

## Article

# PlanteSaine: An Artificial Intelligent Empowered Mobile Application for Pests and Disease Management for Maize, Tomato, and Onion Farmers in Burkina Faso

Obed Appiah <sup>1,2,\*</sup>, Kwame Oppong Hackman <sup>1</sup>, Belko Abdoul Aziz Diallo <sup>1</sup>, Kehinde O. Ogunjobi <sup>1</sup>, Son Diakalia <sup>3,4</sup>, Ouedraogo Valentin <sup>1</sup>, Damoue Abdoul-Karim <sup>5</sup> and Gaston Dabire <sup>4</sup>

<sup>1</sup> Competence Centre, West African Science Service Centre on Climate Change and Adapted Land Use (WASCAL), Blvd Mouammar Kadafi, Patte d'oise, Ouagadougou 06 BP 9507, Burkina Faso; hackman.k@wascal.org (K.O.H.); diallo.b@wascal.org (B.A.A.D.); ogunjobi.k@wascal.org (K.O.O.); ouedraogo.v@wascal.org (O.V.)

<sup>2</sup> Department of Computer Science and Informatics, University of Energy and Natural Resources (UENR), Sunyani P.O. Box 214, Ghana

<sup>3</sup> Directorate of Plant Protection and Packaging (DPVC), Ministry of Agriculture, Animal Resources and Fisheries (MARAH), Ouagadougou 03 BP 7005, Burkina Faso; sondiakalia@yahoo.fr

<sup>4</sup> Gaoua University Center, Nazi BONI University, Bobo-Dioulasso 01 BP 1091, Burkina Faso; gaston.dabire@u-naziboni.bf

<sup>5</sup> Afrique Géosciences, S/C 11 BP 178 Ouaga CMS 11, Ouagadougou, Burkina Faso; akarim.damoue@gmail.com

\* Correspondence: obed.appiah@uenr.edu.gh or appiah.o@wascal.org

**Abstract:** This study presents PlanteSaine, a novel mobile application powered by Artificial Intelligence (AI) models explicitly designed for maize, tomato, and onion farmers in Burkina Faso. Agriculture in Burkina Faso, like many developing nations, faces substantial challenges from plant pests and diseases, posing threats to both food security and economic stability. PlanteSaine addresses these challenges by offering a comprehensive solution that provides farmers with real-time identification of pests and diseases. Farmers capture images of affected plants with their smartphones, and PlanteSaine's AI system analyzes these images to provide accurate diagnoses. The application's offline functionality ensures accessibility even in remote areas with limited Internet connectivity, while its messaging feature facilitates communication with agricultural authorities for guidance and support. Additionally, PlanteSaine includes an emergency alert mechanism to notify farmers about pest and disease outbreaks, enhancing their preparedness to deal with these threats. An AI-driven framework, featuring an image feature extraction phase with EfficientNetB3 and an artificial neural network (ANN) classifier, was developed and integrated into PlanteSaine. The evaluation of PlanteSaine demonstrates its superior performance compared to baseline models, showcasing its effectiveness in accurately detecting diseases and pests across maize, tomato, and onion crops. Overall, this study highlights the potential of PlanteSaine to revolutionize agricultural technology in Burkina Faso and beyond. Leveraging AI and mobile computing, PlanteSaine provides farmers with accessible and reliable pest and disease management tools, ultimately contributing to sustainable farming practices and enhancing food security. The success of PlanteSaine underscores the importance of interdisciplinary approaches in addressing pressing challenges in global agriculture

**Keywords:** artificial intelligence in agriculture; pest and disease detection; mobile application for farmers; Burkina Faso agriculture; real-time plant diagnosis; sustainable farming practices



**Citation:** Appiah, O.; Hackman, K.O.; Diallo, B.A.A.; Ogunjobi, K.O.; Diakalia, S.; Valentin, O.; Abdoul-Karim, D.; Dabire, G. PlanteSaine: An Artificial Intelligent Empowered Mobile Application for Pests and Disease Management for Maize, Tomato, and Onion Farmers in Burkina Faso. *Agriculture* **2024**, *14*, 1252. <https://doi.org/10.3390/agriculture14081252>

