Deep Learning for Plant Disease Detection: A Novel Convolutional Neural Network Approach Using the new dataset HealthySick Plants

Fatima-Zahra Alaoui[1,*], *Zouheir* Banou[1], *Sanaa* El Filali[1], *Rachida* Ait Abdelouahid[2,3], *Laila* El Jiani[1], *Hasnae* Sakhi[1], and *Olivier* Debauche[3]

[1]University Hassan II Casablanca, Faculty of Science Ben M'Sick,
Laboratory of Artificial Intelligence and Systems (LIAS),
Cdt Driss El Harti B.P. 7955, Ben M'Sick, Casablanca 20800, Morocco
[2]University Hassan II Casablanca, Faculty of Science Ben M'Sick,
Laboratory Information Technologies and Modeling (TIM),
Cdt Driss El Harti B.P. 7955, Ben M'Sick, Casablanca 20800, Morocco
[3]University of Mons, Faculty of Engineering, ILIA,
rue de Houdain 9, Mons 7000, Belgium

Abstract. This research explores a novel application of advanced deep learning techniques, with a particular focus on Convolutional Neural Networks (CNNs), to tackle the critical challenge of plant disease detection. A CNN-based classification model is proposed to accurately distinguish between healthy and diseased plants, utilizing state-of-the-art architectures. The model is trained and evaluated on a rigorously curated version of the Plant Village dataset, adapted in this study to focus on healthy versus sick plant classes. The dataset includes a diverse range of plant species and disease types, enhancing the robustness and generalizability of the model. The developed system achieved outstanding performance, with an accuracy of 99.53% and an AUC of 99.96%, demonstrating the reliability of AI-driven solutions in agricultural diagnostics. These findings highlight the transformative potential of integrating artificial intelligence into precision agriculture, contributing to early detection of plant diseases, reducing crop losses, and promoting sustainable and resource-efficient farming practices. This work represents a step forward in applying AI to support global food security and plant health management.

Keywords: Agriculture 5.0 ; CNN ; Deep Learning ; Disease Detection ; Plant Pathology ; Precision Agriculture

1 Introduction

Artificial intelligence (AI), particularly deep learning, has emerged as a transformative tool for disease detection across various fields, providing scalable and practical solutions. In agriculture, AI-driven methods have revolutionized plant health monitoring, offering automated

*e-mail: fatimazahra.alaoui1-etu@etu.univh2c.ma

and accurate identification of plant diseases through image processing techniques. These advancements surpass the limitations of traditional methods, which are often costly, time-consuming, and labor-intensive [26].

In the era of Agriculture 5.0, which focuses on smart and autonomous farming technologies, deep learning plays a pivotal role [20]. Convolutional Neural Networks (CNNs), a subset of deep learning, enable large-scale, real-time crop monitoring. This not only reduces reliance on manual inspections but also promotes sustainable agricultural practices by optimizing the use of plant protection products. Such approaches are critical for tackling global challenges related to food security, resource optimization, and enhancing the resilience of agricultural systems against climate change.

Despite the progress in AI-driven plant disease detection, existing methods often struggle with precision and applicability in real-world agricultural conditions. Many models are constrained by limited datasets, poor generalizability, and suboptimal architectures. This research addresses these limitations by presenting a robust solution for plant disease detection that integrates cutting-edge image processing with a deep learning framework.

This study's contributions include:

Development of the 'HealthySick Plants' Dataset: To enhance model training and evaluation, we introduce a new dataset derived from the widely recognized PlantVillage database[16]. This dataset is curated to support both binary and multi-class classification tasks, ensuring its applicability in diverse agricultural settings.

Classification of Plant Diseases: We propose a CNN-based model capable of detecting and classifying various plant diseases across different plant types. This model demonstrates superior performance compared to existing methods, achieving a classification accuracy of 99.53% and an AUC of 99.96%.

Advancing Sustainable Agriculture: By leveraging the proposed model, this research contributes to reducing the overuse of pesticides and improving crop management efficiency, aligning with the principles of Agriculture 5.0.

This work builds on recent advancements in AI-driven plant disease research while addressing the limitations identified in previous studies. The methodology includes the construction of a well-structured dataset, the implementation of rigorous pre-processing techniques, and the design of an optimized CNN architecture tailored for binary and multi-class classification.

This article is organized as follows: In the Related Work section, we review relevant studies and identify implementing rigorous pre-processing techniques and designing key limitations in current approaches. The Methodology section describes the dataset creation, preprocessing steps, and the CNN model design. Results and Discussion present the model's performance and its implications for sustainable agriculture. The Conclusion outlines the contributions and potential avenues for future research.

2 Literature review

The literature review section provides a comprehensive overview of recent advancements in plant disease detection using deep learning models, with a focus on CNN-based approaches. This section is divided into two parts: the first part, Related Work, explores significant contributions to the field by summarizing key studies, their methodologies, and their results, highlighting both their strengths and limitations. The second part, the Comparative Study, provides a detailed comparison of these studies, focusing on critical metrics such as accuracy, scalability, generalization, and practical applicability.

2.1 Related works

CNNs have become a cutting-edge method for the early detection of plant disease, revolutionizing agricultural science and plant pathology. By analyzing high-resolution images of plant leaves, CNN models can identify disease-related patterns with remarkable precision. This technology provides automated and rapid diagnostics that offer significant solutions to agricultural challenges such as crop yield optimization, food safety, and sustainable farming practices.

A key contribution to this field is the study by Mimi et al. [21], which focuses on leaf diseases in Catharanthus roseus (bright eyes) and Fragaria ×ananassa (strawberry). Detecting disease in time is crucial to preventing reduced productivity or even plant death. This research employs computer vision techniques to automatically identify leaf disorders using mobile phone images, developing and evaluating three machine learning models: a conventional CNN, a CNN-SVM hybrid, and a MobileNetV2-based transfer learning model. The results show high accuracy, with the MobileNetV2-based model achieving 97.35% accuracy on the test dataset. In addition, an Android application was developed to integrate the best-performing model for real-time monitoring of plant conditions. Despite its success, the reliance on mobile-captured images limits the scalability of this approach under diverse environmental conditions.

Similarly, Goncharov et al. [14] focused on disease detection and prevention in crops, using the PlantVillage database [16] for training. However, they identified a significant limitation: the synthetic nature of the PlantVillage images, which may hinder the accuracy of models in real-world agricultural scenarios. In response, they created a specialized dataset focusing solely on grape leaves and developed a Deep Siamese Convolutional Network tailored to this dataset. The results were promising, with disease detection accuracy exceeding 90% for Esca, Black rot, and Chlorosis in grape leaves. However, the narrow focus of the dataset on grape leaves limits its applicability to other crops or diseases.

