International Journal of Computer Applications (0975 - 8887)
Volume 186 - No.78, April 2025

Convolutional Neural Network-based Xception, MobileNetV2 and InceptionV3 Models for Plant Disease Identification in Sub-Saharan Africa

Aminou Halidou
Department of Computer Science,
University of Yaounde I, Cameroon

Youssoufa Mohamadou
University Institute of Technology,
University of Ngaoundere, Cameroon

Pascalin Tiam Apen
University Institute of Technology Fotso Victor,
University of Dschang, Cameroon

Daramy Vandi Von Kallon
Department of Mechanical and Industrial Engineering,
University of Johannesburg, South Africa

William John Baraza
African Advanced Level Telecommunications Institute

Mbouna Gildas Patrick
Department of Computer Science,
University of Yaounde I, Cameroon

Djiembou Tientcheu Victor Nico
Department of Computer Science,
University of Yaounde I, Cameroon

Robndoh Mardochée
Department of Computer Science,
University of Yaounde I, Cameroon

ABSTRACT

Plant disease identification in Sub-Saharan Africa poses a significant challenge, hindered by costly laboratory tests or subjective visual assessments. Recent advances in image-based disease identification show promise, but existing methods are limited in accuracy and efficiency. This study addresses these shortcomings by presenting a convolutional neural network (CNN)-based plant disease classifier, leveraging transfer learning from pre-trained models Xception, MobileNetV2, and InceptionV3. A high generalization rate of 98.76% is achieved in the test data, demonstrating the potential for efficient and accurate identification of plant disease. This research contributes to innovative agricultural management solutions in Sub-Saharan Africa, with implications for improving crop yields, food security, and sustainable agriculture.

General Terms

Plant Disease Detection, Deep Learning Classification

Keywords

Plant diseases, machine learning, deep learning, convolutional neural networks, transfer learning, GAP

1. INTRODUCTION

Agriculture is the main economic sector in the Central African Economic and Monetary Community (CEMAC) region, providing livelihoods to more than 50% of the population [1]. However, agricultural productivity is hindered by various factors, including biotic constraints such as plant diseases, which cause significant economic losses [17]. Plant diseases are changes caused by bacteria, fungi, or viruses in plants that result in substantial financial losses for the agricultural sector. Traditional methods of identifying plant diseases in Sub-Saharan Africa, including costly laboratory tests and visual assessments, are often inaccurate and time-consuming [9]. In contrast, imaging analysis techniques have become increasingly popular in developed countries for the identification of plant diseases, offering non-destructive and real-time results [14]. Recent advances in image processing and machine learning have led to the development of various approaches to detect and classify plant diseases [2, 10, 6, 8]. Deep learning techniques, particularly convolutional neural networks (CNNs), have shown great promise in image classification tasks, including the identification of plant diseases. However, more accurate and efficient plant disease classification systems are needed. This study aims is to address the need by developing a CNN-based plant disease classifier using a database of images. The objective is to design a robust and accurate classifier capable of distinguishing between various plant diseases, thus providing a valuable tool for agricultural management in CEMAC countries.

1.1 Problem Statement

The problem addressed in this study is an image classification problem, in which the goal is to classify images of plants into different disease categories. The development of an accurate and efficient plant disease classification system has significant implications for improving agricultural productivity, reducing economic losses, and improving food security in the CEMAC region.

1.2 Objectives

The main objectives of this study are: Develop a CNN-based plant disease classifier using a database of images. To evaluate the performance of the proposed classifier using various metrics. Compare the performance of the proposed classifier with the existing methods.

1.3 Methodology

The approach encompasses the following steps:
Review of the literature: A comprehensive review of existing studies on the identification of plant diseases using image processing and machine learning techniques will be carried out.
Data Collection: A robust database of images depicting various plant diseases will be assembled.
Data Preprocessing: The collected images will undergo preprocessing to enhance their quality and eliminate noise.
Model Development: A convolutional neural network (CNN)-based classifier for plant diseases will be developed utilizing the preprocessed images.
Model Evaluation: The performance of the proposed classifier will be rigorously assessed using a variety of evaluation metrics.

2. RELATED WORK

Al-Hiary et al. focused on the detection and classification of plant diseases using 500 images of 54 healthy and unhealthy plant leaves divided into 5 classes (diseases are early blight, cottony mold, ash mold, late blight, and tiny whiteness) in the AlGhor region of Jordan using machine learning [2]. They achieved an accuracy ranging from 83% to 94% using the k-means method for image segmentation and ANN (Artificial Neural Networks) for classification. They also extracted features using the GLCM (Gray Level Cooccurrence Matrix) method to calculate the features of pixels located only within the boundaries of the infected areas of the leaf. The system generates an independent color space transformation and a color transformation structure for the RGB sheet image from a given image, It then uses the K-means clustering methodology to group photos to identify the green pixels and compute the threshold values for these pixels using Otsu's method. Using cluster features as a basis, the K-means clustering method aimed to group objects, in this case, pixels. The categorization process involved reducing the total squares of the distances between the items and the centroid of the relevant class or group. GLCM technique was used to extract texture information from diseased leaves segmented. In some works, patterns of pairs of pixels divided by a distance d in a direction θ were identified by the cooccurrence matrices [11]. Typically, multiples of 45° with $d = 1$ and θ = (0°, 45°, 90°, 135°, ...) were considered. The size of the matrix was $N \times N$, with N being the maximum value of the gray levels. They created a matrix $\varphi(d, \theta)$ for every pair (d, θ). After that, an average matrix was computed to make it rotation-invariant. The angular momentum, moment of production, sum and difference of entropy, information measures of correlation, contrast, and correlation were used to calculate the characteristics. The affected cluster's RGB format was changed to HS. Only pixels included within the borders of the unhealthy portions of the leaf were used to determine the characteristics (healthy areas were deleted); the NN model was trained using the collection of training characteristics. The connection weights were constantly adjusted during the training phase until they met the predetermined number of iterations or an acceptable error. To demonstrate the validity of the model between the network's inputs and output, the Mean Squared Error (MSE) criterion was employed to guarantee the ANN model's ability to respond appropriately. The precision of the trained neural network model was confirmed using a set of characteristics of the test data.

Hossain et al. [10] developed a machine learning-based approach to identify and categorize plant diseases. They collected 277 images of healthy and diseased plant leaves from Reddit and Arkansas, two large plant database websites. To overcome the limitations of human visualization techniques, they employed machine learning and image processing techniques. The proposed method involved image acquisition, segmentation, and classification. The images were converted from RGB to Lab* color space, which comprises an "L*" luminosity layer and "a*" and "b*" chromaticity layers. The average color of each sample region was determined in the "ab" space. For segmentation, the classifier with k closest neighbors with three neighbors was used. The segmented parts were represented by green, red, and blue colors for the leaf, disease, and bottom portion, respectively. To separate the infected area from the original leaf image, a morphological opening was applied, followed by dilation and erosion. Six features were extracted from the segmented portion of the disease, including GLCM and color features. The extracted features included mean, standard deviation, energy, contrast, homogeneity, and correlation. The KNN classifier was used to categorize plant diseases into five groups. The classifier provided a list of the k closest data points to assign a class to new unlabeled samples. This approach demonstrated the effectiveness of machine learning and image processing techniques for the identification and classification of plant diseases.