Academic Editor: Francesco Marinello

Received: 5 June 2024

Revised: 11 July 2024

Accepted: 19 July 2024

Published: 30 July 2024



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

## 1. Introduction

Agriculture plays a pivotal role in the economies of many developing nations, including Burkina Faso, where a significant portion of the population relies on farming for their livelihood [1]. However, the agricultural sector is constantly under threat from various

biotic stressors, such as pests and pathogens, which can lead to severe crop losses [2]. Plant diseases caused by multiple pathogens can significantly impair crops' normal growth and development, disrupting their vital functions and ultimately reducing yields [3]. Similarly, pests like insects and fungi can wreak havoc on crops, posing a substantial threat to food security [4]. In Burkina Faso, where agriculture forms the backbone of the economy, the challenges posed by plant diseases and pests are particularly acute. Fighting pests and diseases of maize, tomato, and onion crops is vital to the country's food security and economic stability. These crops are susceptible to a range of diseases and pests that can devastate entire harvests, threatening farmers' livelihoods and the availability of food for the population [1].

Addressing these challenges requires efficient and timely intervention to prevent the spread of diseases and pests in agricultural fields. One promising avenue for assisting farmers in this endeavor is utilizing mobile applications empowered by Artificial Intelligence (AI) [5–9] and remote communication capabilities [10]. These tools provide a new dimension to farming, allowing farmers to access expert knowledge and real-time information, empowering them to make informed decisions.

In response to these challenges, we present PlanteSaine, a mobile application developed for Burkina Faso maize, tomato, and onion farmers. PlanteSaine is a groundbreaking initiative that offers a multifaceted solution to the problems of plant disease and pest management in maize, tomato, and onion crops. Our work contributes significantly to the agricultural landscape by harnessing technology to provide farmers with a tool to safeguard their crops. Users of the app can capture pictures of affected plants with their smartphones, and PlanteSaine's AI algorithms analyze the images and provide diagnoses for the issue. The AI functionality of the app is built by using the top layers of EfficientNetB3 as a deep features selection block and an effective ANN as the classifier. This makes the app occupy a smaller memory space than apps developed with EfficientNetB3.

What sets PlanteSaine apart is its offline functionality, a critical feature for regions with limited access to the Internet [3]. Farmers in remote areas of Burkina Faso can utilize the app without relying on a continuous Internet connection. PlanteSaine stores essential data on user devices, ensuring that users (extension officers, farmers, etc.) can access critical information even in areas with poor network coverage. In addition to diagnosis and offline capabilities, the PlanteSaine messaging option allows users to communicate about diseases and pests with pathologists and entomologists at the Ministry of Agriculture. This feature promotes the exchange of insights and best practices, empowering farmers with expertise. Again, PlanteSaine incorporates an emergency alert mechanism that notifies users about pests, disease outbreaks, and other vital information in real time, enhancing preparedness for dealing with epidemics.

The PlanteSaine mobile app represents a significant step forward in agricultural technology, empowering farmers with tools to protect their crops, enhance productivity, and ensure food security in Burkina Faso. This work underscores the importance of leveraging technology for sustainable agriculture and provides a blueprint for similar initiatives in other regions facing similar agricultural challenges [4]. With PlanteSaine, Burkina Faso is well-positioned to enhance its agrarian resilience, improve the well-being of its farming communities, and secure the nation's food supply.

The key contributions of this study are as follows:

- PlanteSaine leverages an advanced AI framework combining EfficientNetB3 with an Artificial Neural Network (ANN) classifier, achieving significantly higher accuracy and performance in diagnosing plant diseases and pests for maize, tomato, and onion crops.
- Designed to operate with or without Internet access, PlanteSaine ensures that farmers in areas of Burkina Faso with limited Internet access can utilize the app's diagnostic capabilities and access critical agricultural information.
- The app includes a robust messaging feature that connects users with agricultural experts, facilitating real-time advice and knowledge sharing. Additionally, an emergency

alert system promptly informs users about pest and disease outbreaks, enhancing preparedness and response.