Another notable study by Khalid et al. [20] explored the use of Deep Learning (DL) techniques, specifically CNNs and MobileNet architectures, for early detection of plant disease. The CNN model achieved an accuracy of 89%, with high precision and recall scores (96%), while MobileNet achieved higher precision (96%) but with slightly lower precision, recall, and F1-score. The integration of eXplainable Artificial Intelligence (XAI) with GradCAM provided valuable visual insights into the model predictions. Although the study demonstrates the transformative potential of DL in the detection of plant diseases, it is limited by the lack of deployment-ready systems. Future work aims to develop real-time mobile or web applications and integrate DL models with IoT devices for automated disease detection at the farm level.

Mohanty et al. [10] trained a deep convolutional neural network (CNN) using a public dataset of 54,306 images of healthy and diseased plant leaves. The dataset covered 14 crop species and 26 diseases (or their absence). The trained model achieved an accuracy of 99.35% on a held-out test set. This approach demonstrates the potential for smartphone-assisted crop disease diagnosis using deep learning, leveraging large publicly available image datasets. However, challenges remain in deploying this technology in resource-limited agricultural environments.

2.2 Comparative study

The table 1 presents a summary of related studies in the field of CNN-based plant disease detection. These works show notable advances in the use of deep learning models to detect specific diseases and crops, achieving impressive accuracy levels.

Table 1: Comparison of CNN-Based Models for Plant Disease Detection

Study	**Model and Dataset**	**Key Findings and Limitations**
Mimi et al. (2023)	MobileNetV2 on mobile-captured images of *C. roseus* and *F. ×ananassa*	97.35% accuracy; limited scalability due to mobile image dependency
Goncharov et al. (2019)	Deep Siamese CNN on grape leaf subset (PlantVillage)	90% accuracy; narrow focus limits generalizability
Khalid et al. (2024)	CNN and MobileNet on PlantVillage dataset	89% (CNN); not integrated with real-time/IoT systems
Mohanty et al. (2016)	Deep CNN on 54,306 PlantVillage images	99.35% accuracy; lacks adaptability to low-resource contexts

These studies in the table 1 underscore key limitations, such as reliance on narrow datasets, limited generalizability, and challenges in practical deployment, emphasizing the need for scalable and adaptable solutions for real-world agricultural applications.

3 Methods and materials

This section provides a detailed explanation of the methodology used in this study, highlighting each contribution. It presents the development of the HealthySick Plants dataset, which provides a robust foundation for training and evaluating deep learning models in diverse agricultural contexts. Additionally, this section describes the architecture of the Convolutional Neural Network (CNN) model.

3.1 Methodology

In the methodology of this study, the Business Process Model Notation (BPMN) is presented in detail to describe the various processes involved in the system. This representation provides a clear and structured view of all the steps and data flows in the plant disease detection process.

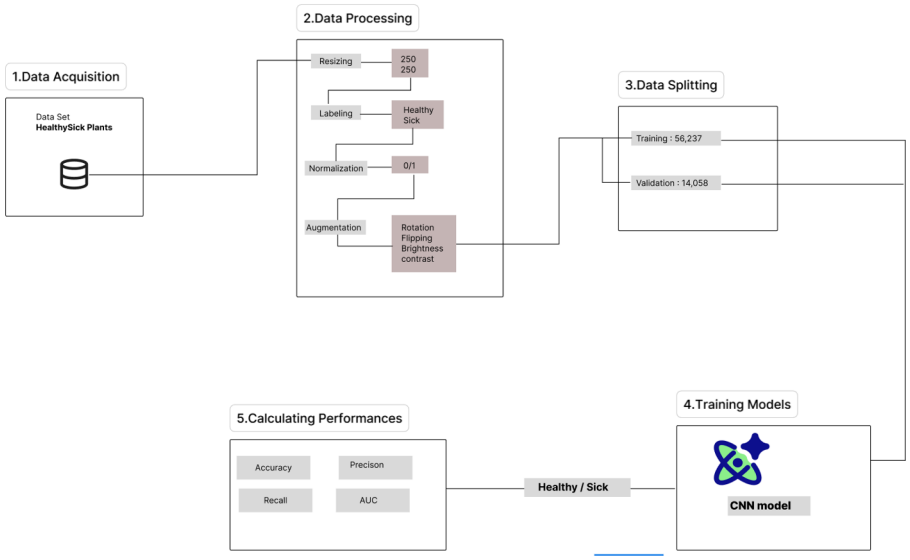


Figure 1: Workflow for Plant Diseases detection.

This figure 1 presents the workflow for classifying plant images as healthy or diseased using the newly developed dataset, "HealthySick Plants". The process begins with data acquisition. The images undergo preprocessing, including resizing, normalization, and data augmentation to enhance quality and dataset diversity.

The dataset, comprising 70,295 images (56,237 for training and 14,058 for validation), is split into healthy and diseased classes. A CNN model is trained using the training set, with evaluation metrics such as accuracy, precision, recall, and AUC applied to assess performance on the validation set.

The BPMN provides a structured and efficient method for visualizing and analyzing these workflows, allowing a detailed understanding of how various components interact within the system. Figure 2 serves as an intricate visual representation, capturing the complex interactions between components and processes designed to accurately detect plant diseases.

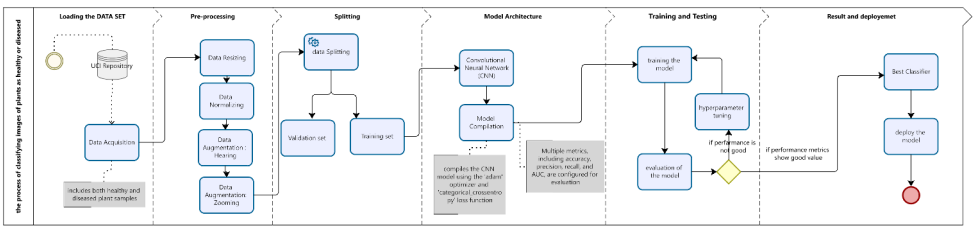


Figure 2: Business Process Model Notation of a System for Detecting Plant Diseases.

Figure 2 illustrates the comprehensive workflow for classifying plant images as healthy or diseased. It begins with data acquisition, where images of plant leaves are collected and stored in a centralized repository, ensuring a diverse representation of healthy and diseased plants. These images are then preprocessed to enhance their quality and prepare them for

model training. Preprocessing includes resizing all images to a uniform dimension of 250 × 250 pixels, normalizing pixel values to improve model convergence, and applying data augmentation techniques such as rotation, zooming, and brightness adjustments to increase the diversity and robustness of the dataset. Once preprocessed, the dataset is split into a training set, used to optimize the model's parameters, and a validation set, reserved for evaluating the model's generalization capabilities on unseen data. The core of the workflow involves a Convolutional Neural Network (CNN) architecture, which is compiled using the Adam optimizer and a categorical cross-entropy loss function. Evaluation metrics, including accuracy, precision, recall, and AUC, are configured to assess the model's predictive performance comprehensively. During the training and testing phase, CNN is trained on the training dataset and its performance is evaluated on the validation dataset. If the results are unsatisfactory, hyperparameter tuning is performed to optimize the model configuration. Once the best-performing model is selected, it moves to the deployment phase, where the classifier is used to process new plant images and predict their health status. This systematic workflow ensures accuracy, robustness, and real-world applicability, making it a reliable tool for the management of agricultural diseases.