DeChant et al. [6] employed deep learning techniques to automatically identify maize plants affected by fire blight from field photographs. They captured 1,834 photos of NLB-infected and uninfected leaves between 28 and 78 days after inoculation (DPI) using a Canon EOS Rebel or Sony a6000 camera. After disqualifying 38 low-quality photos, they categorized the remaining images according to the presence of lesions. The authors utilized the Bisque image processing platform to annotate visible lesions with a line along their main axis. They randomly split the dataset into training sets (70.%), validation sets (15%), and testing sets (15%), consisting of 78 photos of uninfected leaves and 10,000 images of infected leaves. The test set was used only for evaluating the final system's performance. To identify diseased leaves, DeChant et al. developed a three-stage analysis procedure. Initially, they trained convolutional neural networks (CNNs) to detect lesions in small leaf patches. In the second stage, these CNNs generated heat maps that indicated the likelihood of infection in different areas of the image. Finally, they used these heat maps to classify the images. The authors employed the Adam optimization algorithm in networks A, C, and the Stage 3 network, while using RMSprop in network B. Training the first-stage network took approximately three days using an NVIDIA Titan X GPU, while training the third-stage network took thirty minutes. At runtime, generating a heatmap for a single image required about two minutes, while classifying three heatmaps took less than a second. The proposed method achieved a 96.7% classification rate, 96.8% accuracy, 10% error rate, 97.4% recall, and an F1 score of 0.971. These results demonstrate the effectiveness of the convolutional neural network (CNN) computing pipeline for automated disease identification. A confusion matrix displayed the distribution of errors across the test set, highlighting the importance of the Step 3 classifier in aggregating local segment scores into a final classification.
Francis et al. [8] investigated the use of convolutional neural networks (CNNs) to detect and classify plant leaf diseases using the PlantVillage dataset. The dataset comprises 54,306 images, divided

into 14 species and 38 classes. For their experiment, they utilized 200 images per class, resulting in 7,600 loaded images, which were divided into 6,800 training images and 800 testing images. The authors pre-processed the images to suitable dimensions and applied data augmentation techniques, including rotation, zoom, height adjustment, and width shift. The CNN architecture consisted of multiple conv2D layers with 32, 64, and 128 filters, followed by a fully connected layer with 1024 filters. The features were extracted from the images using the pooling and conv2D layers, and the probability distribution of the neural network output was obtained using the Dense layer (38, activation="softmax"). To avoid overfitting, the authors employed Dropout, a regularization technique that randomly sets input units to zero during training. The ReLU activation function was applied to introduce non-linearity into the network. The model's "flatten" layer was used to combine feature maps into a single, one-dimensional vector. Testing was performed on Google Colab with an integrated GPU: Tesla K80. The convolutional neural network algorithm for the detection and classification of plant disease was implemented in Python. They achieved a 99. 89% classification rate using 90% of the data for training and 10% to test the model. The model was trained over 200 epochs. In another study, Chen et al. [4] combined deep convolutional neural networks (CNNs) with transfer learning for the identification of plant diseases. They fine-tuned the pre-trained VGGNet model and combined it with Inception modules to improve multiscale feature extraction. The model was trained on images of maize and rice leaves, taken in complex environments, and achieved an average precision of 92 00% to predict rice diseases under real-world conditions. Rani et al. [15] employed transfer learning using ResNet, VGG-16, EfficientNet, and Inception models, initially trained on ImageNet, to classify plants. The models achieved a maximum accuracy of 100% in the Sunflower dataset, 97.35% in the ColiFlower dataset, and 94.31% in the Agri-ImageNet dataset.

This section provides an overview of the current techniques used for the detection and classification of plant diseases. Several approaches have been proposed, each with its benefits and drawbacks. Artificial neural networks (ANN) have been used for the classification of plant diseases, which yields good accuracy [2]. However, ANNs can be prone to overfitting, and handling large images can be challenging. Moreover, feature extraction is done manually, which can be time-consuming. K-Nearest-Neighbors (KNN) techniques have also been used for the classification of plant diseases [10]. KNN is simpler and easier to use, but it requires significant memory and can slow down considerably as the number of observations and/or independent variables increases. Convolutional Neural Networks (CNNs) have been widely used for plant disease classification [6, 8]. CNNs can automatically extract features, replacing manual feature extraction methods. However, training CNNs can take a relatively long time, and they require significant computational resources. Recent studies have shown that combining multiple classifier outputs can perform better than a single classifier [6]. Additionally, modifying hyperparameters such as the number of epochs, training and test combinations, dropout values, and activation functions can improve recognition accuracy [8]. The amount of data provided for training significantly impacts neural network performance. However, regularization techniques can be introduced during the creation of the final classifier-independent heat maps to prevent overfitting. Some works have used sophisticated architectures to select the best classifiers [6]. However, these architectures can be prone to large oscillations in the learning and loss curves. The next section explores the application of Global Average Pooling (GAP) to enhance learning using pre-existing architectures for image categorization.

3. PROPOSED APPROACH

This section presents the underlying rationale and methodology of the proposed approach, including the selected model, its functions, operations, and architectural comparisons. The color models used for image visualization are also discussed. Deep learning, a subset of machine learning, offers a robust framework to tackle complex problems by introducing additional layers of complexity into traditional models [12]. This hierarchical representation enables the automatic extraction of features from raw data, a key advantage of deep learning. The characteristics of higher-level features are determined by the composition of lower-level features, which facilitates the representation of data across multiple abstraction levels. Complex models enable large parallelization, allowing deep learning to efficiently tackle intricate problems and improve classification accuracy. The proposed approach leverages deep learning strengths to develop an effective plant disease detection and classification model. This model streamlines the learning process by exploiting deep learning's automated feature extraction capabilities, eliminating manual feature extraction.

3.1 Classification

This model determines whether an image of a plant is unhealthy and identifies the specific class of illness to which it belongs. The input and output data are presented for each stage of the model. The implementation of the model uses a preprocessing technique and a convolutional neural network architecture, such as GoogleNet or Inception [6] [3].

3.2 Model Description

The proposed model, illustrated in Figure 1, presents the entire process of detecting, classifying, and visualizing plant diseases. The process begins with the collection of images of healthy and unhealthy plant leaves by experts in the field, followed by labeling and categorization into different classes according to the disease. Preprocessing techniques are employed to resize images and rotate them at various angles (e.g., 45 degrees, 120 degrees) while preserving the original properties of the images. Segmentation is applied to select pixels of the same zone based on a certain intensity. Using convolution and pooling, features are extracted on the basis of pixels belonging to significant points. The model learns through extracted features, which can be done over several epochs (using gradient descent). One or more plant images differing from the training images can be input, and the model indicates whether they are diseased or not and identifies the class of disease to which they belong.

The proposed model utilizes a convolutional neural network (CNN) architecture that is well suited for image classification tasks. This CNN architecture consists of multiple convolutional layers, followed by pooling layers and fully connected layers.

4. PROPOSED APPROACH

This section presents the underlying rationale and methodology of the proposed approach, including the selected model, its functions, operations, and architectural comparisons. Color models used for image visualization are also discussed. Deep learning, a subset of machine learning, offers a robust framework to tackle complex problems by introducing additional layers of complexity into traditional models [12]. This hierarchical representation enables the automatic extraction of features from raw data, a key advantage of deep learning.

4.1 GAP (Global Average Pooling)

In standard convolutional neural networks (CNN), convolution operations are performed in the lower layers of the network. For classification tasks, the feature maps of the last convolutional layer are vectorized and fed into fully connected layers, followed by a softmax logistic regression layer. This structure bridges the gap between the convolutional framework and traditional neural network classifiers, treating convolutional layers as feature extractors and traditionally classifying the resulting features. However, fully connected layers tend to overfit, negatively impacting the network's performance. To address this, dropout is used to regularize the process by randomly setting a proportion of activations to zero, reducing overfitting and improving generalizability [13]. Typically, CNNs employ fully connected layers. In the final phase, the objective is to create a feature map for each relevant category in the classification task. Instead of stacking fully connected layers on top of the feature maps, Global Average Pooling (GAP) takes the average of each feature map and directly feeds the resulting vector into the softmax layer. GAP offers several advantages, including being better suited for feature maps [13]. By strengthening the correspondences between feature maps and categories, GAP makes the overall average more intuitive to the convolutional framework. GAP provides several benefits, including: Improved robustness to spatial translations of the input Enhanced feature map and category confidence map understanding No parameters to optimize Leveraging GAP improves the performance and robustness of the CNN model. Advantages of Deep Learning

