score	Approximate Number of original images i.e., before augmentation	Number of Models implemented in the work (best performing model)
2019	Coulibaly et al. [21]	1	Customized	2	95	NA	0.9175	124	1(VGG-16)
2020	Tiwari [91]	1	Yes	30	95.58	NA	0.96	340	2(CNN)
2020	Geetharamaniet al. [29]	1	Yes	38	96.46	NA	0.9815	54,305	5(DeepCNN)
2020	Agarwal et al. [3]	2	Yes	10	98.4	NA	NA	18,160	4(Proposed CNN)
2020	Agarwal et al. [4]	1	Yes	8	93.75	0.9357	NA	1000	4(Proposed CNN)
2020	Waheed et al. [95]	1	No	4	98.06	NA	0.98	12,332	5(Optimized DenseNet)
2020	Saleem et al. [71]	1	Yes	38	73.07	NA	NA	54,305	3(Single Shot MultiBox Detector_Inception_V2)
2020	Karlekar& Seal [46]	1	Yes	16	98.14	NA	0.97	486	10(Proposed one named as SoyNet)
2020	Chen et al. [16]	2	Yes	Upto 05	92	NA	NA	1000	5(Proposed one named as INC-VGGN)
2020	Chen et al. [17]	2	Yes	16	99.81	NA	NA	1000	8(Proposed one named as MobileNet-Beta)
2020	Jadhav et al. [41]	2	No	4	98.75	NA	NA	Upto 650	2(AlexNet)

Table 18 (continued)

Year of Publication	Reference	Num. Of Experiments Performed	Whether subject is Public dataset	Number of classes for classification	Max. Accuracy (%)	Max. AUC score	Max. F1-Score	Approximate Number of original images i.e., before augmentation	Number of Models implemented in the work (best performing model)
2020	Bi et al. [12]	1	No	3	77.65	NA	NA	334	3(MobileNet)
2021	Atilaet al. [7]	1	Yes	38	99.47	NA	NA	55,448	12(EfficientNetB5)
2021	Shah et al. [74]	1	Yes	38	NA	NA	0.991	54,305	Self Crafted Deep model ResTS
2021	Joshi et al. [44]	1	No	3	97.4	0.9758	NA	433	6(Proposed VirLeafNet-3)
2021	Gokulnath& Devi [30]	1	Yes	4	98.93	NA	0.9665	7500	5(A loss-fusion based proposed resilient CNN)
2021	Tiwari et al. [92]	1	Yes	27	99.9	NA	0.933	25,493	4(Proposed Dense CNN)
2021	Sujatha et al. [83]	1	No	5	89.5	0.97	0.895	609	3(VGG-16)
2021	Deepalaksh-miet al. [23]	1	Yes	2	94.5	NA	NA	600	1(CNN)
2022	Syed-Ab-Rahmanet al. [86]	1	Yes	4	86.18 to 97.2	NA	0.90 to 0.98	600	Self Crafted Deep model
	Ours/Max	4	Yes	38	98.85	0.9899	0.9887	More than 54,000	18(PlaNet)
	Ours /Avg.	4	Yes	Maximum 38 classes, Minimum 15 classes	97.95	0.9752	0.9686	Maximum 54,305, Minimum 5208	18(PlaNet)