- With its high accuracy, precision, and recall rates across multiple crops, PlanteSaine presents a scalable and adaptable solution for agricultural disease and pest management. Its successful implementation can serve as a blueprint for similar initiatives in other developing regions, promoting sustainable agriculture and food security.

## 2. Related Works

The growing interest in using AI techniques to recognize or detect plant diseases has been analyzed by researchers (e.g., Singh (2020) [11] and Abade et al. (2021) [12]) in their review works. The works mainly reviewed the effective use of diverse imaging techniques and computer vision approaches for identifying crop diseases. Deep learning (DL), support vector machine (SVM), K-means clustering, and K-nearest neighbors (K-NN) are the major ones used for such tasks. The works also mention that, with the advent of DL, the shift is moving away from classical machine learning-based approaches, which mostly employ image processing techniques for feature extractions. It was observed that most image processing techniques used in recognizing or identifying plant diseases are mainly employed at the preprocessing stage to help improve the quality of the images (dataset) so that the DL model can be trained effectively. Again, the 121 papers reviewed by Abade et al. (2021) [12] concluded that Convolution Neural Network (CNN) has become a state-of-the-art technique for identifying and classifying plant diseases. The papers reviewed by Abade et al. (2021) [12] were published between 2011 and 2021, spanning 10 years. Therefore, they can serve as a critical guide to understanding the domain and direction of DL for plant disease and pest classification, especially with approaches related to disease detection, dataset characteristics, crop and pathogens investigation, and so on. According to the various literature reviews, CNN-based architecture is an impressive solution for crop disease detection and can contribute to shaping the future of computer vision in this direction. In this Section, we present some of the standard works that have employed Deep Learning (DL) for pest and disease identification, the use of the DL model for tomato and maize and other crops' disease and pest identification, and, finally, we explore how deep feature extraction approaches are currently deployed to improve accuracy as well as reduce the resource consumption of AI-enabled with the advent of DL, the shift is moving away from classical machine learning-based approaches mobile apps.

Identification of plant leaf disease using a nine-layer convolutional neural network was deployed by Geetharamani and Arun Pandian (2019) [13] to identify disease. The Deep CNN model was trained using a publicly available dataset with 39 different classes of plant leaves and background images. To improve the model's performance, the authors employed six image augmentation methods: gamma correction, noise injection, image flipping, PCA color augmentation, rotation, and scaling to have a robust model for disease classification. The proposed model achieved 96.46% accuracy. While the proposed model was strong, the work focused on only plant diseases. The authors in [14,15] also proposed CNN models for identifying plant diseases with significant levels of accuracy. Again, the number of classes for each category of plants was small; therefore, new models must be developed to handle higher numbers of diseases per plant category for practical application in real-world scenarios.

Ferentinos (2018) [16] developed a deep-learning model for plant disease detection and diagnosis. Specialized deep-learning models were trained to identify diseased or healthy leaves. In the work, several CNN-based models were implemented to determine which one effectively distinguishes healthy leaves from diseased ones. These models (binary classifiers) were designed to identify one out of two possible states (healthy or diseased) of the leaf. VGG architecture achieved 99.53% accuracy, topping all the models used in the experiment. Though the results produced by Ferentinos are impressive, current datasets warrant a multi-class model for identifying plant diseases.