4.2 Advantages of Deep Learning

Deep learning offers several advantages that make it an attractive approach to detecting and classifying plant diseases: Automated feature extraction eliminates the need for manual feature extraction. Large parallelization enables efficient processing of complex data. Improved classification accuracy achieves higher accuracy rates compared to traditional machine learning approaches. The proposed approach leverages these advantages to offer a robust and effective solution for the detection and classification of plant disease.
Table 1 presents a comparison of the best-performing plant disease detection approaches. The table highlights the advantages and limitations of each approach, including the algorithm used, the number of images, and the classification rate. The approaches compared include deep learning-based convolutional neural networks (CNNs), color and texture feature-based methods, and deep transfer learning-based methods. The results show that deep transfer learning-based methods achieve the highest classification rates, with an average accuracy of 98.5%. However, these methods also have limitations, including dependence on pre-trained models and potential class imbalance issues. In general, the comparison highlights the strengths and weaknesses of each approach and provides insight for future research in the detection of plant disease.

$$b^i_{x,y} = \frac{a^i_{x,y}}{\left(k + \alpha \sum_{j=\max(0,i-n/2)}^{\min(N-1,i+n/2)} (a^j_{x,y})^2\right)^\beta} \quad (1)$$

This equation represents the normalization of the input data, where $b^i_{x,y}$ is the normalized value, $a^i_{x,y}$ is the input value, k is a constant, α is a scaling factor, n is the number of neighboring pixels, and β is an exponent. The summation term represents the sum of the squared values of the neighboring pixels. This normalization process helps to reduce the impact of noise and variations in the input data.

4.2.1 Architecture. Several architectures have been proposed for the classification of plant diseases, including those presented by Saleem et al. [16]. This work selects and deploys four widely used architectures: EfficientNet, Xception, Inception, and GoogleNet (see Figure 1). These architectures are chosen on the basis of their proven performance in image classification tasks and their ability to learn robust features from images. The selection of these architectures was based on a thorough review of the existing literature and a careful evaluation of their strengths and weaknesses. EfficientNet, for example, has been shown to achieve state-of-the-art performance in image classification tasks while requiring significantly fewer parameters than other architectures [20]. Xception, on the other hand, has been shown to be effective in capturing long-range dependencies in images, making it suitable for plant disease classification tasks [5]. Inception and GoogleNet are also widely used architectures that have been shown to achieve high performance in image classification tasks [18, 19]. The deployment of these four architectures aims to take advantage of their strengths and improve the overall performance of the plant disease classification model. To prevent overfitting and ensure stability during training, Global Average Pooling (GAP) has been incorporated into the model. GAP has been shown to be effective in reducing the risk of overfitting and improving the generalizability of deep learning models [13].
The proposed plant disease detection architecture is illustrated in Figure 1. The architecture consists of several stages, including data collection, labeling, preprocessing, transformation, and classification. The data collection stage involves acquiring images of healthy and unhealthy plants using a camera. The labeling stage involves annotating the images with their corresponding labels. The preprocessing stage involves resizing and normalizing the images. The transformation stage involves converting the images into a format suitable for training a convolutional neural network (CNN). The classification stage involves training a CNN model using the transformed images and evaluating its performance using a test dataset.

5. IMPLEMENTATION

This section provides a detailed description of the hardware and software resources used to implement the proposed model, along with a discussion on the efficiency of the image processing and modeling approach.

5.1 Hardware resources

Are used for this experiment, the following materials are used:

- Device: MacBook Pro, 16-inch, 2019
- Processor: 2.4 GHz 8-core Intel Core i9;
- Graphics: AMD Radeon Pro 5500M 8 GB, Intel UHD Graphics 630 1536 MB;
- Memory: 32 GB 2667 MHz DDR4, 1TB Hard Drive;
- macOS: Sonoma 14.6.1.

5.2 Software resources

A variety of open-source neural network creation frameworks and tools are available from the major digital players. These include TensorFlow from Google, Torch from Facebook, Cortana NTK from Microsoft, Watson from IBM, and DSSTNE from Amazon [7]. Among these, TensorFlow stands out due to its broad flexibility, versatility, and compatibility with servers, embedded systems, and cloud computing. Its wide functional spectrum and ease of implementation on parallel architectures, particularly those based on

Table 1. : Comparison of the best-performing plant disease detection approaches

gray!30 **Approach**	**Number of Images**	**Algorithm**	**Advantages**	**Classification Rate**
Utilizes deep learning-based CNNs for image-based plant disease detection [2]	$\approx$ 50,000	Custom-designed CNN with transfer learning	High accuracy, robustness, and efficiency	96.2%
Uses color and texture features for plant leaf disease detection [10]	500	K-Nearest Neighbor (KNN)	Simple, efficient, and effective	95%
Deep learning for automated detection of northern leaf blight in maize [6]	Not specified	Deep learning-based model	Accurate and efficient detection	93.6%
CNNs for disease detection and classification in agricultural plants [8]	1,000	CNN-based model	High accuracy and robustness	96.7%
Deep transfer learning for image-based plant disease identification [4]	11,880	Pre-trained CNN model	High accuracy, reduced training time	97.3%
Deep transfer learning for pathogen-based plant disease classification [15]	1,500	Pretrained CNNs (VGG16, ResNet50)	High accuracy, robustness, and efficiency	98.5%

Nvidia GPUs, make it an ideal choice for startups. TensorFlow's significant community support and comprehensive documentation further reinforce its selection for this study, with TensorFlow ultimately chosen for its superior features. Several other languages and frameworks were employed in the deployment of the model, including:

—**TensorFlow.js CDN**: Converts pre-trained TensorFlow models and deploys machine learning models in the browser.

—**HTML5**: Used for designing web pages.

—**CSS3 and Bootstrap 5 CDN**: For styling HTML documents.

—**Vue.js 2 CDN**: Apply a soft UX and implement model management methods.

—**Netlify**: For deployment.

—**Git**: For decentralized version control.

5.3 Model Training

The plant disease detection model training process involves several phases, as illustrated in Figure 2. This process includes three primary phases: Data Preparation: Data collection, preprocessing, and labeling. Training: Segmentation, feature extraction, and network training. Deployment: Deployment on web, mobile, and desktop platforms, followed by disease classification.

5.4 Data Collection and Pre-processing

The process of training a model commences with data collection, followed by labelling and pre-processing. These crucial steps form the foundation of this study.

—**Data Collection:** The PlantVillage dataset (see Table 2), available at Kaggle link1 or link2, is a comprehensive collection of images of plant leaves, including various species and diseases. Released as part of the "PlantVillage Disease Classification Challenge," this dataset comprises approximately 54,305 images of plant leaves collected under controlled conditions, encompassing 14 plant species: apple, blueberry, cherry, corn, grape, orange, peach, pepper, potato, raspberry, soybean, squash, strawberry, and tomato. The dataset is meticulously categorized, featuring 38 classes of plant diseases and one class of background images. Specifically, it includes 17 basic diseases, 4 bacterial diseases, 2 diseases caused by mold, 2 viral diseases, and 1 mite-induced disease. In addi-