compare our work with that of Agarwal et al. [4]. Furthermore, our results are superior to those obtained in [4] and consistent with those obtained in [3]. In the case of Saleem et al. [71], the same dataset has been selected, which has been our first dataset on which we have implemented our first experiment. Saleem et al. [71] describe the efficiency of their model in terms of mean average precision (MAP), while we have shown our results in terms of AUC, F1-Score and accuracy. If we compare our accuracy (or F-Score or AUC) with MAP, we find our work and results in a more comfortable zone to justify the superiority of our work. When we compare (Jadhav et al. [61], Deepalakshmi et al. [23], Tiwari [91], Karlekar and Seal [46], and Chen et al. [16], we find that our implemented work is more passionate than these four works in terms of the number of subject datasets, subject images, generated results, classification classes, and models implemented. We know that the larger the size of the dataset provided to imagenets, the more generic and accurate the model will be. We have worked on more than 54,000 images and they have selected no more than 1200 images. Thus, our work is more generic and the performance is not dataset dependent. In the case of [46], the authors have focused on a particular crop only, while we have sample images of multiple crops. Chen et al. [17] presented a high accuracy, better than our average, to show that their new imagenet worked, but other parts of their experiment, like the number of subject images and the fact that they only used one dataset, were less impressive than ours. The same argument is also employed for comparison with Chen et al. 2020 [18], where the results are no better than ours and the image count is even lower than in Chen et al. 2020 [17]. Presenters (Waheed et al. [95]) have worked on private collected images of corn leaf images for performance evaluation of their approach, in which 12,332 images were classified into 4 classes. Our work is more general as we experiment on a larger number of images (datasets and implemented models too) and the number of classes is far greater in our case. However, the accuracy of their independently performed experiments (in our case, four experiments) is in line with ours. To show the comparison in a table format (refer Table 18), a detailed comparison table is made. It can be observed that because the majority of the literature does not use all of the data in the PlantVillage dataset, the number of classifications is significantly smaller than the dataset's 38 classes. This work involves the use of all plant varieties for the tests and involves the model teaching of a total of 38 groups. The suggested solution has a remarkably high accuracy score, which indicates excellent success compared to other state-of-the-art techniques. Although some of the individual pictures are incorrectly classified, the approach proposed correctly identifies most of the diseases and results in high accuracy for multi-experiments in the unseen pictures, indicating that the approach proposed, PlaNet, can identify significant plant conditions.

Based on the results of experiments, it can be assumed that the proposed method works well for identifying types of plant disease and can be used in other fields, like online fault diagnosis, target recognition, etc. The primary reason for the overwhelming response to the suggested solution (**PlaNet**) is that it leverages not only the generalized characteristics of images retrieved from ImageNet via pre-trained networks, but also the experimental dataset's special features through the newly expanded layers. Thus, the proposed solution achieves the best results in experiments using public datasets since it completely trains the network parameters and avoids the overfitting issue. In other areas, **PlaNet** shows promise and surpasses existing CNN designs, which are the cutting-edge machine learning techniques for image identification, by meeting the demand for an efficient model that can be trained quickly without sacrificing performance.

6 Practical applicability and conclusion

Digital pictures are used to find out what's wrong with plants and put them into groups so that they can produce more. Additionally, the kinds of crops and the causes of rodents, illnesses, and weeds must be determined. This study shows that the deep learning model for finding plant diseases doesn't need supervision, is very accurate, is widely used, and is very successful. But there are many things that make it hard to accurately diagnose plant diseases in different environments. A precise model of recognition could help make things work, which would solve the problem of ecosystem complexity. Deep learning techniques, and CNNs in particular, have become popular. They are used in software for analyzing images and recognizing patterns, and they have helped find leaf diseases in plants. This work creates a recognition model that combines the computer vision algorithm and the transfer learning algorithm to solve these problems and improve the identification process. This makes it possible to identify plant diseases in a complex setting. In addition to responding to complicated scenarios, the model improves recognition accuracy. The prospective goal of this research is to expand its scope to include additional plant species and groupings. This would allow us to forecast more accurately and respond to more complex issues. On the basis of a novel CNN architecture, a bespoke deep learning model (**PlaNet**) is developed for detecting plant illnesses and categorizing photos using standard, publicly available, and well-recognized image datasets. CNN divided leaf pictures into 38 groups of healthy and non-healthy plants in the first trial; 15 classes in the second and third studies; and 30 classes in the fourth experiment (the second and third experiments were combined to classify images into 30 groups). The addition of data augmentation strategies balances the samples in each class. Data augmentation is utilized to increase the amount of relevant data. Thus, it contributed to the generalization of the suggested model. In addition, the coefficient of determination is computed using the AUC, precision, and F1 values. In all, 18 models were investigated for this investigation, and **PlaNet** exhibits superior performance over other influential CNN models almost all the time. The average accuracy of the suggested model is 97.95%, which is almost higher than all other competitive state-of-the-art classifiers. Our experiment and comparison with other deep architectures reveal that our strongest deep learning-focused model is able to distinguish between various plant illnesses. Strong trees, fertile soil, and the absence of external pests and illnesses all contribute to a healthy climate. Our proposed technique would contribute significantly to agricultural research. From the comparison analysis, it can be concluded that **PlaNet** fared better than the common CNN models used in this article. In addition, the approach suggested in this work employs an image as an input to CNN and self-learns to attain a high rate of recognition. Consequently, our technology outperforms conventional deep learning models.