3.2 Development of new dataset HealthySick Plants

For this study, we developed a new dataset named HealthySick Plants, derived from the PlantVillage dataset [29]. The dataset consists of 20,639 images, including 13,779 images of diseased plant leaves and 13,779 images of healthy (uninfected) plant leaves, ensuring a well-balanced representation. These images cover 15 crop species, such as potatoes, corn, tomatoes, and peppers, and encompass a variety of plant pathologies, including five fungal diseases, two mold-related diseases, two bacterial infections, one tick-borne disease, and two viral diseases. Additionally, the dataset includes healthy leaf samples from the same crop species.

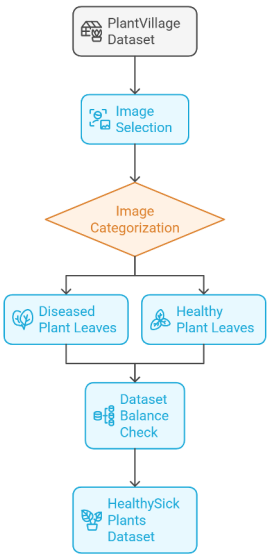


Figure 3: Workflow for Constructing the HealthySick Plants Dataset from the PlantVillage dataset

To enhance the utility and simplicity of the dataset, we conducted a comprehensive review and restructuring of the original PlantVillage dataset. This revision aimed to streamline the classification process for improved efficiency and intuitive use. One significant modification was merging all disease classes into a single category labeled ‘sick‘, consolidating all plant diseases under one label for consistent classification. Similarly, all healthy (non-diseased) plant classes were combined into a single category labeled ‘healthy‘. This restructuring not only simplifies the classification process but also facilitates data management and analysis, making the dataset more suitable for real-world applications in crop protection and agriculture. By providing a well-organized and balanced dataset, the HealthySick Plants dataset offers a reliable foundation for training robust deep learning models capable of accurately detecting and classifying plant diseases. Table 2 provides a detailed overview of the HealthySick Plants dataset used for the image classification of plant diseases. The dataset is divided into two main classes: "healthy" and "sick," representing the conditions of the plants depicted in the images.

Table 2: HealthySick Plants dataset used for image classification of plant diseases.

Class	Original Images	Training Images	Validation Images
Healthy	22,294	56,237	14,058
Sick	48,001	56,237	14,058

- **Number of Original Images**: This column lists the total number of images originally available in each class. The "healthy" class contains 22,294 images, containing approximately 31.7% of the total dataset, while the "sick" class has 48,001 images, representing approximately 68.3%. This distribution indicates that the dataset is imbalanced, with a significantly higher number of images in the "sick" category.

- **Training Images**: This column details the number of images allocated to train the model. Both the "healthy" and "sick" classes were augmented to 56,237 images each for training purposes. This approach helps to balance classes during model training, ensuring that the model does not become biased toward the more prevalent class.

- **Validation Images**: This column shows the number of images used to validate the performance of the model. Each class has 14,058 images allocated for validation, allowing for an unbiased assessment of the model’s generalization capabilities.

Despite the original imbalance, the balanced distribution of training images across both classes ensures that the model is trained on an equal number of examples from each class. This strategy is essential to prevent the model from favoring the ’sick’ class due to its higher prevalence in the original dataset. However, the validation set retains its original distribution, objectively evaluating the model’s performance on naturally distributed data.

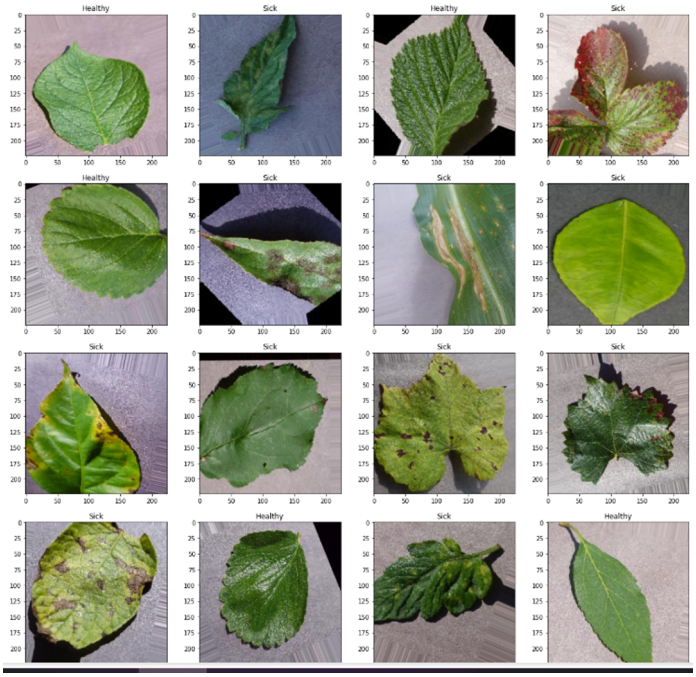


Figure 4: Dataset sample.

3.3 Pipeline for plant Disease Detection

3.3.1 Image acquisition and preprocessing

In this section dedicated to image acquisition and preprocessing, the workflow includes acquiring, loading, and exploring diverse datasets using libraries such as OpenCV (opencv.org) and OS. This phase is crucial because it establishes the groundwork for the subsequent analysis and model development. Initially, images are sourced from various channels to ensure a complete representation of the targeted plant diseases. Subsequently, these images are loaded into the system, taking advantage of the capabilities of OpenCV and OS libraries to efficiently manage file operations. Dataset exploration involves scrutinizing the characteristics and dimensions of the images and addressing potential challenges such as variations in lighting, resolution, and background clutter. In addition, employing preprocessing techniques can enhance image quality, eliminate noise, and standardize features, facilitating more precise disease detection. In general, meticulous attention to image acquisition and preprocessing is imperative to optimize the performance and reliability of the subsequent stages of the plant disease detection process [23].

3.3.2 Images Resizing

Image resizing is a crucial preprocessing step in the detection of plant diseases through Convolutional Neural Networks (CNN) [18]. Its primary objective is to adjust the dimensions of the input images while preserving their aspect ratio. This process serves various purposes in CNN-based disease detection [4]. Resizing ensures a consistent input size for the images, thus achieving uniformity across the dataset. This uniformity greatly aids in efficient model training, enhancing CNN's capacity to effectively learn pertinent features from the images.

- *batch_size* : A batch size of 100 means that in each training iteration, the model will process and update its weights based on 100 training samples [12].
- *img_height* : This variable is set to the value 250. It can represent the height (in pixels) of an image in the dataset. This is commonly used in computer vision tasks where images are resized or processed before being fed into a neural network.
- *img_width* : This variable is also set to the value 250. This value can represent the width (in pixels) of the image in the dataset. Like *img_height*, it is often used in computer vision tasks to determine the size of an image [12].