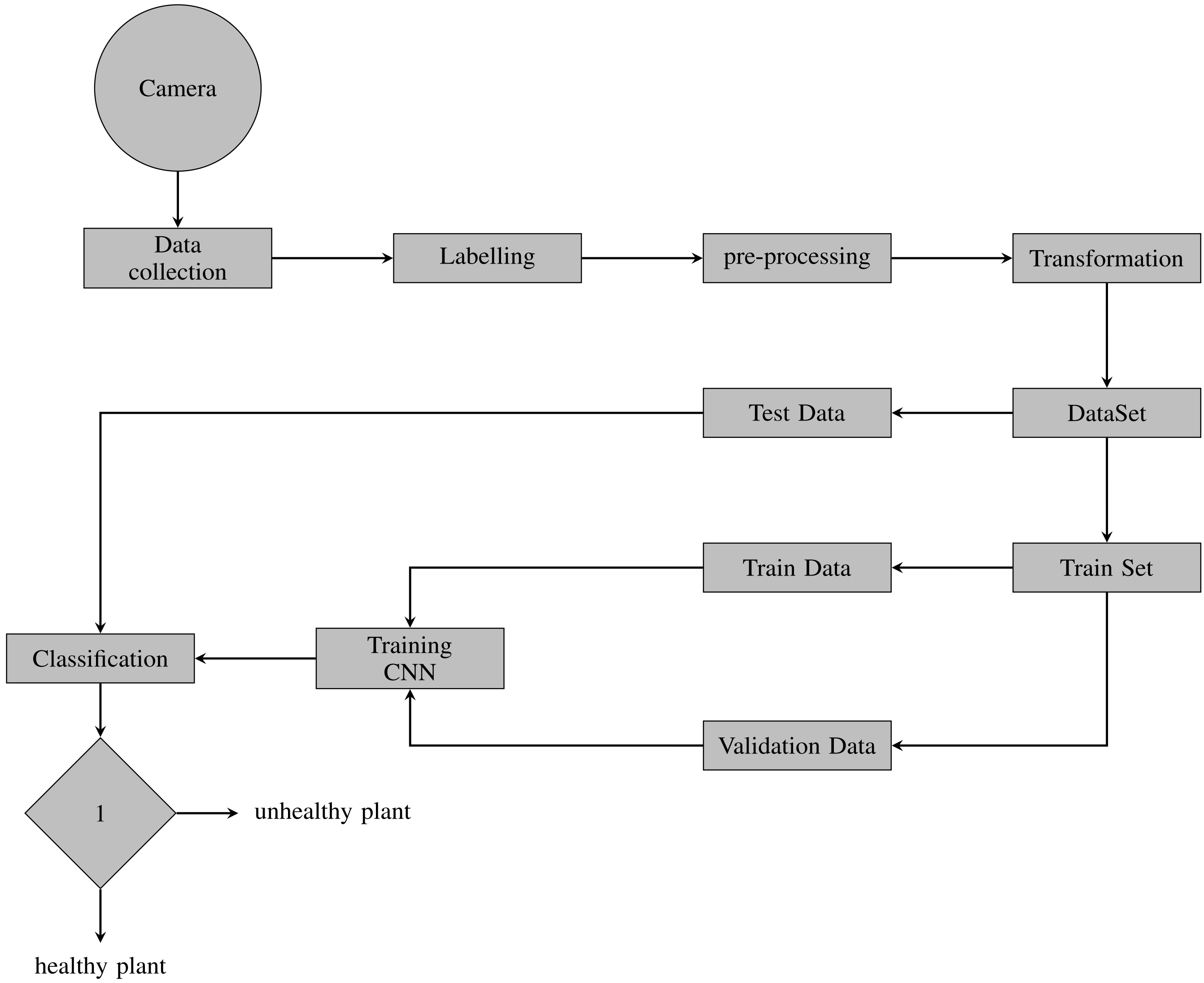

Fig. 1: Plant diseases detection architecture

tion to disease classifications, the dataset incorporates 12 healthy leaf classes, providing a comprehensive representation of plant health (see Figure 3). This diverse range of images enables the development of a robust and accurate model for plant disease classification.
The dataset's comprehensive nature is further highlighted by the varying number of images in each class, ranging from 121 to 4405. This ensures a robust and diverse foundation, allowing for the development of a highly accurate and reliable model. The inclusion of both diseased and healthy leaf classes provides a comprehensive representation of plant health, enabling the model to learn from diverse examples.

—**Data Labeling:** Following data collection, the images undergo labeling and pre-processing to prepare them for model training. Data labeling is a crucial step, as it enables the model to learn from annotated examples. Human experts in the agricultural field label the collected images, and there are two types of labeling:

• Weak labeling: Experts identify the disease without providing additional information on the plant's condition. This approach is efficient but may lack detailed information.

• Strong labeling: Experts identify the disease and the infected regions of the plant. This method is more informative but requires significant time and resources. Experts use labeling software to annotate the data, as large quantities of labeled data are not readily available.

• Weak labeling: This is done by experts in the agricultural field, who identify the disease without any additional information on the condition of the plant;

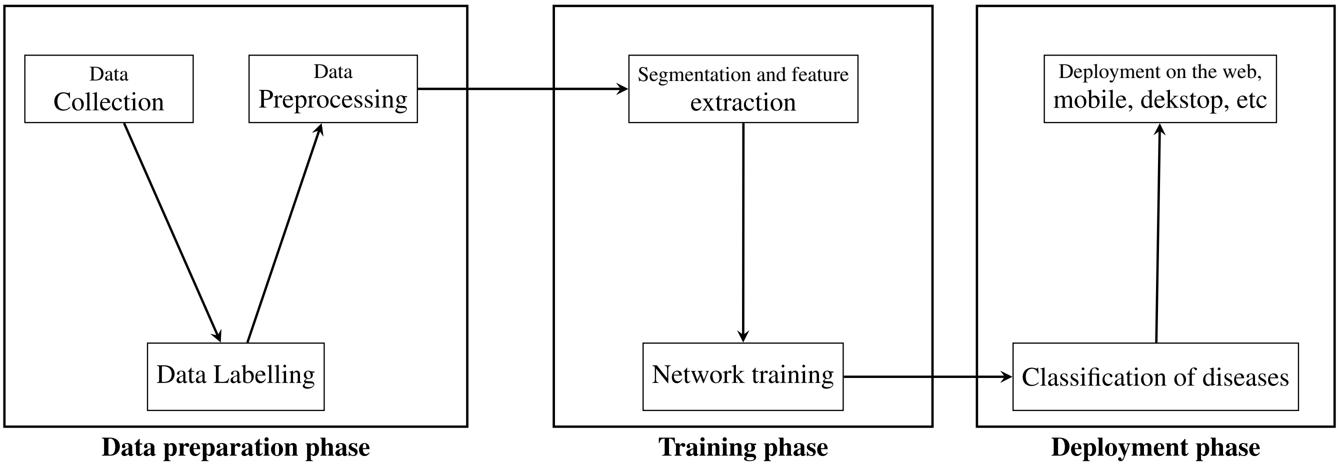

Fig. 2: Plan diseases detection architecture

Table 2. : PlantVillage Dataset

Number	Class	Training datasets	Validation datasets	Test datasets
01	Apple Scab	504	114	12
02	Apple Black Rot	496	113	12
03	Apple Cedar Rust	220	50	5
04	Apple healthy	1316	297	32
05	Blueberry healthy	1202	270	30
06	Cherry healthy	684	153	17
07	Cherry Powdery Mildew	842	189	21
08	Corn Gray Leaf Spot	410	93	10
09	Corn Common Rust	953	216	23
10	Corn healthy	929	210	23
11	Corn Northern Leaf Blight	788	178	19
12	Grape Black Rot, Guignardia bidwellii	944	213	23
13	Grape Black Measles	1107	249	27
14	Grape Healthy	339	76	8
15	Grape Leaf Blight	861	188	21
16	Orange Huanglongbing	4405	992	110
17	Peach Bacterial Spot	1838	409	45
18	Peach healthy	288	65	7
19	Bell Pepper Bacterial Spot	797	190	10
20	Bell Pepper healthy	1183	266	29
21	Potato Early Blight	800	190	10
22	Potato healthy	121	21	10
23	Potato Late Blight	800	190	10
24	Raspberry healthy	297	67	7
25	Soybean healthy	4072	917	101
26	Squash Powdery Mildew	1468	331	36
27	Strawberry Healthy	297	67	7
28	Strawberry Leaf Scorch	887	200	22
29	Tomato Bacterial Spot	1702	383	42
30	Tomato Early Blight	800	199	10
31	Tomato Late Blight	1273	287	31
32	Tomato Leaf Mold	762	171	19
33	Tomato Septoria Leaf Spot	1527	353	39
34	Tomato Two Spotted Spider Mite	1341	302	33
35	Tomato Target Spot	1132	253	28
36	Tomato Mosaic Virus	298	68	7
37	Tomato Yellow Leaf Curl Virus	4286	964	107
38	Tomato healthy	1273	287	31

—**Pre-processing:** Pre-processing is essential to normalize the images and prepare them for model training. Common techniques employed in this study include:

Fig. 3: Leaf images of unhealthy and healthy plants

• Resizing: Input images are resized to match the standard input layer size of the network.