Based on deep learning models, this study shows a new way to find and categorize plant diseases. In this article, the author suggests a quick, automated, and less expensive model that not only increases the power of the neural network compared to the standard model, but also reduces the number and consistency of the neural network's specifications in the dataset and improves performance. The main goal of this study was to come up with a way for farm manufacturers to quickly and meaningfully classify the position and type of plant diseases and to suggest that they take steps to prevent diseases. In addition, this research improves and increases the precision of the existing theory. At the same time, it is crucial in terms of environmental complexity when studying plant disease detection, and it allows researchers to reflect on the crucial role environmental complexity plays in

the identification of plant diseases. Deep CNN outperforms other machine learning models in terms of predicted accuracy, F1-Score, and AUC. If an effective deep learning strategy could be implemented, it would be advantageous to reduce the need for fungicide treatments. Thus, the paradigm is linked to agricultural production knowledge technology and is favorable to the evolution of intelligent agriculture in a sustainable manner. This research may also be valuable for several robotic systems that identify and differentiate between healthy and unhealthy crops in real-time. The proposed model may be utilized as a decision-making aid to assist growers and farmers in recognizing and classifying certain diseases. This will lead to the automation of agriculture. The proposed approach defined all plant classes, but only a small number of plant classes achieved comparatively lower accuracy. Therefore, with a few small changes, the network can also be improved for better fidelity. Farmers may use this technology to automate the classification of tree diseases since the picture is so precise. This saves both time and money since the need for experts may be minimized in certain circumstances. We could also use a simple web app to implement the suggested model, making it easy for people with no experience to use our method. This technique is especially valuable for farmers in distant places where plant pathologists are few.

7 Limitations of the study

The main problem with the proposed system is that the model has not been tested in a variety of situations, such as lighting, occlusion, lighting changes, etc. In some situations, the system may not work as well as it could, but this can be fixed by training the model with pictures taken in different situations. The authors know that putting the proposed method to use on field data from the real world could make it less accurate [58]. In the future, we will focus on the difficulties of such an experiment and the changes that need to be made to the model. In addition, we aim to extensively enhance the data to boost the model's robustness. The possibility of many concurrent issues in the plant's leaves warrants more research. The next limitation is that a real-world application should be able to categorize photos of a disease as it emerges on a plant, despite the fact that these are basic conditions. In fact, many diseases manifest themselves not only on the upper side of leaves (or at all), but also on several other plant parts. Consequently, future image collection operations should aim to gather images from a range of perspectives, preferably in as realistic a setting as possible. It is critical to emphasize that the presented strategy is meant to supplement rather than replace current methods of disease diagnosis. Laboratory testing is always more reliable than diagnoses based just on visual symptoms, and an early-stage diagnosis based solely on a visual examination may be challenging. Clearly, there are still several challenges to overcome, and certain problems remain unresolved. This indicates that, for the time being, devices for the automated detection of plant diseases can only offer a highly educated estimate, allowing users to take prompt action, especially in the absence of expert technical assistance. As technology progresses, some of these limitations may be eliminated, but there is still plenty to explore. In addition, the integration of the proposed system with sensors and mobile devices (for field testing) must be evaluated. The experimental findings demonstrated that the suggested approach performed effectively on both publicly available datasets. It is necessary to deploy and test mobile devices and sensors in order to automatically monitor and detect a vast array of plant disease data. In the meantime, we aim to apply it to more real-world settings. Also, feature visualization and multi-objective

hyperparameter settings are hot topics in deep learning, and they could be used to get more accurate results that would help us understand plant diseases better.