3.3.3 Labeling and reshaping images

In the work, we emphasize the importance of labeling and re-positioning images of diseased plants, whether they are sick or healthy [13]. We are involved in the labeling process, which includes classifying each image and labeling it as a symptom of the disease or representing a healthy plant.

3.3.4 Image Normalization

Image normalization is crucial for deep learning tasks such as image classification to detect plant diseases [7, 17, 32]. It involves scaling pixel values to have a mean of 0 and a standard deviation of 1 [7, 17]. Normalization improves the ability of neural networks to converge during training by reducing the internal shift of covariates. The internal covariate shift refers to the change in the distribution of network activations due to changes in the parameters of the previous layers during training [17]. Different normalization methods have been evaluated for their impact on deep learning performance, including setting the mean and standard deviation, histogram matching, percentiles, combining percentiles and histogram matching, and fixed window [7]. The percentile method combined with histogram matching worked best to predict sex and treatment outcomes [7]. Normalization plays a minor role in segmentation tasks, with less than 0.02 difference in Dice score between the best and worst performing networks [7]. However, it is especially important for classification tasks, with the largest difference in performance depending on the normalization method [7].

Batch normalization, which normalizes inputs to each layer by the mean and variance of the current batch, has been shown to accelerate deep network training and reduce the need for careful parameter initialization and lower learning rates [17]. It allows for the use of much higher learning rates and significantly improves accuracy [17].

3.3.5 Data Augmentation

Data augmentation is a technique that is used to artificially increase the size of the training dataset by applying different transformations to existing images [27]. This helps improve the generality and robustness of the Deep Learning model. For image data, common data enhancement techniques include rotation, flipping, zooming, and random brightness or contrast changes.

Table 3: Comparison of the number of images before and after data augmentation.

Stage	**Total Number of Images**
Before Augmentation	56,237
After Augmentation	70,295

Table 3 compares the number of images in the dataset before and after applying data augmentation techniques. The HealthySick Plants dataset consisted of 56,237 images and through augmentation, the total number of images increased to 70,295.

3.3.6 Splitting Dataset

To divide the image dataset of the diseased plant into training and testing groups, we used Python and libraries such as NumPy or Scikit-learn [31].

Before starting training the CNN model, we performed data preprocessing. This step was crucial in ensuring the quality and relevance of the dataset for this purpose. We performed operations such as resizing, normalization, and data balancing to ensure that the model could learn the characteristics and patterns associated with healthy and diseased plants in the HealthySick Plants dataset.

Furthermore, we employed data-augmentation techniques to enhance the model's ability to deal with variations in lighting, different angles, and other environmental factors. This process involved generating additional training examples through random transformations such as rotation, flipping, and brightness adjustments [19]. These augmented images helped our CNN model to generalize better and improve its accuracy in detecting plant diseases under various real-world conditions.

Table 4: Dataset distribution for image classification of plant health.

Dataset (Set)	**Class**	**Number of Images**
Training	Healthy	17,836
Training	Sick	38,401
Validation	Healthy	4,458
Validation	Sick	9,600
Total	Healthy	22,294
Total	Sick	47,001

The dataset used for the classification of plant health consists of a total of 70,295 images, distributed between two classes, healthy and sick. The training set contains most of the data, with 56,237 images (17,836 healthy and 38,401 sick), while the validation set is smaller, with 14,058 images (4,458 healthy and 9,600 sick). This distribution ensures a robust dataset for training and validating the model, with a higher representation of the sick class to reflect real-world challenges in plant health monitoring.

3.4 Classification and detection of plants diseases: Model Architectures

In the field of plant pathology and agricultural sciences, the use of convolutional neural networks (CNN) for the detection of plant disease has emerged as a transformative and highly effective approach. CNNs are a type of deep learning model designed for image analysis, which makes them well-suited to the difficult task of identifying diseases in plants. This technology represents a significant advancement over traditional methods, which often rely on manual visual inspection, specialist knowledge, and tedious processes.

3.4.1 Hyperparameters

This table 5 details the hyperparameters used to train the proposed CNN model.

Table 5: Hyperparameters used for training the CNN model.

Hyperparameter	**Value**
Optimizer	Adam
Learning Rate	0.001
Number of Epochs	50
Batch Size	100
Loss Function	Categorical Cross-Entropy

In the table5 Adam optimizer was chosen for its efficient and adaptive learning capabilities, with a learning rate set to 0.001 to balance convergence speed and stability. The model was trained for 50 epochs with a batch size of 100 to optimize performance while managing computational efficiency. Categorical Cross-Entropy was selected as the loss function, given its suitability for multi-class classification tasks.

3.4.2 Feature Extraction

The proposed model employs a convolutional neural network (CNN) architecture for feature extraction, which is designed to efficiently capture spatial hierarchies in the input images. The feature extraction pipeline begins with an **Input Layer** that processes images of dimensions 250×250 pixels with three color channels (RGB), resulting in an input shape of $(250, 250, 3)$.

The first layer of the network, a **Convolutional Layer** (denoted as *conv1*), consists of 32 filters, each with a kernel size of 3×3. This layer applies the filters to the input image, generating an output feature map of size $(124, 124, 32)$. The reduction in spatial dimensions results from the convolution operation, which preserves important features while reducing the overall size of the feature maps.

Following the first convolutional layer, a **MaxPooling Layer** (denoted as *maxpool1*) with a pool size of 2×2 is applied. This operation further reduces the spatial dimensions of the feature maps to $(62, 62, 32)$ by downsampling the input, effectively retaining the most significant features and reducing computational complexity.

To further enhance the model's capacity to capture intricate patterns, a second **Convolutional Layer** (denoted as *conv2*) is introduced. This layer utilizes 16 filters with a kernel size of 3×3, which processes the downsampled feature maps of the previous layer, producing an output with dimensions $(61, 61, 16)$. The addition of this second convolutional layer serves a dual purpose: it allows the model to detect more complex and abstract features by building upon the basic patterns identified in the earlier layers, and it reduces the depth of the feature maps. The reduction in the number of filters from 32 to 16 is a strategic design choice aimed at balancing the model's expressiveness with computational efficiency. In doing so, the model can focus on refining the features most relevant to the classification task without unnecessarily increasing the computational load.

This is followed by a second **MaxPooling Layer** (denoted as *maxpool2*), which applies a 2×2 pooling operation, reducing the dimensions of the feature map to $(30, 30, 16)$. This additional downsampling step continues to reduce the computational load while preserving the most critical information from the feature maps.