Agarwal et al. (2020) [17] developed a robust model for identifying diseases in tomato crops. According to Agarwal et al. (2020) [9], pre-trained CNN models perform reasonably well but are computationally expensive (running time and memory demand). Using a dataset from PlantVillage [18], a publicly available dataset, the proposed lightweight model achieved 98.4% accuracy, performing better than the traditional machine learning techniques relying on handcrafted image feature extraction techniques and some pre-trained models. Interestingly, VGG16, a pre-trained model, achieved 93.5% accuracy, suggesting that lightweight models can deliver the required solution at some point. It must be mentioned that new pre-trained models have recently been developed to help image classification tasks with impressive accuracies. For example, the EfficientNet family of pre-trained models has demonstrated its effectiveness in computational complexity and high accuracy at fewer epochs during the models' training. Developing solutions using one of the state-of-the-art pre-trained models will ensure that such solutions provide acceptable outcomes. However, the generalization of developed models can sometimes be challenging.

The use of hybrid DL-based and ensemble classifier models to improve the classification of diseases in tomato crops has been demonstrated by [19–23]. The concept of the ensembled classifiers is to amalgamate the outcomes of weak and relatively more robust models, thereby creating a new classifier known to yield superior results compared to individual classifiers. These ensemble methods operate faster and more effectively and hold promise for AI models on resource-constrained devices. Given the current emphasis within the research community on rapid and accurate methods for diagnosing and classifying various plant types and their diseases, this development is significant [11]. The models were trained on the publicly available PlantVillage dataset, consisting of tomato images, and subsequently tested using publicly accessible Taiwan tomato leaf data. The most successful ensemble technique achieved an impressive accuracy of 95.98% on the tomato dataset. However, it is worth noting that the dataset used in this study solely comprises images of tomato leaves, indicating a limitation in identifying diseases affecting fruits and pests attacking tomato plants.

The challenge of achieving high accuracy performance for multiclass disease detection influenced the design of a new approach to preprocessing images for Deep Learning models [24]. In their work, anisotropic filtering was applied to all images during the preprocessing stages to remove unwanted distortions in images. U-net was further used to segment an image into background and foreground, where the foreground pixels were accepted as pixels belonging to regions of interest. An accuracy of 92.4% was recorded for the Dbnet/Alexnet model on segmented images, which was impressive compared to the unsegmented images. Bacterial spot, healthy, late blight, early blight, leaf mold, tomato mosaic virus, tomato yellow leaf mold, tomato mosaic virus, and tomato yellow leaf curl virus were the classes used to train the model. The classes used to develop the model were only diseases that affect the leaves of the tomato. As much as the work significantly contributes to developing DL models for plant pathology, the work did not consider diseases and pests that affect tomato fruits. However, more can be done to achieve robust models for identifying tomato diseases and pests [25].

Sathiya et al. (2022) [26] claimed that the current disease detection methodologies are unsuitable for real-time field scenarios because they are subject to prediction errors. This may be attributed to the controlled environment setup during the creation of the dataset (image acquisition), which may be different during the usage of the model. The study proposed using soft computing techniques (CDD-H\_HSC) to automatically identify diseases, especially at the earlier stages where most existing solutions fail. Badiger et al. (2023) [24] and Sathiya et al. (2022) [26] also employed segmentation to separate the diseased area from the input plant leaf image at the preprocessing stage using the Multi-Swarm Coyote Optimization (MSCO) technique. To deal with the curse of dimension in model development, the Improved Chan–Vese Snake Optimization (ICVSO) algorithm is employed to minimize the dimensionality of the features for training. The Fitness-Distance Balance Deep Neural Network (FDB-DNN) classifier model identifies the leaf and its

associated disease. The work demonstrates the benefit of segmentation in improving the identification of disease. The stages involved in this work make it a bit complicated to deploy on low-resource devices such as smartphones.

Shewale et al. (2023) [27] proposed a deep-learning architecture for early plant leaf disease detection. The significant importance of such architecture or model is its ability to detect disease early, enabling the farmer to take action before he loses all his crops when the disease appears late. The model was developed to identify and classify diseases in tomato leaves. The strength of the proposed model in their work is its ability to generalize well compared to existing models deployed in real-time. The model was, however, not integrated into any device for farmers to use in the field for the identification or classification of diseases, as mentioned by the authors. There is, therefore, the need to work on models that can easily be deployed on devices such as mobile phones for adoption by farmers. The work again acknowledged that numerous approaches are available for automatic disease recognition. However, more needs to be undertaken to make tools easy and accessible to farmers all over the world.