8 Future work

By adding classification algorithms and marking training data with pictures of different diseases, this method can be made even better so that it can automatically diagnose different diseases. Additional newly proposed CNN models can be applied to plants that will assist farmers in developing a completely automatic and self-training disease detection method. One can gather new photographs from multiple sources of various plant types, regional locations, leaf growth, cultivation environments, picture qualities, and modes in order to expand the collection capacity. The enhanced dataset, covering a wider range of plant species and more groups of leaf diseases, will help the model achieve better performance and accuracy. The bigger aim of the functional work would be to expand our plant disease detection goals from leaves to other areas of the plants (flowers, fruits, and stems). We may expand this concept to better identify plant diseases. Improved models can help plant pathologists and farmers detect and deal with plant diseases more rapidly in mobile environments. One goal of our research has been to provide a basic infection detection approach that may be quickly adjusted for online use in the field. Specific sensors for practical usage may be created in the future based on these findings. These sensors must be durable, reasonably priced, and user-friendly. This study may be extended to other crops and biological realms, including X-ray pictures, CT-scan images of the brain, etc., to diagnose illnesses with image classification and support patients with low cost, better and quick outcomes.

Data availability The datasets generated during and/or analysed during the current study are available in the internet repository, please refer (https://github.com/spMohanty/PlantVillage-Dataset/tree/master/raw, https://www.kaggle.com/emmarex/plantdisease) and (https://www.kaggle.com/vbookshelf/v2-plant-seedlings-dataset) of this paper for datasets.

Declarations

Conflict of interest The authors declare that neither associations nor perceived conflicts of interest nor competing financial interests exist in this paper.

References

1. Abbas A, Jain S, Gour M, Vankudothu S (2021) Tomato plant disease detection using transfer learning with C-GAN synthetic images. Comput Electron Agric 187:106279
2. Agarwal N, Singhai J, Agarwal DK (2017) Grape Leaf Disease Detection and Classification Using Multi- Class Support Vector Machine. Proceeding of IEEE International conference on Recent Innovations in Signal Processing and Embedded Systems (RISE), 238–244. https://doi.org/10.1109/RISE.2017.8378160
3. Agarwal M, Gupta SK, Biswas KK (2020) Development of Efficient CNN model for Tomato crop disease identification. Sustain Comput: Inform Syst 28:100407
4. Agarwal M, Gupta S, Biswas KK (2020) A new Conv2D model with modified ReLU activation function for identification of disease type and severity in cucumber plant. Sustain Comput: Inform Syst 30:100473
5. Ardakani AA, Kanafi AR, Acharya UR, Khadem N, Mohammadi A (2020) Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks. Comput Biol Med 121:103795