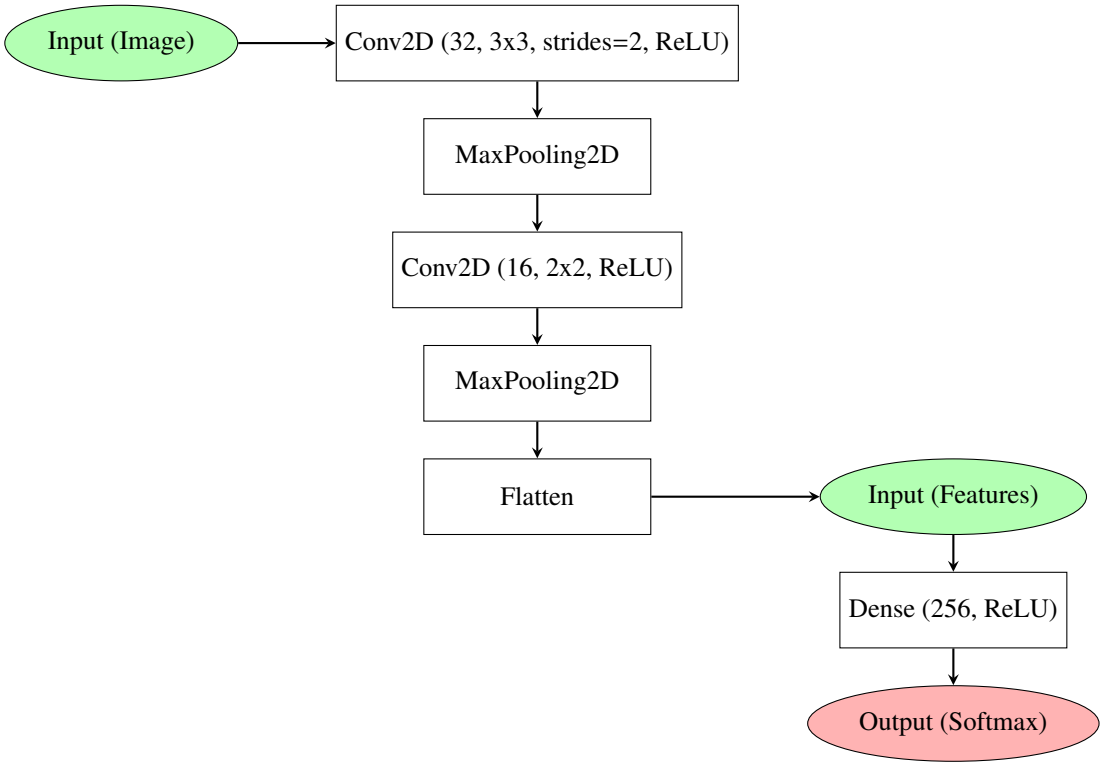


Figure 5: Model architecture .

Finally, the feature extraction process concludes with a **Flatten Layer** that converts the two-dimensional feature maps into a one-dimensional vector of 14,400 elements. This flattened vector serves as a compact representation of the features extracted from the input image, which is then passed to subsequent classification layers.

This feature extraction component is crucial as it transforms raw input images into high-level representations, capturing essential spatial features necessary for accurate image classification in the following stages of the model.

3.4.3 Classifier

Following the feature extraction phase, the model transitions into the classification phase, which is responsible for making the final prediction based on the features extracted by the preceding layers. The classifier is composed of fully connected (Dense) layers that progressively reduce the dimensionality of the data and ultimately output the classification results.

The classification pipeline begins with an **Input Layer** (denoted as *input_clf*), which receives the flattened feature vector from the feature extraction component. This vector has a dimension of 14,400 elements, representing the high-level features extracted from the input image.

The first layer of the classifier is a **Dense Layer** (denoted as *hl1*), which consists of 256 neurons. This layer applies a fully connected operation to the input vector, transforming the

input shape from (14400) to (256). The primary function of this layer is to capture complex interactions between features, allowing the model to learn and represent more abstract concepts that are critical to accurate classification.

Finally, the classification process concludes with another **Dense Layer** (denoted as *output*), which serves as the output layer. This layer reduces the dimensionality further to 2 neurons, corresponding to the two possible classes in the classification task. The output of this layer is a vector of size (2), which represents the predicted probabilities for each class.

This classifier architecture is designed to efficiently map the high-dimensional feature space to the final output classes, enabling the model to make accurate predictions based on the learned features.

3.4.4 Training and Evaluation

This scientific paper outlines a comprehensive training and evaluation framework for a prominent deep learning model, the Convolutional Neural Network (CNN). These models have been used to effectively detect plant diseases, differentiating between healthy and diseased plants [8]. This study meticulously documents the entire process, including data preprocessing, model architecture selection, and the training phase, including optimizer choices and learning rate schedules. Subsequently, we conducted a rigorous evaluation, employing an independent test dataset, reserved for this purpose to maintain the integrity of the assessment. This dataset remains intact during the training and validation phases [28]. Throughout this research, we scrutinize the models for various factors, including alterations in image quality, disease severity, and background clutter, offering valuable insights into their adaptability and robustness in real-world agricultural scenarios. The work underscores the importance of a systematic approach that takes advantage of state-of-the-art deep learning models to address the critical task of detecting plant disease, thus contributing to agricultural systems' overall health and productivity.

4 Experiments and results

This section presents the results, and analyses conducted to evaluate the performance of the proposed methodology for plant disease classification and detection. The results are analyzed in detail, including accuracy, precision, recall, and AUC metrics, to assess the model's effectiveness and reliability. A comparison with related work is also included to highlight the contributions and improvements achieved in this study.

4.1 Results

Table 6 provides a breakdown of the performance metrics for the convolutional neural network (CNN) model used in the experiment. The table presents key evaluation metrics, including precision, recall, F1-score, area under the curve (AUC), and loss, for both the training and validation datasets. By comparing the results across the training and validation subsets, we can evaluate the generalizability of the model and ensure that it does not suffer from significant overfitting.

4.2 analysis

This section presents a detailed analysis of the performance of the convolutional neural network (CNN) model, including its performance history to illustrate the learning behavior over

epochs. Additionally, confusion matrices for both training and validation datasets are analyzed to provide insights into the model's classification accuracy and its ability to distinguish between healthy and diseased plants.

Table 6: Performance metrics for the convolutional neural network (CNN) model.

Model	Subset	Accuracy	Precision	Recall	F1 Score	AUC	Loss
CNN	Training	99.53	99.53	99.53	99.53	99.96	0.01
	Validation	98.29	98.29	98.29	98.29	99.63	0.05

The performance of the convolutional neural network (CNN) model is summarized in Table 6, demonstrating its high efficacy in the image classification task. The model achieved an impressive accuracy of 99.53% in the training set and 98.29% in the validation set, indicating robust generalizability. Both precision and recall metrics are equally high, with values of 99.53% for training and 98.29% for validation, suggesting that the model not only predicts the correct class with high accuracy, but also identifies a substantial portion of actual positive instances. The F1-score, which balances precision and recall, further supports the model's reliability with identical values, emphasizing its consistency in performance. Moreover, the AUC values, approaching 100%, illustrate the model's exceptional ability to distinguish between different classes. The minimal loss observed, 0.01 for training and 0.05 for validation, reflects the model's low error rate, confirming that the CNN effectively learned the underlying patterns in the data with minimal overfitting. In general, these metrics collectively affirm the strong performance and suitability of the CNN model for the classification task at hand. To further illustrate the model learning process and the consistency of its performance in the training and validation datasets, Figure 6 presents the learning curves for various metrics over 50 training epochs.