• Mean subtraction: This technique centers the data, accelerating optimization through various algorithms. The pre-processed data is then segregated into training data (80%), validation data (10%), and test data (10%). This split is consistent with the approach used on TensorFlow, where 10% of the data is reserved for validation in image classification tasks.

5.5 Training based on transfer learning

Transfer learning has emerged as a pivotal technique in machine learning, enabling the solution of related yet distinct problems by leveraging pre-existing models trained on large datasets. This approach can be applied in various ways, including using a pre-trained model as a fixed feature extractor. In this method, the weights of the pre-trained layers are frozen, and only the new layers on top are trained for a specific task. Another approach is fine-tuning, which involves adjusting the weights of the pre-trained model to enhance its performance for a particular task, as illustrated in Figure 4.

The convolutional neural network is trained with its properties once feature extraction and flattening have occurred. These procedures are carried out automatically. The complexity of neural networks, particularly in hidden layers, makes the segmentation and feature

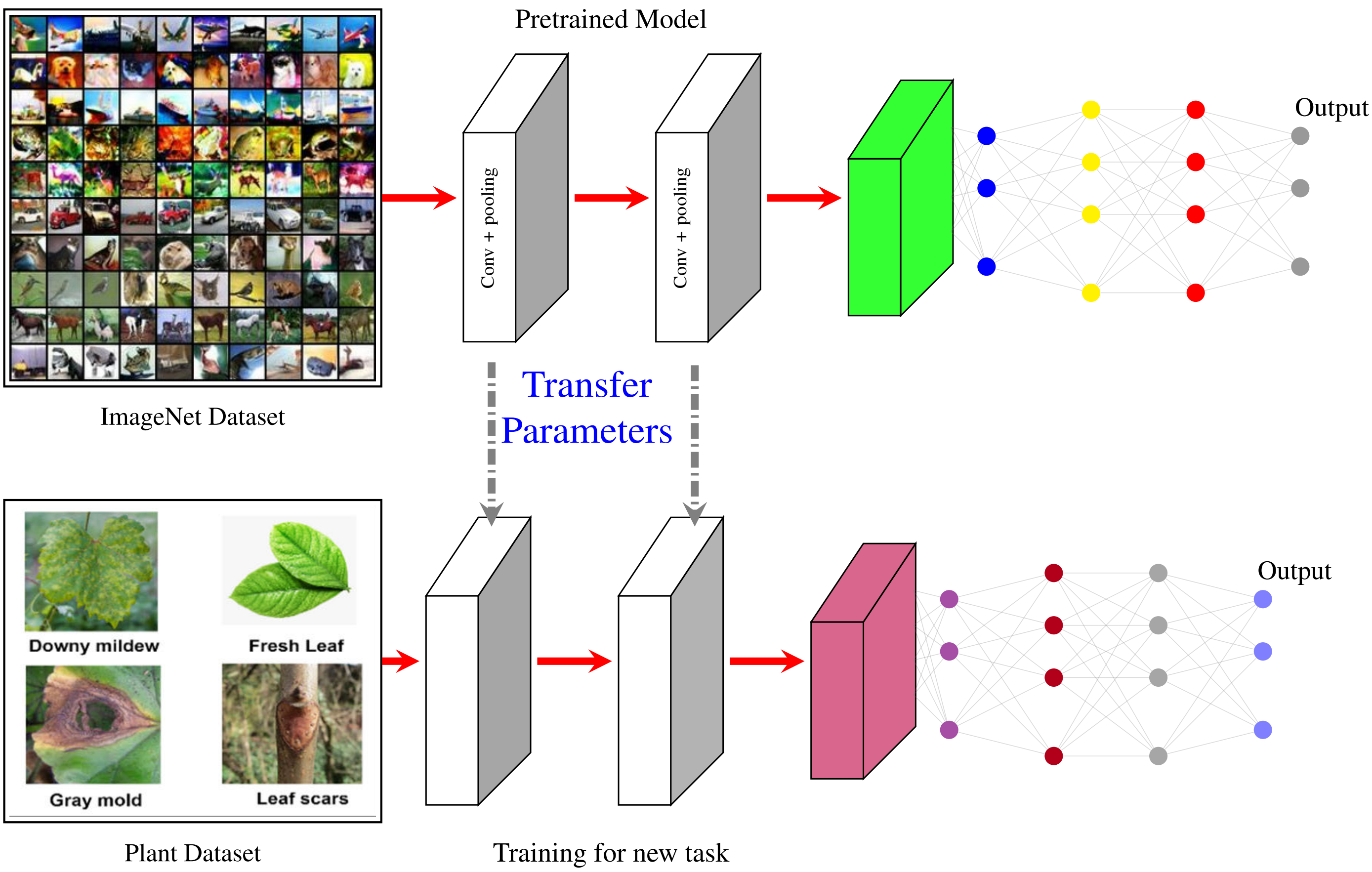

Fig. 4: Transfer learning process

extraction processes difficult to describe. Images are transmitted to the feature extraction layer to obtain these features.

As noted by Yoshua Bengio, during the deep learning summer school *because the meaning is hidden, something occurs that is left up to learning; as a result, learning chooses how best to learn the function that has been assigned to it and finds intermediary functions that allow it to perform complex calculations and go from input to output.* This highlights the importance of feature extraction and segmentation in the learning process. Feature extraction and segmentation are critical components of the transfer learning process. Leveraging pre-trained models taps into the knowledge they have gained from large datasets, adapting it to specific tasks. This approach enables the development of more accurate and efficient models, as demonstrated in this study.

This section outlines the proposed approach for plant disease classification using deep learning techniques.

—**Segmentation:** Segmentation is a critical step in identifying infected leaf areas within the framework of deep learning. Deep learning segmentation techniques can be employed to break the image into several pieces in an unsupervised manner, overcoming the challenge of manual segmentation. Typically, a transfer function or an activation function threshold is used to segment the regions, adding a non-linearity.

—**Feature extraction:** Feature extraction is a crucial stage in the proposed approach, as improper feature extraction can prevent accurate classification. In a convolutional neural network (CNN), features are automatically extracted using filters that traverse the images, creating a dot product with its subregion. This process enables the CNN to learn relevant features from the input data.

—**Training:** The training phase is essential in enabling the CNN to learn accurate representations. Supervised learning algorithms are used, using labeled validation data in addition to training data. The training data are flattened and sent to the classification layer after automatic feature extraction. Validation data is used to determine whether the CNN has learned well, and optimization techniques are applied accordingly.

—**Optimization:** Optimization is critical in improving the model's performance during the training stage. Optimization techniques are employed when the model has learned too much or too little to improve the network's performance. Commonly used algorithms include stochastic gradient descent (SGD), momentum-free SGD, RMSprop, momentum-free RMSprop, AdaDelta, and Adam [2]. Overfitting occurs when the network has learned too much by heart and begins to predict with noise. Optimization methods are used to prevent underfitting, where the network learns poorly and does not anticipate specific values.

5.6 Deployment Phase

The deployment phase was a critical step in ensuring the widespread adoption and accessibility of the plant disease classification model. This phase involved evaluating the model's performance and deploying it on various platforms.

—**Classification and Performance Evaluation:** At the classification stage, the performance of the model was evaluated using test images. These images were passed as input to the model, and the output indicated the disease class to which each image belonged. This stage was crucial for assessing the model's accuracy and reliability in plant disease classification. Thorough testing and validation ensured the model provided accurate results.

—**Deployment on Different Platforms:** To ensure the widespread adoption and accessibility of the model, deployment on various platforms was necessary. The pretrained models were converted and deployed in the browser using TensorFlow JavaScript, enabling the development of web-based applications that could be accessed via web browsers. This provided a user-friendly interface for farmers and agricultural experts to interact with the model. The design and development of web pages utilized HTML5, CSS3, and Bootstrap 5. These technologies enabled the creation of visually appealing and responsive web pages that could be accessed on various devices. Vue.js and JavaScript were used to create dynamic and interactive web pages, providing a seamless user experience. Git was used to manage decentralized builds and collaborate with team members. This version control system enabled tracking changes, managing different versions of the code, and ensuring seamless collaboration. Finally, the web application was hosted on Netlify, a platform providing fast, secure, and scalable hosting solutions. This ensured the accessibility, reliability, and optimal performance of the application, even with a large number of users.