6. Arribas JI, Sánchez-Ferrero GV, Ruiz-Ruiz G, Gómez-Gil J (2011) Leaf classification in sunflower crops by computer vision and neural networks. Comput Electron Agric 78(1):9–18
7. Atila Ü, Uçar M, Akyol K, Uçar E (2021) Plant leaf disease classification using efficientnet deep learning model. Ecol Inform 61:101182
8. Babu MP, Rao BS (2007) Leaves recognition using back propagation neural network-advice for pest and disease control on crops. IndiaKisan. Net: Expert Advisory System, 607–626
9. Badage A (2018) Crop disease detection using machine learning: Indian agriculture. Int Res J Eng Technol 5(09)
10. Bansal P, Kumar R, Kumar S (2021) Disease detection in Apple leaves using deep convolutional neural network. Agriculture 11(7):617
11. Barbedo JG (2018) Factors influencing the use of deep learning for plant disease recognition. Biosyst Eng 172:84–91
12. Bi C, Wang J, Duan Y, Fu B, Kang JR, Shi Y (2020) Mobilenet based apple leaf diseases identification. Mob Netw Appl:1–9
13. Brahimi M, Boukhalfa K, Moussaoui A (2017) Deep learning for tomato diseases: classification and symptoms visualization. Appl Artif Intell 31(4):299–315
14. Calderón R, Navas-Cortés JA, Zarco-Tejada PJ (2015) Early detection and quantification of Verticillium wilt in olive using hyperspectral and thermal imagery over large areas. Remote Sens 7(5):5584–5610
15. Camargo A, Smith JS (2009) An image-processing based algorithm to automatically identify plant disease visual symptoms. Biosyst Eng 102(1):9–21
16. Chen J, Chen J, Zhang D, Sun Y, Nanehkaran YA (2020) Using deep transfer learning for image-based plant disease identification. Comput Electron Agric 173:105393
17. Chen J, Zhang D, Nanehkaran YA (2020) Identifying plant diseases using deep transfer learning and enhanced lightweight network. Multimed Tools Appl 79(41):31497–31515
18. Chen J, Zhang D, Nanehkaran YA, Li D (2020) Detection of rice plant diseases based on deep transfer learning. J Sci Food Agric 100(7):3246–3256
19. Chollet F (2017) Xception: Deep learning with depthwise separable convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1251–1258
20. Chouhan SS, Singh UP, Sharma U, Jain S (2021) Leaf disease segmentation and classification of JatrophaCurcas L. and PongamiaPinnata L. biofuel plants using computer vision based approaches. Measurement 171:108796
21. Coulibaly S, Kamsu-Foguem B, Kamissoko D, Traore D (2019) Deep neural networks with transfer learning in millet crop images. Comput Ind 108:115–120
22. Cseke L, Podila G, Kirakosyan A, Kaufman P (2009) Plants as Sources of Energy. In: Recent Advances in Plant Biotechnology. Chapter., 9, pp 163–210. https://doi.org/10.1007/978-1-4419-0194-1_9
23. Deepalakshmi P, Lavanya K, Srinivasu PN (2021) Plant Leaf Disease Detection Using CNN Algorithm. Int J Inf Syst Model Des (IJISMD) 12(1):1–21
24. DeLong ER, DeLong DM, Clarke-Pearson DL (1988) Comparing the areas under two or more correlated receiver operating characteristic curves: a nonparametric approach. Biometrics 44:837–845
25. Du JX, Zhai CM, Wang QP (2013) Recognition of plant leaf image based on fractal dimension features. Neurocomputing 116:150–156
26. Ferentinos KP (2018) Deep learning models for plant disease detection and diagnosis. Comput Electron Agric 145:311–318
27. Fuentes A, Yoon S, Kim SC, Park DS (2017) A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. Sensors 17(9):2022
28. Fuentes A, Yoon S, Park DS (2020, February) Deep Learning-Based Techniques for Plant Diseases Recognition in Real-Field Scenarios. In International Conference on Advanced Concepts for Intelligent Vision Systems. Springer, Cham, pp. 3–14
29. Geetharamani G, Pandian A (2020) Identification of plant leaf diseases using a nine-layer deep convolutional neural network. Comput Electr Eng 76:323–338
30. Gokulnath BV, Devi GU (2021) Identifying and classifying plant disease using resilient LF-CNN. Ecol Inform 63:101283
31. Grinblat GL, Uzal LC, Larese MG, Granitto PM (2016) Deep learning for plant identification using vein morphological patterns. Comput Electron Agric 127:418–424
32. Guo Y, Han S, Li Y, Zhang C, Bai Y (2018) K-Nearest Neighbor combined with guided filter for hyperspectral image classification. Procedia Comput Sci 129:159–165
33. Hamuda E, Glavin M, Jones E (2016) A survey of image processing techniques for plant extraction and segmentation in the field. Comput Electron Agric 125:184–199