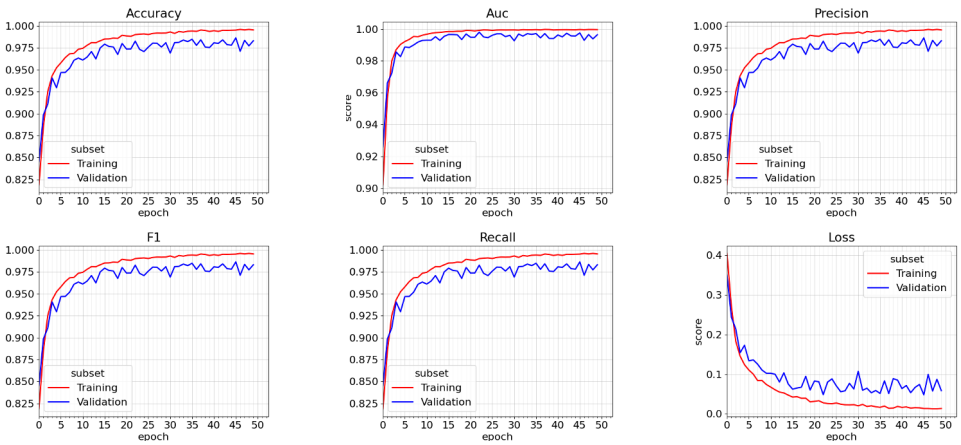
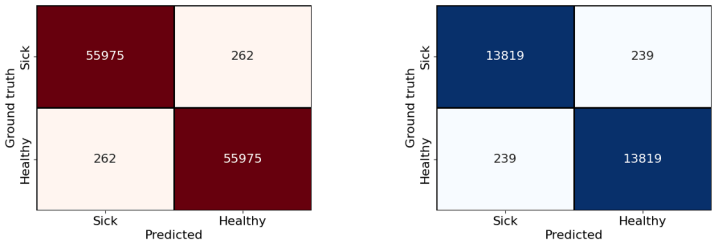


Figure 6: model performance history

Figure 6 provides a set of six graphs that display the progression of key performance metrics: accuracy, AUC, precision, F1-score, recall, and loss throughout the training process. Each plot includes curves for both the training and validation subsets, enabling a clear comparison of the model's behavior on seen and unseen data. The accuracy and AUC curves,

both nearing 1.0 as the epochs progress, indicate that the model rapidly learns to classify the images correctly and maintains this ability across validation data, reflecting excellent generalization. The precision, F1-score, and recall curves similarly show consistent improvement, eventually stabilizing at high values, corroborating the model's robustness in identifying and classifying positive instances. The loss plot further illustrates the model's learning efficiency, with both training and validation loss decreasing steadily and converging to low values, signifying the effective minimization of prediction errors. Collectively, these plots provide a comprehensive view of the CNN model's performance, affirming its reliability and effectiveness for the image classification task.

Figure 7 presents a comparative analysis of model performance in training and validation datasets using confusion matrices. The red matrix corresponds to the training data, while the blue matrix illustrates the performance on the validation data.



(a) train confusion matrix (b) val confusion matrix

Figure 7: Confusion matrices for training and validation

In the training set, the model correctly classified 55,975 instances in both the Sick and Healthy categories, representing 99.53% accuracy in each category. The model misclassified only 262 instances as false positives or false negatives, which constitutes only 0.47% of the total training data. Similarly, in the validation set, the model correctly classified 13,819 instances in both categories, achieving 98.29% accuracy. The 239 misclassifications in the validation set account for just 1.71% of the data.

These confusion matrices emphasize the model's high accuracy, with the vast majority of predictions falling on the diagonal, indicating correct classifications. The close alignment between the training and validation matrices, with less than a 1.25% difference in accuracy, suggests that the model generalizes well to unseen data and does not overfit. The strong diagonal dominance in both matrices, where correct classifications vastly outnumber misclassifications, underscores the model's reliability and effectiveness in correctly identifying the classes in this image classification task.

4.3 Comparative with literature review

we compare the results of our study with those of previous works to highlight advancements in plant disease detection using CNN-based models.

Table 7: Comparison of CNN-based models for plant disease detection.

Study	Focus	Model	Accuracy
Mimi et al. (2023)	Leaf disease detection	MobileNetV2	97.35%
Goncharov et al. (2019)	Disease detection for grape leaves	Deep Siamese CNN	90%
Khalid et al. (2024)	Early plant disease detection	CNN	89%
Mohanty et al. (2016)	Multi-crop, multi-disease classification	CNN	99.35%
This Study	Generalized multi-disease detection	CNN	**99.53%**

Comparison of the results of this study with previous work, as presented in the table 7, highlights the significant advancements of this novel approach in the detection of plant disease. While earlier studies, such as those by Mimi et al. [21], Goncharov et al. [14], and Mohanty et al. [10], achieved high levels of accuracy (up to 99.35%), they often exhibit critical limitations. Some studies focus on specific crops, such as grape leaves or limited species like Catharanthus roseus and Fragaria ×ananassa, which reduces their applicability to other crops or agricultural environments. In contrast, the CNN model, trained on the HealthySick Plants dataset, stands out due to its superior performance and generalization capabilities, achieving an accuracy of 99.53%.

5 Discussion

5.1 Advancements and Contributions of This Study

Building on previous work in this domain, such as studies by Mimi et al. [21], Goncharov et al. [14], and Konstantinos et al. [9], this research takes significant strides toward overcoming some of these challenges. While prior models have demonstrated exceptional performance within specific domains—such as leaf diseases in Catharanthus roseus and strawberries (Mimi et al.) or grape leaves (Goncharov et al.)—these studies often focus on limited datasets or single-crop applications. Although effective, these approaches lack scalability and generalizability, which are critical for real-world deployment.

The distinguishing feature of the work lies in its broad scope and generalization capability. Unlike prior models, the CNN-based framework was trained on a diverse dataset (HealthySick Plants), encompassing 20,639 images of 15 crop species affected by a wide range of diseases. By consolidating disease classifications into a single 'sick' category and healthy plant classifications into a 'healthy' category, our dataset and model achieve a higher level of simplicity, robustness, and applicability.

5.2 Experimental Results and Model Performance

The model achieves superior performance, with a classification accuracy of 99.53% and an AUC of 99.96%, surpassing many existing models cited in the literature. These results are supported by strong metrics for precision and recall, indicating both accuracy and reliability in detecting plant diseases. Confusion matrices further validate the model's effectiveness, demonstrating consistent and accurate classification of both healthy and diseased plants across the training and validation datasets.

Moreover, the approach reduces computational overhead by offering a single, general-purpose model that can replace multiple disease-specific models. This is particularly advantageous for resource-constrained environments, where deploying multiple models may not be

feasible. The streamlined framework enables efficient and simultaneous detection of multiple diseases, making it a practical solution for field conditions, especially on low-cost hardware.