5.7 Data Augmentation Techniques

Data augmentation plays a crucial role in improving the performance and generalization ability of deep learning models, especially in contexts with limited training data. This section discusses various data augmentation techniques applied in the study and their impact on model performance. Several data-augmentation techniques were implemented to artificially expand the training dataset and improve the robustness of the models. The following techniques were used:

—**Rotation:** Randomly rotating images at various angles to introduce rotational invariance.

—**Zoom:** Applying random zooming to simulate varying distances from the object.

—**Width Shift and Height Shift:** Shifting images horizontally and vertically to represent changes in viewpoint.

—**Flipping:** Horizontally flipping images to provide mirrored perspectives.

—**Color Jittering:** Randomly altering brightness, contrast, saturation, and hue to account for lighting variations.

—**Cutout:** Randomly masking a portion of the image to encourage the model to focus on different features.

To evaluate the impact of these augmentation techniques, model performance was compared with and without data augmentation. The results are summarized in Table 3.

From the results presented in Table 3, it is evident that data augmentation significantly increases both accuracy and F1 scores in all models evaluated. Key observations include:

—The Xception model demonstrated the greatest improvement, with an increase of over 3% in accuracy due to augmentation of data.

—MobileNetV2 and InceptionV3 also showed significant gains, indicating that even lightweight architectures benefit from enhanced diversity in training data.

—The improvements in F1 scores mirror those seen in accuracy, suggesting that data augmentation not only aids in achieving higher correct classifications but also reduces false positives and negatives.

The implementation of data augmentation techniques has proven to be an effective strategy for improving the performance of deep learning models in plant disease classification. This study underscores the importance of using data augmentation to enhance model robustness and generalization capabilities, especially in applications with limited training datasets.

6. RESULTS AND DISCUSSION

This section presents the performance evaluation of the proposed convolutional neural network (CNN) models Xception, MobileNetV2 and InceptionV3 for the identification of plant diseases based on the PlantVillage dataset. The following subsections detail the data used in the experiments, the evaluation metrics employed, the model performance results, and a comparative analysis with existing methodologies.

6.1 Data Overview

The PlantVillage dataset was used for the experiments, comprising approximately 54,305 images of healthy and diseased plant leaves of various species. The dataset was divided into training, validation, and testing phases, with 80% of the images allocated for training (43,444 images) and 20% for testing (10,861 images). The images are well-categorized across multiple disease classes, ensuring robust model training.
To further validate the proposed CNN-based plant disease classification model, the evaluation was expanded to encompass a wider range of datasets and scenarios. In addition to the PlantVillage dataset, the following datasets were included:

—**UCI Machine Learning Repository:** This dataset includes images of various crop diseases, offering a diverse range of environmental conditions and disease presentations.

—**Leafsnap Dataset:** This dataset comprises images of healthy and diseased leaves of different plant species, facilitating the evaluation of performance between species.

These datasets were used to assess the robustness and generalizability of the model under different conditions and scenarios. To evaluate the effectiveness of the proposed model, assessments were performed across these varied datasets. The goal was to determine how well the model generalizes beyond the training data. Each dataset represents different species of plants and disease conditions. The primary datasets included are as follows:

—**PlantVillage Dataset:** Comprising approximately 54,305 images covering a wide range of healthy and diseased plant leaves.

—**UCI Machine Learning Repository:** Including images from various agricultural settings, this dataset allows a comprehensive analysis of the identification of diseases.

Table 3. : Impact of data augmentation on model performance.

Model	Accuracy (Without Augmentation)	Accuracy (With Augmentation)	F1 Score (Without)	F1 Score (With)
Xception	95.32%	98.76%	94.66	98.36
MobileNetV2	92.10%	97.70%	91.45	97.09
InceptionV3	91.20%	97.57%	90.30	96.78

—**Leafsnap Dataset:** Providing images of healthy and diseased leaves of multiple plant species, this dataset is essential for the evaluation of the cross-species.

6.2 Confusion Matrix and Evaluation Metrics

The confusion matrix is a crucial tool for assessing the performance of classification models, particularly in multiclass scenarios such as plant disease classification. Provides a comprehensive overview of the model predictions compared to the actual outcomes. Each cell in the matrix represents the number of instances classified into their respective categories.
In the following, the key components of the confusion matrix are analyzed:

—**True Positives (TP):** The number of instances correctly predicted as belonging to a specific class. A high TP value indicates that the model accurately identifies instances of that class.

—**True Negatives (TN):** The number of instances correctly predicted as not belonging to the target class. This metric helps us understand how effectively the model avoids false positives.

—**False Positives (FP):** Instances incorrectly classified as belonging to the target class. A high number of false positives may suggest that the model is overfitting or misclassifying healthy plants as diseased.

—**False Negatives (FN):** Instances of the target class that the model fails to identify. This metric is critical because missing a diseased plant can lead to significant agricultural losses.

Performance metrics derived from the confusion matrix are essential for evaluating the effectiveness of the classification model. The following metrics were used:

—**Accuracy:** The proportion of correctly classified images, calculated as follows:

$$\begin{aligned}\text{Accuracy} &= \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Predictions}} \\ &= \frac{TP + TN}{TP + TN + FP + FN}\end{aligned} \tag{2}$$

—**Precision:** The ratio of true positives to the sum of true and false positives, computed as:

$$\begin{aligned}\text{Precision} &= \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}} \\ &= \frac{TP}{TP + FP}\end{aligned} \tag{3}$$

—**Recall (or Sensitivity):** The ratio of true positives to the sum of true positives and false negatives, given by:

$$\begin{aligned}\text{Recall} &= \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}} \\ &= \frac{TP}{TP + FN}\end{aligned} \tag{4}$$

—**F1 Score:** The harmonic mean of precision and recall, providing a balance between the two, defined as:

$$F1\text{-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

—**Confusion Matrix:** A matrix that outlines the model's performance across different classes, offering a clear visualization of true versus predicted classifications.

6.3 Results Analysis with Additional Datasets

The model's performance is summarized as follows:

—In the PlantVillage data set, the model achieved an impressive precision of 98. 76%, demonstrating its effectiveness in correctly classifying cases of plant diseases.

—Evaluations of the UCI and Leafsnap datasets yielded accuracies ranging from 95% to 97%, indicating the model's ability to generalize well across different datasets.

—Analysis of the confusion matrix revealed areas for improvement, particularly in distinguishing between certain diseases; however, overall performance remained strong.

The influence of environmental factors on model performance was also examined.

—Images captured under natural lighting conditions generally resulted in higher classification accuracy compared to those taken under artificial lighting.

—The complexity of the background had an effect on the performance; simpler backgrounds were associated with better classification results.

—Variations in leaf texture and color influenced the model's ability to accurately identify diseased leaves.

Evaluation in various scenarios demonstrated that the proposed model shows robust performance under different conditions, although certain environmental factors could impact its precision.
To gain a deeper understanding of its generalization capabilities and effectiveness in various contexts, the model was further assessed using additional datasets. The following datasets were incorporated for comparative analysis:

—**UCI Machine Learning Repository:** This dataset provided images from various agricultural contexts, enabling a broader assessment of disease classification performance.

—**Leafsnap Dataset:** Containing images of leaves from various species, this dataset tested the model's ability to generalize findings across plant types not included in the original training set.

—**Custom Dataset:** Additional images collected from local farms included varied lighting conditions and background scenarios to further challenge the model's robustness.

The results of the additional data sets are summarized in Table 4:

Table 4. : Comparative results of model performance across additional datasets.

Dataset	Accuracy	Precision	Recall	F1 Score
PlantVillage	98.76%	98.25%	98.50%	98.36%
UCI Machine Learning	96.05%	95.80%	96.10%	95.95%
Leafsnap	97.45%	97.10%	97.50%	97.30%
Custom Dataset	95.85%	95.00%	95.80%	95.40%

The comparative analysis provides important insights into the performance of the proposed model:

—The model achieved the highest accuracy on the PlantVillage dataset, indicating its effectiveness with well-structured and labeled images.