34. Hassan SM, Maji AK, Jasiński M, Leonowicz Z, Jasińska E (2021) Identification of plant-leaf diseases using CNN and transfer-learning approach. Electronics 10(12):1388
35. Hassanien AE, Gaber T, Mokhtar U, Hefny H (2017) An improved moth flame optimization algorithm based on rough sets for tomato diseases detection. Comput Electron Agric 136:86–96
36. 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
37. Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, ..., Adam H (2017) Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861
38. Hu G, Yang X, Zhang Y, Wan M (2019) Identification of tea leaf diseases by using an improved deep convolutional neural network. Sustain Comput: Inform Syst 24:100353
39. Huang G, Liu S, Van der Maaten L, Weinberger KQ (2018) Condensenet: An efficient densenet using learned group convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2752–2761
40. Islam M, Dinh A, Wahid K (2017) Detection of potato Diseases Using Image Segmentation and Multi-class Support Vector Machine. IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE), 1–4
41. Jadhav SB, Udupi VR, Patil SB (2020) Identification of plant diseases using convolutional neural networks. Int J Inf Technol:1–10
42. Jiang Z, Dong Z, Jiang W, Yang Y (2021) Recognition of rice leaf diseases and wheat leaf diseases based on multi-task deep transfer learning. Comput Electron Agric 186:106184
43. Johannes A, Picon A, Alvarez-Gila A, Echazarra J, Rodriguez-Vaamonde S, Navajas AD, Ortiz-Barredo A (2017) Automatic plant disease diagnosis using mobile capture devices, applied on a wheat use case. Comput Electron Agric 138:200–209
44. Joshi RC, Kaushik M, Dutta MK, Srivastava A, Choudhary N (2021) VirLeafNet: Automatic analysis and viral disease diagnosis using deep-learning in Vignamungo plant. Ecological Informatics 61:101197
45. Kamilaris A, Prenafeta-Boldú FX (2018) Deep learning in agriculture: A survey. Comput Electron Agric 147:70–90
46. Karlekar A, Seal A (2020) SoyNet: Soybean leaf diseases classification. Comput Electron Agric 172:105342
47. Kaur P, Pannu HS, Malhi AK (2019) Plant disease recognition using fractional-order Zernike moments and SVM classifier. Neural Comput & Applic 31(12):8749–8768
48. Khamparia A, Singh A, Luhach AK, Pandey B, Pandey DK (2020) Classification and identification of primitive Kharif crops using supervised deep convolutional networks. Sustain Comput: Inform Syst 28:100340
49. Khan MA, Akram T, Sharif M, Awais M, Javed K, Ali H, Saba T (2018) CCDF: Automatic system for segmentation and recognition of fruit crops diseases based on correlation coefficient and deep CNN features. Comput Electron Agric 155:220–236
50. Khan MA, Akram T, Sharif M, Javed K, Raza M, Saba T (2020) An automated system for cucumber leaf diseased spot detection and classification using improved saliency method and deep features selection. Multimed Tools Appl 79:18627–18656
51. Khanna M, Agarwal A, Singh LK, Thawkar S, Khanna A, Gupta D (2021) Radiologist-level two novel and robust automated computer-aided prediction models for early detection of COVID-19 infection from chest X-ray images. Arab J Sci Eng:1–33
52. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Adv Neural Inf Proces Syst 25:1097–1105
53. Lee SH, Chan CS, Wilkin P, Remagnino P (2015, September) Deep-plant: Plant identification with convolutional neural networks. In: 2015 IEEE international conference on image processing (ICIP), IEEE, pp. 452–456
54. Ma J, Du K, Zheng F, Zhang L, Gong Z, Sun Z (2018) A recognition method for cucumber diseases using leaf symptom images based on deep convolutional neural network. Comput Electron Agric 154:18–24
55. Majumdar D, Ghosh A, Kole D, Chakraborty A, Majumder D (2014) Application of Fuzzy C-Means Clustering Method to Classify Wheat Leaf Images Based on the Presence of Rust Disease. Advances in Intelligent Systems and Computing, 327. Advance online publication. https://doi.org/10.1007/978-3-319-11933-5_30
56. Mittal M, Goyal LM, Kaur S, Kaur I, Verma A, Hemanth DJ (2019) Deep learning based enhanced tumor segmentation approach for MR brain images. Appl Soft Comput 78:346–354