5.3 limitations of study

Despite the remarkable progress in plant disease detection and classification using CNNs, several challenges persist for large-scale practical implementation, particularly when dealing with real-world agricultural conditions and resource constraints. CNN-based detection systems often rely on high-quality images, which are contingent on the resolution and positioning of sensors. Low-cost sensors commonly used in agriculture may lack the required resolution to capture subtle disease symptoms, potentially reducing the accuracy of detection under field conditions.

Environmental factors such as lighting variability, shadows, and weather conditions further complicate the deployment of such models. Poor lighting or heavy shading can obscure symptoms, while diverse soil types and complex backgrounds introduce noise, impacting the robustness of the models. These limitations highlight the need for future advancements to improve model resilience and adaptability.

5.4 Strategies to Address Current Challenges

To address these limitations, several promising directions for future research can be explored. First, integrating advanced preprocessing techniques, such as contrast enhancement and image denoising, can help mitigate the effects of poor image quality. Additionally, combining visual data with complementary information from multispectral or thermal imaging sensors could enable earlier and more accurate detection of diseases, even in challenging conditions.

Second, employing domain adaptation techniques and self-supervised learning could enhance the generalization capabilities of models. These approaches would allow the models to adapt dynamically to varying environmental and crop-specific conditions, thereby improving their applicability in diverse agricultural settings. Lastly, leveraging edge computing and lightweight model architectures could optimize deployment on embedded systems, ensuring that solutions are both computationally efficient and scalable.

6 Conclusion

This study presents a significant advancement in the detection of plant disease by developing a convolutional neural network (CNN) model that offers a generalized and scalable solution. Unlike prior approaches that often focused on specific diseases or crops, the model leverages a diverse dataset and a simplified classification structure, enabling it to accurately identify a wide range of diseases across multiple plant species. These innovations make the model exceptionally robust and practical for real-world agricultural applications.

A key contribution of this work is the creation of the HealthySick Plants dataset, which ensures broad applicability by encompassing various plant species and diseases. By grouping diseases into a single "sick" category and healthy plants into a single "healthy" category, the dataset enhances usability while maintaining scalability. The proposed CNN-based model further distinguishes itself through its superior generalization capabilities, achieving remarkable performance metrics, including an accuracy of 99.53%, an AUC of 99.96%, and strong precision and recall scores. These results set a new benchmark in the field of plant disease detection, emphasizing the model's reliability and adaptability in diverse agricultural environments. The experiments underscore the efficiency of the model, demonstrating minimal misclassifications and consistent performance across both training and validation datasets. This

efficiency extends to its computational design, which replaces multiple specialized models with a single, unified framework. This feature is particularly advantageous for deployment on resource-constrained systems, such as embedded devices widely used in agricultural settings.

Beyond its technical achievements, the model offers practical benefits by enabling real-time detection of plant diseases. By empowering farmers to take timely and informed actions, it minimizes crop losses, reduces the overuse of pesticides, and promotes the efficient use of resources such as water and fertilizers. This adaptability positions the model as a valuable tool for precision agriculture, where continuous monitoring and automated responses are essential for optimizing yields and minimizing environmental impact.

Building on these results, future work will focus on deploying this model within an Internet of Things (IoT) framework capable of continuously monitoring plant health. By integrating sensors and cloud-based systems, this next-generation platform will predict and detect plant diseases in real time, offering an even more comprehensive solution for smart agriculture. Additional research will also explore the integration of thermal and multispectral imaging to further enhance disease detection and classification under varying environmental conditions.

In conclusion, this study marks a significant step forward in the development of intelligent agricultural systems. By providing a scalable, accurate, and adaptable solution, the work lays the foundation for smarter and more sustainable farming practices, addressing critical challenges in food security, resource optimization, and climate resilience.

Abbreviations

The following abbreviations are used in this manuscript:

AUC	Area Under the Curve
BPMN	Business Process Model Notation
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Network
DL	Deep Learning
GIS	Geographic Information Systems
LR	Learning Rate
RGB	Red, Green, Blue
SVM	Support Vector Machine
XAI	eXplainable Artificial Intelligence

Supplementary Information

This article does not include supplementary files. However, if additional data, such as full unprocessed scans or other supporting materials, are required, they can be provided upon request. For journal-specific supplementary requirements, please refer to the relevant guidelines.

Acknowledgements

The authors acknowledge the contributions of Hassan II University, Casablanca, Morocco, for supporting this research. We also extend our gratitude to the University of Liège, Gembloux Agro-Bio Tech (GxABT), DEAL, and the University of Mons, Infortech/Numediart, Faculty of Engineering, ILIA for their collaborative efforts. Special thanks to the LTIM Research Laboratory for providing resources and facilities.

Declarations

Funding

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Conflict of Interest/Competing Interests

The authors declare that they have no conflicts of interest.

Ethics Approval and Consent to Participate

Not applicable.

Consent for Publication

Not applicable.

Data Availability

The datasets generated and analyzed during the current study are available from the corresponding author upon reasonable request.

Materials Availability

Not applicable.

Code Availability

The code used in this study is available upon reasonable request.

References

[1] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G., Davis, A., Dean, J., Devin, M. & Others Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *ArXiv Preprint ArXiv:1603.04467*. (2016)https://arxiv.org/abs/1603.04467

[2] Ahmed, S., Ibrahim, M., Nadim, M., Rahman, M., Shejunti, M., Jabid, T. & Ali, M. MangoLeafBD: A comprehensive image dataset to classify diseased and healthy mango leaves. *Data In Brief*. **47** pp. 108941 (2023)https://doi.org/10.1016/j.dib.2023.108941

[3] Ahlawat, S. & Choudhary, A. Hybrid CNN-SVM classifier for handwritten digit recognition. *Procedia Computer Science*. **167** pp. 2554-2560 (2020)https://doi.org/10.1016/j.procs.2020.03.309

[4] Akbar, M., Ullah, M., Shah, B., Khan, R., Hussain, T., Ali, F., Alenezi, F., Syed, I. & Kwak, K. An effective deep learning approach for the classification of Bacteriosis in peach leaves. *Frontiers In Plant Science*. **13** pp. 1064854 (2022) https://doi.org/10.3389/fpls.2022.1064854

[5] Anand, T., Sinha, S., Mandal, M., Chamola, V. & Yu, F. AgriSegNet: Deep aerial semantic segmentation framework for IoT-assisted precision agriculture. *IEEE Sensors Journal*. **21**, 17581-17590 (2021)10.1109/JSEN.2021.3071290

[6] Amara, J., Bouaziz, B. & Algergawy, A. A deep learning-based approach for banana leaf diseases classification. *Datenbanksysteme Für Business, Technologie Und Web (BTW 2017) - Workshopband*. (2017)https://dl.gi.de/items/13766147-8092-4f0a-b4e1-8a11a9046bdf

[7] Albert, S., Wichtmann, B., Zhao, W., Maurer, A., Hesser, J., Attenberger, U., Schad, L. & Zöllner, F. Comparison of Image Normalization Methods for Multi-Site Deep Learning. *Applied Sciences*. (2023) https://doi.org/10.3390/app13158923

[8] Chaudhary, P., Verma, A., Kukreja, V. & Sharma, R. Integrating Deep Learning and Ensemble Methods for Robust Tomato Disease Detection: A Hybrid CNN-RF Model Analysis. *2024 11th International Conference On Reliability, Infocom Technologies And Optimization (Trends And Future Directions) (ICRITO)*. pp. 1-4 (2024)10.1109/ICRITO61523.2024.10522213

[9] Ferentinos, K. Deep learning models for plant disease detection and diagnosis. *Computers And Electronics In Agriculture*. **145** pp. 311-318 (2018)https://doi.org/10.1016/j.compag.2018.01.009

[10] Mohanty, S. P., Hughes, D. P., and Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, **7**, 1419. doi:10.3389/fpls.2016.01419.