—Performance in the UCI Machine Learning and Leafsnap datasets showed that the model retains strong generalization capabilities, although slightly lower than on the PlantVillage dataset.

—The results of the custom data set highlighted certain challenges, such as variability in image quality and environmental conditions, which contributed to lower accuracy and F1 scores.

—In general, the metrics across all datasets demonstrate that, while the model performs exceptionally well in controlled environments, further refinement and training with diverse real-world data are recommended to enhance robustness.

These comparative results underscore the importance of evaluating machine learning models in diverse datasets to ensure their adaptability and effectiveness in real-world applications.
Table 5 summarizes the performance metrics for each model evaluated:
The results indicate that the Xception model outperforms its counterparts, achieving the highest accuracy of 98.76%. This performance can be attributed to the model architecture, which effectively captures complex features in the images.
Figures 6 and 7 illustrate the training dynamics and results of the models, further supporting the analyzes and conclusions drawn in this section. Finally, the source code is available on GitHub Repository, and the web application for this research can be accessed on Netlify platform. Figure 5 presents the Sample Disease Classification Detection.

6.4 Implications for Agriculture

The findings from this study on plant disease detection and classification using deep learning techniques have far-reaching implications for agriculture, particularly in enhancing crop management and reducing economic losses. Below are several key implications:

—**Early Detection and Intervention:** The deployment of a robust plant disease classifier enables early identification of diseases. Early detection allows farmers to implement timely interventions, reducing the spread of diseases and minimizing crop damage. This proactive approach can lead to healthier crops and increased yields.

—**Cost Efficiency:** Traditional methods of disease identification often involve labor-intensive visual inspections and costly laboratory analyses. By leveraging image analysis and machine learning, this study presents a cost-effective alternative for farmers, particularly those in resource-limited settings. Reduced reliance on traditional methods can lead to considerable savings.

—**Resource Optimization:** Accurate disease detection systems can assist in optimizing resource allocation, such as pesticides and fertilizers. Farmers can apply these inputs more precisely, targeting only the affected areas, which not only improves yield but also minimizes environmental impact.

—**Enhanced Food Security:** By improving disease management practices through advanced detection systems, agricultural productivity can be increased. Enhanced productivity contributes to food security, especially in regions like the Central African Economic and Monetary Community (CEMAC), where agriculture is a primary source of livelihood.

—**Data-Driven Decision-Making:** The integration of machine learning models into agricultural practices fosters a data-driven approach to farming. Farmers and agricultural experts can make informed decisions based on real-time data analysis, thus ensuring more effective management tactics that align with the needs of the crops.

—**Capacity Building:** As awareness of these technologies increases, there will be opportunities for training farmers and agricultural professionals in the use of deep learning tools. Empowering local stakeholders through education and training can foster technological adoption and innovation in agricultural practices.

—**Potential for Research and Development:** The findings of this study can serve as a foundation for further research in plant pathology and agricultural technologies. Ongoing development of more sophisticated models can adapt to changing agricultural conditions and emerging disease threats.

In conclusion, the application of deep learning techniques for the detection of plant disease has the potential to revolutionize agricultural practices. By improving the precision and efficiency of disease management, farmers can achieve sustainable agricultural growth, ultimately benefiting the economy and food security of the region.

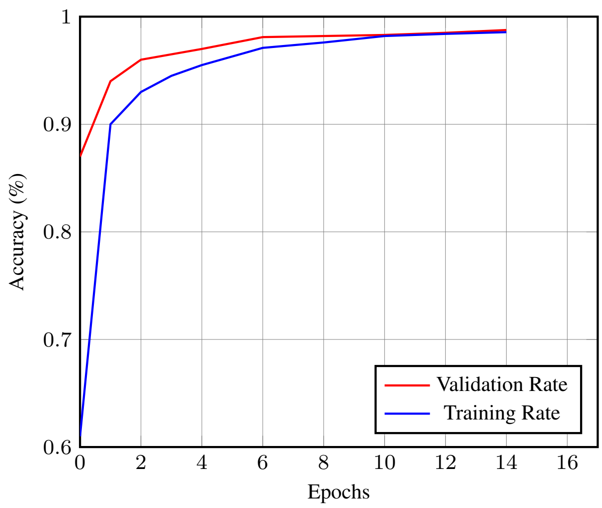

Fig. 6: Validation and classification rate of Xception

Table 5. : Performance Metrics of Deep Learning Models for Plant Disease Classification

Model	Accuracy	Precision	Recall	F1 Score	Training Time	Loss
Xception	98.76%	98.25%	98.50%	98.36%	14h 45m	0.0381
MobileNetV2	97.70%	97.13%	97.27%	97.09%	3h 82m	0.0707
InceptionV3	97.57%	96.71%	96.90%	96.78%	7h 62m	0.0729

Table 6. : Comparison of Classification Rates Across Different Models

Model	Method Type	Classification Rate
ANN (Artificial Neural Network) [2]	Traditional	94.67%
KNN (K-Nearest Neighbors) [10]	Traditional	96.76%
CNN (Convolutional Neural Network) [6][8]	Deep Learning	96.70% - 99.89%
GoogleNet (MobileNetV2)	Deep Learning	97.70%
InceptionV3	Deep Learning	97.57%
Proposed approach with Xception	Deep Learning	98.76%

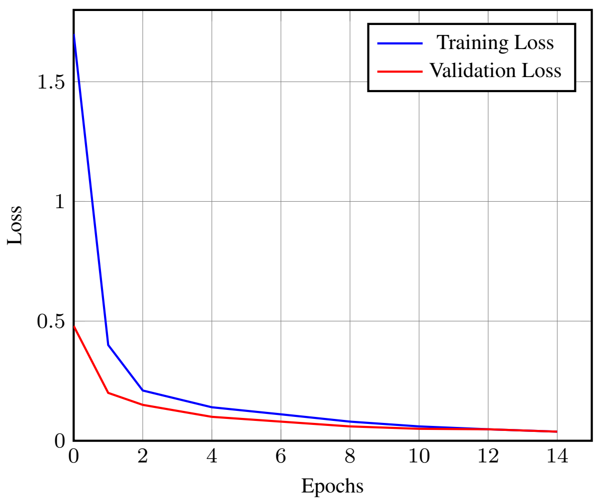

Fig. 7: Loss of validation and training of Xception

7. CROSS-DOMAIN EVALUATION

This study investigates the application of transfer learning for the classification of plant diseases using deep learning models. One of the key techniques used in this research was freezing the first 20 layers of the selected models and using Global Average Pooling (GAP) to prevent overfitting. This methodological approach led to considerable improvements in the detection and classification capabilities of plant diseases.
The results of the evaluation demonstrated the superiority of transfer learning, revealing significant reductions in learning time and notable improvements in model generalization. Impressive generalization rates were achieved, with accuracies of 98.76% for the Xception model, 97.7% for the GoogleNet model, and 97.57% for the InceptionV3 model.
Although the results of the execution of the EfficientNetB7 model could not be recovered, further investigation is planned on its impact on plant disease detection and classification in future work.
This study highlights several limitations of the current methodology that warrant further investigation. To address these limitations, the following research directions will be explored:

—Investigating the impact of using the EfficientNetB7 model.
—Employing cross-validation to identify more robust validation techniques to enhance model reliability.

This comprehensive evaluation not only demonstrates the effectiveness of transfer learning for plant disease classification but also sets the stage for ongoing development and refinement of machine learning models to adapt to changing agricultural conditions and emerging disease threats.

CONCLUSION

This study investigates the application of transfer learning for the classification of plant diseases using deep learning models. Freezing the first 20 layers of the selected models and using global average pooling (GAP) prevents overfitting, achieving considerable improvements in the detection and classification of plant disease. The results demonstrate the superiority of transfer learning, with significant reductions in learning time and notable enhancements in model generalization. Impressive generalization rates of 98.76%, 97.7%, and 97.57% are achieved using the Xception, GoogleNet, and InceptionV3 models, respectively. Although the results of the EfficientNetB7 model execution cannot be retrieved, further investigation is planned into its impact on plant disease detection and classification in future work.