Using CNN to classify maize leaf disease has also gained momentum over time [28,29]. A prominent one was proposed in the work by Priyadharshini et al. (2018) [30]. The work modified the LeNet, a CNN-based architecture, to develop a model that can identify maize disease with an accuracy of 97.89%. The authors claim that CNN models are effective models for classification tasks; therefore, more experiments can be conducted to develop models that can achieve the perfect level of accuracy. The work demonstrated this by experimenting with various kernel sizes and depths. In the study, the authors concluded that kernels of sizes 3, 9, and 3 are better suited for maize leaf disease classification. A major limitation of the classifier is that the number of diseases used to train the model needs to be bigger to deploy such a model for wide acceptability. A dataset containing a significant number of maize diseases and pests can help develop comprehensive AI models that can effectively identify pests and diseases and, therefore, can be deployed for farmer usage.

Esgario et al. (2020) [31] presented a paper on a mobile app that assists farmers in identifying pests and diseases of coffee leaves. The app has an AI model developed using deep learning architecture to make it possible to identify pests and diseases effectively. The authors achieved these remarkable results by designing a Computer Vision pipeline that segments and classifies leaf lesions and estimates the severity of stress caused by biotic agents in coffee leaves. ResNet50 is deployed for this solution and biotic stress classification; the accuracy rates were greater than 97%. A major limitation acknowledged by the researchers is that their proposed framework could only work when the smartphone with the app was connected to the Internet. This challenge has been observed with several mobile app solutions, such as Plantix [32]. Indeed, there is a trade-off between accuracy and running time, and therefore, lightweight CNN models that can fit on mobile phones cannot predict as perfectly as those that are run on the web server. Nonetheless, there is the need to develop an app that farmers can use without the need to connect to access the Internet since many farmers in developing countries such as Burkina Faso will not be able to benefit from the power of AI in transforming the domain of agriculture if Internet connectivity is always needed for this task.

Jonathan et al. (2022) [33] presented an innovative in-field plant disease detection solution using a smartphone's deep learning-based trained AI model. Plant Pathology on Palms deployed models on mobile phones. In the work, a lightweight deep learning-based model is developed and integrated into a mobile phone for identifying diseases. As mentioned earlier, the reason for developing lightweight models is to mitigate the challenge of effectively running huge CNN models on resource-limited devices. The proposed framework by Jonathan consumes about 1 MB and achieved 97%, 97.1%, and 96.4% accuracy on apple, citrus, and tomato leaves datasets, respectively. One of their tiny models achieves 93.33% accuracy. The framework was compared with three variants of MobileNetV2 and demonstrated promising performance stability by outperforming most of the settings provided for the MobileNetV2 variants. The work contributes significantly



to the discussion of using smartphone and edge computing devices for the real-time classification or identification of diseases. However, the solution focused on three crops and some diseases that affect these crops. Expanding the dataset and developing alternative models that will be robust on the expanded dataset will also contribute to pushing machines to assist in the early detection of pests and diseases and, therefore, help farmers take the necessary action before they lose their crops completely.

Kaushik et al. (2023) [34] proposed a deep learning model-based framework for mobile applications (TomFusioNet) to identify and classify tomato diseases and pests. TomFusioNet's pipeline comprises two modules, namely, DeepRec and DeepPred. DeepRec provides preliminary disease recognition results, while DeepPred further identifies the type of disease in the crop. A multi-objective optimization-based, non-dominated sorting genetic algorithm was employed for hyperparameter tuning of DeepPred. The proposed models, DeepPred and DeepRec, achieved an average accuracy of 99.93% and 98.32%, respectively. TomFusioNet demonstrates a superior performance with an AUC value of 99.10% and a convergence loss of 0.021. The framework, however, was developed to analyze leaves of tomato crops for disease identification and classification.