[11] Fuentes, A., Yoon, S., Kim, S. & Park, D. A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*. **17**, 2022 (2017)https://doi.org/10.3390/s17092022

[12] Gao, F. & Zhong, H. Study on the Large Batch Size Training of Neural Networks Based on the Second Order Gradient. (2020)https://arxiv.org/abs/2012.08795

[13] Genaev, M., Skolotneva, E., Gultyaeva, E., Orlova, E., Bechtold, N. & Afonnikov, D. Image-Based Wheat Fungi Diseases Identification by Deep Learning. *Plants*. **10**, 1500 (2021)https://doi.org/10.3390/plants10081500

[14] Goncharov, P., Ososkov, G., Nechaevskiy, A., Uzhinskiy, A. & Nestsiarenia, I. Disease detection on the plant leaves by deep learning. *Advances In Neural Computation, Machine Learning, And Cognitive Research II: Selected Papers From The XX International Conference On Neuroinformatics, October 8-12, 2018, Moscow, Russia*. pp. 151-159 (2019)https://doi.org/10.1007/978-3-030-01328-8_16

[15] M. Shafiq and Z. Gu, "Deep Residual Learning for Image Recognition: A Survey," *Applied Sciences*, vol. 12, no. 18, p. 8972, 2022, doi:10.3390/app12188972.

[16] Hughes, D.P. & Salathe, M. An open access repository of images on plant health to enable the development of mobile disease diagnostics, *arXiv preprint arXiv:1511.08060* (2015)https://arxiv.org/abs/1511.08060

[17] Ioffe, S. & Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. (2015) https://asvk.cs.msu.ru/~sveta/%D1%80%D0%B5%D1%84%D0%B5%D1%80%D0%B0%D1%82/batch_normalization.pdf

[18] Khaleel, M., Sai, P., Kumar, A., Raja, P. & Hoang, V. Rose Plant Leaves: Disease Detection and Pesticide Management using CNN. *2022 6th International Conference On Trends In Electronics And Informatics (ICOEI)*. pp. 1067-1072 (2022)10.1109/ICOEI53556.2022.9776898

[19] Kahloot, K. & Ekler, P. Algorithmic Splitting: A Method for Dataset Preparation. *IEEE Access*. **9** pp. 125229-125237 (2021)10.1109/ACCESS.2021.3110745

[20] Khalid, M. M., and Karan, O. (2024). "Deep Learning for Plant Disease Detection," *International Journal of Mathematics, Statistics, and Computer Science*, vol. 2. doi:10.59543/ijmscs.v2i.8343. Available at: https://ijmscs.org/. ISSN: 2704-1077, eISSN: 2704-1069 https://doi.org/10.59543/ijmscs.v2i.8343.

[21] Mimi, A., Zohura, S., Ibrahim, M., Haque, R., Farrok, O., Jabid, T. & Ali, M. Identifying selected diseases of leaves using deep learning and transfer learning models. *Machine Graphics & Vision*. **32** (2023)10.22630/MGV.2023.32.1.3

[22] Mohanty, S., Hughes, D. & Salathé, M. Using deep learning for image-based plant disease detection. *Frontiers In Plant Science*. **7** pp. 1419 (2016)https://doi.org/10.3389/fpls.2016.01419

[23] Oo, Y. & Htun, N. Plant Leaf Disease Detection and Classification using Image Processing. *International Journal Of Research And Engineering*. (2018)http://dx.doi.org/10.21276/ijre.2018.5.9.4

[24] Raj, M., Gupta, S., Chamola, V., Elhence, A., Garg, T., Atiquzzaman, M. & Niyato, D. A survey on the role of Internet of Things for adopting and promoting Agriculture 4.0. *Journal Of Network And Computer Applications*. **187** pp. 103107 (2021)https://doi.org/10.1016/j.jnca.2021.103107

[25] Ramcharan, A., Baranowski, K., McCloskey, P., Ahmed, B., Legg, J. & Hughes, D. Deep learning for image-based cassava disease detection. *Frontiers In Plant Science*. **8** pp. 1852 (2017)https://doi.org/10.3389/fpls.2017.01852

[26] Saleem, M., Potgieter, J. & Arif, K. Automation in Agriculture by Machine and Deep Learning Techniques: A Review of Recent Developments. *Precision Agriculture*. **22**, 2053-2091 (2021)https://doi.org/10.1007/s11119-021-09806-x

[27] Shorten, C. & Khoshgoftaar, T. A survey on Image Data Augmentation for Deep Learning. *Journal Of Big Data*. **6** pp. 1-48 (2019)https://api.semanticscholar.org/CorpusID:195811894

[28] Singh, G. & Yogi, K. Performance evaluation of plant leaf disease detection using deep learning models. *Archives Of Phytopathology And Plant Protection*. **56** pp. 209 - 233 (2023)https://doi.org/10.1080/03235408.2023.2183792

[29] Sladojevic, S., Arsenovic, M., Anderla, A., Culibrk, D. & Stefanovic, D. Deep Neural Networks Based Recognition of Plant Diseases by Leaf Image Classification. *Computational Intelligence And Neuroscience*. **2016** pp. 11 (2016)https://doi.org/10.1155/2016/3289801

[30] P. K. Tripathy, A. K. Tripathy, A. Agarwal, and S. P. Mohanty, "MyGreen: An IoT-Enabled Smart Greenhouse for Sustainable Agriculture," *IEEE Consumer Electronics Magazine*, vol. 10, no. 4, pp. 57-62, 2021, doi: 10.1109/MCE.2021.3055930

[31] Y. Xu and R. Goodacre, "On Splitting Training and Validation Set: A Comparative Study of Cross-Validation, Bootstrap and Systematic Sampling for Estimating the Generalization Performance of Supervised Learning," *Journal of Analysis and Testing*, vol. 2, no. 3, pp. 249–262, 2018, doi:10.1007/s41664-018-0068-2.

[32] Zhang, K., Zuo, W., Chen, Y., Meng, D., and Zhang, L. (2017). "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, at:https://doi.org/10.1109/TIP.2017.2662206.