57. Mittal A, Kumar D, Mittal M, Saba T, Abunadi I, Rehman A, Roy S (2020) Detecting pneumonia using convolutions and dynamic capsule routing for chest X-ray images. Sensors 20(4):1068
58. Mohanty SP, Hughes DP, Salathé M (2016) Using deep learning for image-based plant disease detection. Front Plant Sci 7:1419
59. Mokhtar U, Ali MA, Hassanien AE, Hefny H (2015) Identifying two of tomatoes leaf viruses using support vector machine. In: Information Systems Design and Intelligent Applications. Springer, New Delhi, pp 771–782
60. Mostafa AM, Kumar SA, Meraj T, Rauf HT, Alnuaim AA, Alkhayyal MA (2021) Guava Disease Detection Using Deep Convolutional Neural Networks: A Case Study of Guava Plants. Appl Sci 12(1):239
61. Mukhopadhyay S, Paul M, Pal R, De D (2021) Tea leaf disease detection using multi-objective image segmentation. Multimed Tools Appl 80(1):753–771
62. Murase H (2000) Artificial intelligence in agriculture. Comput Electron Agric 29(1/2):1–2
63. Mustafa MS, Husin Z, Tan WK, Mavi MF, Farook RSM (2020) Development of automated hybrid intelligent system for herbs plant classification and early herbs plant disease detection. Neural Comput & Applic 32(15):11419–11441
64. Nagaraju M, Chawla P, Upadhyay S, Tiwari R (2022) Convolution network model based leaf disease detection using augmentation techniques. Expert Syst 39(4):e12885
65. Palaniswamy SK, Venkatesan R (2021) Hyperparameters tuning of ensemble model for software effort estimation. J Ambient Intell Humaniz Comput 12(6):6579–6589
66. Pawar R, Jadhav A (2018) Pomogranite disease detection and classification. IEEE International Conference on Power, Control, Signals and Instrumentation (ICPCSI)
67. Priya CA, Balasaravanan T, Thanamani AS(2012) An efficient leaf recognition algorithm for plant classification using support vector machine. In: International conference on pattern recognition, informatics and medical engineering (PRIME-2012).IEEE. pp.428–432
68. Rahimzadeh M, Attar A (2020) A modified deep convolutional neural network for detecting COVID-19 and pneumonia from chest X-ray images based on the concatenation of Xception and ResNet50V2. Inform Med Unlocked 19:100360
69. Rangarajan AK, Purushothaman R, Ramesh A (2018) Tomato crop disease classification using pre-trained deep learning algorithm. Procedia Comput Sci 133:1040–1047
70. Rumpf T, Mahlein AK, Steiner U, Oerke EC, Dehne HW, Plümer L (2010) Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance. Comput Electron Agric 74(1):91–99
71. Saleem MH, Khanchi S, Potgieter J, Arif KM (2020) Image-Based Plant Disease Identification by Deep Learning Meta-Architectures. Plants 9(11):1451
72. Sambasivam G, Opiyo GD (2021) A predictive machine learning application in agriculture: Cassava disease detection and classification with imbalanced dataset using convolutional neural networks. Egypt Inform J 22(1):27–34
73. Sankaran S, Mishra A, Ehsani R, Davis C (2010) A review of advanced techniques for detecting plant diseases. Comput Electron Agric 72(1):1–13
74. Shah D, Trivedi V, Sheth V, Shah A, Chauhan U (2021) ResTS: Residual deep interpretable architecture for plant disease detection. Inf Process Agric 9:212–223
75. Sharif M, Khan MA, Iqbal Z, Azam MF, Lali MIU, Javed MY (2018) Detection and classification of citrus diseases in agriculture based on optimized weighted segmentation and feature selection. Comput Electron Agric 150:220–234
76. Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556
77. Singh & Misra (2017) Detection of plant leaf diseases using image segmentation and soft computing techniques. Inf Process Agric 4(1):41–49
78. Singh S, Hoiem D, Forsyth D (2016) Swapout: Learning an ensemble of deep architectures. arXiv preprint arXiv:1605.06465
79. Singh T, Kumar K, Bedi SS (2021) A Review on Artificial Intelligence Techniques for Disease Recognition in Plants. IOP Conf Ser Mater Sci Eng 1022(1):012032 IOP Publishing
80. Singh LK, Garg H, Khanna M (2022) Deep learning system applicability for rapid glaucoma prediction from fundus images across various data sets. Evol Syst 13:1–30
81. Singh LK, Garg H, Khanna M (2022) Performance evaluation of various deep learning based models for effective glaucoma evaluation using optical coherence tomography images. Multimed Tools Appl 81:27737–27781
82. Strange R, Scott P (2005) Plant Disease: A Threat to Global Food Security. Annu Rev Phytopathol 43(1):83–116. https://doi.org/10.1146/annurev.phyto.43.113004.133839 PMID:16078878