Nag et al. (2023) [35] have also proposed a mobile app-based tomato disease identification using fine-tuned CNNs. The study employed SqueeXNet 1.1, VGG19, AlexNet, ResNet-50, and DenseNet-121 on a combined created image and images from PlantVillage. DenseNet-121 leads the pack of CNN architectures with 99.85% accuracy. The work primarily focused on the diseases that affect the leaves. The need for an Internet connection to operate the mobile app can sometimes be challenging, especially in regions where Internet connectivity is poor.

Using pre-trained models for developing models for plant pests and disease detection presents a viable alternative to deploying DL on mobile phones [36]. Wang et al. (2023) [37] adopted EfficientNetB0 to develop a model to classify diseases in tomato plants. The developed model could achieve 91.4% average accuracy, which the authors believe is impressive. Implementing the pre-trained model in a mobile application also exhibited satisfactory computational complexities (time and space) compared with other deep learning models implemented on mobile or smartphones. The authors acknowledged the necessity of improving the model's accuracy and reliability to encourage farmers to accept the app wholeheartedly as a critical tool for productivity. The challenge again is the sharp drop in performance of models developed using laboratory-captured images (images captured under controlled lighting and uniform background conditions) [38–40]. This has led to the recent direction of dataset creation for training models that can effectively classify diseases in the wild. The need to, therefore, have a dataset that largely mimics the field environment is crucial for the adaptability of mobile apps for plant diseases and pest classification [40]. Ngugi et al. (2020) [40] created a dataset using such conditions and deployed a framework with a segmentation mechanism to separate regions of interest from the background. The proposed algorithm records a 94.39% mean F1 score against the rest. Wang (2023) [22] demonstrates the efficacy of the EfficientNet family of models in the classification of diseases. Developing apps that integrate several EfficientNet models for identifying pests and diseases in three or more cropping systems will result in a high demand for computational resources by the app. Leveraging the strength of EfficientNet and other techniques can help improve the development of efficient apps.

Deep-feature extraction techniques to extract the critical characteristics of images and train classifier models such as SVM, KNN, and Extreme Learning Machines (ELM) have also gained popularity due to outstanding performances. Türkoğlu et al. (2019) [41] used such an approach for plant disease and pest detection. In their work, FC6 layers of AlexNet, VGG16, and VGG19 models produce better accuracy scores when compared with the others. EfficientNet models have proven superior to the model proposed by these authors and, therefore, can be deployed to extract features to train models for classifying plant pests and diseases.

In this work, we propose a framework that uses an EfficientNetB3 deep feature technique and a customized ANN to effectively classify pests and diseases in onion, tomato, and maize farms. The framework reduces computation complexity and improves accuracy compared with the state-of-the-art EfficientNetB3 model.

### 3. Materials and Methods

The Methodology Section of this paper outlines a comprehensive approach for developing and evaluating a novel framework for crop pest and disease classification. Central to this approach is implementing the EfficientNetB3 architecture, characterized by its pre-trained base block and head block for feature extraction and classification, respectively. The model's training parameters, including optimizer, learning rate, and loss function, are meticulously defined to ensure optimal performance. An early stopping mechanism is employed during training to address challenges such as underfitting and overfitting. Furthermore, the methodology involves collecting and annotating diverse datasets collected using mobile phones with high-resolution cameras. Image augmentation techniques are applied to enhance dataset diversity and prepare it for robust machine learning model training. The evaluation process involves comparing the proposed framework with baseline models and assessing the effectiveness of image segmentation techniques. Finally, the PlanteSaine mobile app, integrated with the developed classifiers, facilitates real-time image capture and analysis, offering a user-friendly interface for farmers and extension officers.