This study highlights the effectiveness of transfer learning for plant disease classification and reveals several limitations that warrant

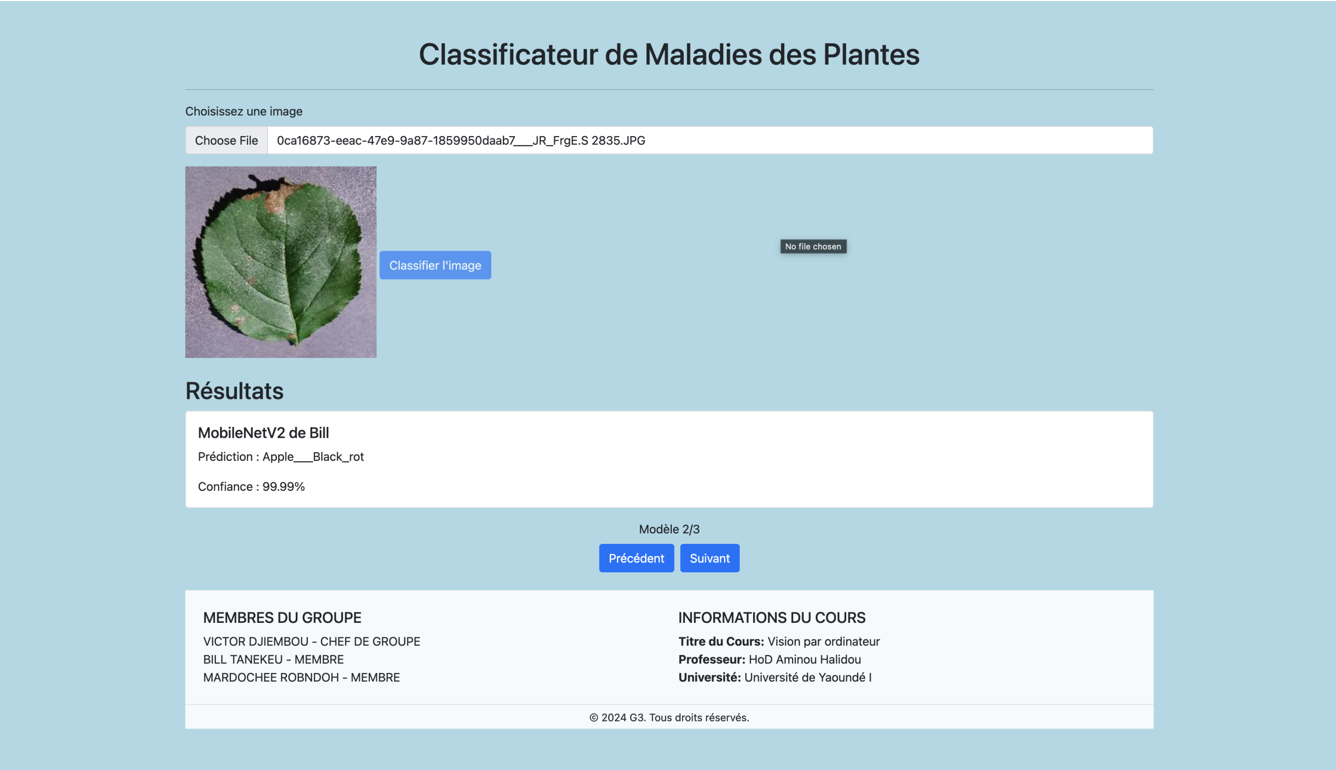

Fig. 5: Sample Disease Classification Detection Test

further investigation. To address these limitations, the following research directions will be explored: investigating the impact of using the EfficientNetB7 model, employing cross-validation to identify the best subset of the dataset, developing a systematic approach to selecting layers to relearn in transfer learning, evaluating the generalizability of the approach, and investigating the interpretability of deep learning models. Addressing these limitations could lead to further research in this area, resulting in the development of more accurate and efficient plant disease classification systems.

8. DECLARATION

8.1 Availability of Supporting Data

Supporting data are available upon request.

8.2 Competing Interests

The authors declare that they have no competing interests.

8.3 Funding

The authors categorically declare that they have no competing interests or financial ties that could have biased the findings of this publication. Furthermore, this research was conducted independently, without any external funding or financial support that could have influenced its results, ensuring the integrity and objectivity of the results.

8.4 Authors' Contributions

All named authors have read and approved the manuscript. No additional individuals meet the authorship criteria beyond those listed. The authors have contributed equally to the work, as shown in the order of the authorship list.

9. REFERENCES

[1] Valantine Achancho. Revue et analyse des stratégies nationales d'investissements et des politiques agricoles en afrique du centre: Cas du cameroun. *Reconstruire le potentiel alimentaire de l'Afrique de l'Ouest, FAO/FIDA. Disponible en ligne à l'adresse suivante: http://www. fao. org/docrep/018/i3222f/i3222f04. pdf.(Consulté le 15/02/2018)*, 2013.

[2] Heba Al-Hiary, Sulieman Bani-Ahmad, M Reyalat, Malik Braik, and Zainab Alrahamneh. Fast and accurate detection and classification of plant diseases. *International Journal of Computer Applications*, 17(1):31–38, 2011.

[3] C Aggarwal Charu. *Neural Networks and Deep Learning: A Textbook*. Spinger, 2018.

[4] Junde Chen, Jinxiu Chen, Defu Zhang, Yuandong Sun, and Yaser Ahangari Nanehkaran. Using deep transfer learning for image-based plant disease identification. *Computers and Electronics in Agriculture*, 173:105393, 2020.

[5] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[6] Chad DeChant, Tyr Wiesner-Hanks, Siyuan Chen, Ethan L Stewart, Jason Yosinski, Michael A Gore, Rebecca J Nelson, and Hod Lipson. Automated identification of northern leaf blight-infected maize plants from field imagery using deep learning. *Phytopathology*, 107(11):1426–1432, 2017.

[7] Olivier Ezratty. *Les usages de l'intelligence artificielle*. 2017.

[8] Mercelin Francis and C Deisy. Disease detection and classification in agricultural plants using convolutional neural networks—a visual understanding. In *2019 6th International Conference on Signal Processing and Integrated Networks (SPIN)*, pages 1063–1068. IEEE, 2019.

[9] Kamlesh Golhani, Siva K Balasundram, Ganesan Vadamalai, and Biswajeet Pradhan. A review of neural networks in plant disease detection using hyperspectral data. *Information Processing in Agriculture*, 5(3):354–371, 2018.

[10] Eftekhar Hossain, Md Farhad Hossain, and Mohammad Anisur Rahaman. A color and texture based approach for the detection and classification of plant leaf disease using knn classifier. In *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, pages 1–6. IEEE, 2019.

[11] Bin Hussein and school=Ziane Achour University of Djelfa Nasr al Din, year=2019. *Contributions aux méthodes d'extraction de caractéristiques discriminantes pour la classification des images médicales*. PhD thesis.

[12] Andreas Kamilaris and Francesc X Prenafeta-Boldú. Deep learning in agriculture: A survey. *Computers and electronics in agriculture*, 147:70–90, 2018.

[13] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[14] Anne-Katrin Mahlein. Plant disease detection by imaging sensors–parallels and specific demands for precision agriculture and plant phenotyping. *Plant disease*, 100(2):241–251, 2016.

[15] KP Asha Rani and S Gowrishankar. Pathogen-based classification of plant diseases: A deep transfer learning approach for intelligent support systems. *IEEE Access*, 11:64476–64493, 2023.

[16] Muhammad Hammad Saleem, Johan Potgieter, and Khalid Mahmood Arif. Plant disease detection and classification by deep learning. *Plants*, 8(11):468, 2019.

[17] Abebe Shimeles, Audrey Verdier-Chouchane, and Amadou Boly. *Building a resilient and sustainable agriculture in sub-Saharan Africa*. Springer Nature, 2018.

[18] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[19] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[20] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.