83. Sujatha R, Chatterjee JM, Jhanjhi NZ, Brohi SN (2021) Performance of deep learning vs machine learning in plant leaf disease detection. Microprocess Microsyst 80:103615
84. Sukegawa S, Yoshii K, Hara T, Yamashita K, Nakano K, Yamamoto N, … Furuki Y (2020) Deep neural networks for dental implant system classification. Biomolecules 10(7):984
85. Sun H, Xu H, Liu B, He D, He J, Zhang H, Geng N (2021) MEAN-SSD: A novel real-time detector for apple leaf diseases using improved light-weight convolutional neural networks. Comput Electron Agric 189:106379
86. Syed-Ab-Rahman SF, Hesamian MH, Prasad M (2022) Citrus disease detection and classification using end-to-end anchor-based deep learning model. Appl Intell 52(1):927–938
87. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, ..., Rabinovich A (2015) Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 1–9
88. Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z (2016) Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2818–2826
89. Tan M, Le Q (2019) Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning. PMLR, pp. 6105–6114
90. Thawkar S, Sharma S, Khanna M, Kumar Singh L (2021) Breast cancer prediction using a hybrid method based on Butterfly Optimization Algorithm and Ant Lion Optimizer. Comput Biol Med 139:104968
91. Tiwari S (2020) A comparative study of deep learning models with handcraft features and non-handcraft features for automatic plant species identification. International Journal of Agricultural and Environmental Information Systems (IJAEIS) 11(2):44–57
92. Tiwari V, Joshi RC, Dutta MK (2021) Dense convolutional neural networks based multiclass plant disease detection and classification using leaf images. Ecol Inform 63:101289
93. Tiwari V, Joshi RC, Dutta MK (2022) Deep neural network for multi-class classification of medicinal plant leaves. Expert Syst 39:e13041
94. Vishnoi VK, Kumar K, Kumar B (2021) Plant disease detection using computational intelligence and image processing. J Plant Dis Prot 128(1):19–53
95. Waheed A, Goyal M, Gupta D, Khanna A, Hassanien AE, Pandey HM (2020) An optimized dense convolutional neural network model for disease recognition and classification in corn leaf. Comput Electron Agric 175:105456
96. Wang G, Sun Y, Wang J (2017) Automatic image-based plant disease severity estimation using deep learning. Comput Intell Neurosci 2017:1–8
97. Wetterich CB, de Oliveira Neves RF, Belasque J, Ehsani R, Marcassa LG (2017) Detection of Huanglongbing in Florida using fluorescence imaging spectroscopy and machine-learning methods. Appl Opt 56(1):15–23
98. Yang X, Guo T (2017) Machine learning in plant disease research. Eur J BioMed Res 3(1):6–9
99. Yang CC, Prasher SO, Enright P, Madramootoo C, Burgess M, Goel PK, Callum I (2003) Application of decision tree technology for image classification using remote sensing data. Agric Syst 76(3):1101–1117
100. Zhang S, Wu X, You Z, Zhang L (2017) Leaf image based cucumber disease recognition using sparse representation classification. Comput Electron Agric 134:135–141
101. Zhang X, Qiao Y, Meng F, Fan C, Zhang M (2018) Identification of maize leaf diseases using improved deep convolutional neural networks. IEEE Access 6:30370–30377
102. Zhang S, Zhang S, Zhang C, Wang X, Shi Y (2019) Cucumber leaf disease identification with global pooling dilated convolutional neural network. Comput Electron Agric 162:422–430
103. Zhang YD, Dong Z, Chen X, Jia W, Du S, Muhammad K, Wang SH (2019) Image based fruit category classification by 13-layer deep convolutional neural network and data augmentation. Multimed Tools Appl 78(3):3613–3632
104. Zoph B, Vasudevan V, Shlens J, Le QV (2018) Learning transferable architectures for scalable image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 8697–8710

Authors and Affiliations

Munish Khanna[1] · Law Kumar Singh[2] · Shankar Thawkar[3] · Mayur Goyal[1]

Law Kumar Singh
lawkumarcs@gmail.com

Shankar Thawkar
shankar.thawkar@gmail.com

Mayur Goyal
mayurgoyal.mg@gmail.com

[1] Department of Computer Science and Engineering, Hindustan College of Science and Technology, Mathura 281122, India

[2] Department of Computer Engineering and Application, GLA University, Mathura, India

[3] Department of Information Technology, Hindustan College of Science and Technology, Mathura 281122, India