#### 3.1. Dataset

##### 3.1.1. Image Capturing

The dataset was gathered using mobile phones as the primary data collection tool. The mobile phones employed for this purpose featured a camera with a resolution of 12 megapixels, a wide-angle lens, and a CMOS sensor, with the ability to capture images at a speed of 30 frames per second. It had a storage capacity of 128 GB, offering ample space to store the collected data. The mobile phone was also equipped with connectivity options, including Wi-Fi and Bluetooth, and operated on the Android operating system. It also had GPS capabilities for geotagging the data. The KoboToolbox platform was deployed for this exercise and enabled the automatic transmission of the field photos and their respective metadata for further processing. Fifteen (15) extension officers from the Ministry of Agriculture were tasked to collect images from the three cropping systems (i.e., tomato, onion, and maize) in three regions (Plateau Central, Centre Ouest, and Central Sud) of Burkina Faso.

##### 3.1.2. Image Labeling

Following data collection, the images were obtained from Kobotoolbox and subsequently annotated using the Computer Vision Annotation Tool (CVAT 2.0.0). Over 29,000 were submitted by extension officers within 12 months spanning the two main seasons in the country (rainy and dry). Images were preprocessed by deleting blurred and poorly captured images.

##### 3.1.3. Data Augmentation

The generator's parameters and formulas define the augmentation methods, which are deliberate choices to enhance the dataset's diversity and prepare it for robust machine learning model training in computer vision applications. Table 1 presents the image augmentation techniques employed in this work, and each parameter serves a specific purpose in addressing the challenges and variations encountered in real-world scenarios.

**Table 1.** Image Augmentation Techniques.

Generator	Parameter/Formula
Rotation range	15
Width shift range	0.1
Height shift range	0.1
Brightness range	[0.5, 1.5]
Horizontal flip	True
Vertical flip	True
Shear range	0
Zoom range	0.2
Grayscale Image	Grayscale = 0.299R + 0.587G + 0.114B
Mean Filter	$I_{\text{filtered}}(i,j) = \frac{1}{m \times n} \sum_k = -aa \sum l = -bbI(i + k, j + l)$ Window Size: $7 \times 7$

The rotation range (15 degrees) introduces variability in object orientations, making the model adaptable to objects appearing at different angles. Width and height shift ranges (0.1) allow the model to recognize objects from various perspectives, mimicking shifts in object positions within the frame. Brightness range ([0.5, 1.5]) accounts for changing lighting conditions, a common real-world challenge, while horizontal flipping and the exclusion of vertical flipping (true and false, respectively) simulate different object viewpoints, valuable for orientation-independent object recognition. Shear range (0) is included to handle distorted objects, and zoom range (0.2) trains the model to recognize objects at varying scales, which is relevant in situations where objects are at different distances.

The grayscale image conversion reduces computational complexity when color is not a distinguishing factor. Applying a mean ( $7 \times 7$ ) filter smooths the data and reduces noise, enabling the model to focus on critical features while minimizing the impact of irrelevant variations. In combination, these augmentation methods create a diverse dataset, helping the deep learning models to generalize effectively to the complexities of capturing images in farms by different farmers with diverse camera specifications, lighting conditions, and orientations.

#### 3.1.4. Description of Dataset

Tables 2–4 present the dataset summary used to develop the proposed framework. Sample images of maize, tomato, and onions are displayed in Figures 1–3, respectively.

**Table 2.** Maize Pest and Diseases.

Class	Number of Images
chenille_légionnaire_a	1100
chenille_légionnaire_p	1100
curvulariose_d	1100
feuille_saine	1100
helminthosporiose_d	1100
maladie_abiotique_d	385
pucerons_p	220
rouille_d	649
striure_d	1100

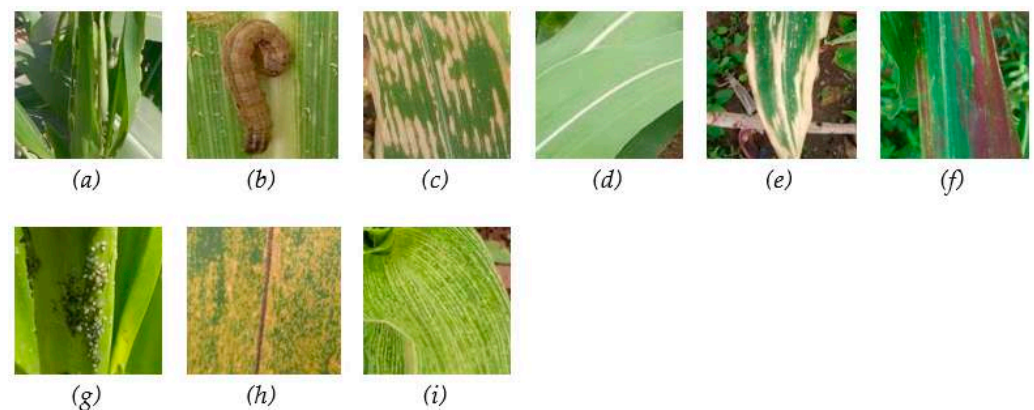


**Table 3.** Tomato Pest and Diseases.

Class	Number of Images
acarien_d	1100
alternariose_acarien_d	1100
alternariose_d	1100
coup_du_soleil_d	1100
exces_d'azote_d	1100
feuille_saine	1100
flétrissement_bactérien_d	1100
fruit_sain	1100
fusariose_d	1100
helicoverpa_armigera_p	1100
mildiou_de_la_tomate_d	1100
pourriture_apicale_d	1100
tuta_absoluta_p	1100
virose_d	1100

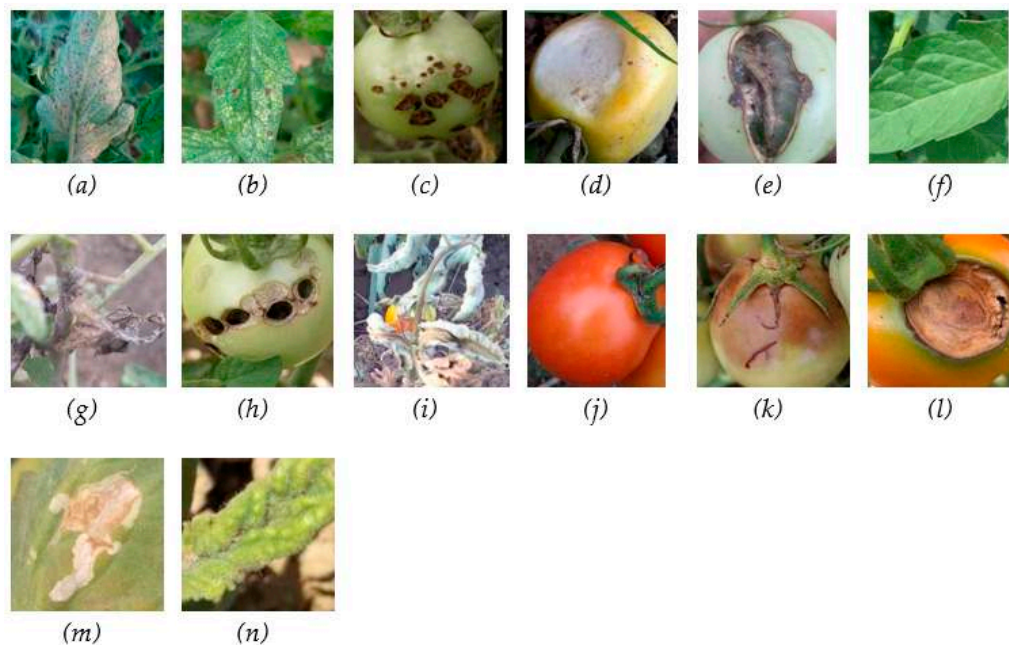
**Table 4.** Onion Pest and Diseases.

Class	Number of Images
alternariose_d	2200
chenilles_p	2200
feuille_saine	2200
fusariose_d	2200
pourriture_des_bulbes_d	505
virose_d	609

**Figure 1.** Sample images from the maize dataset ((a) chenille légionnaire\_a, (b) chenille légionnaire\_p, (c) curvulariose, (d) feuille saine, (e) helminthosporiose, (f) maladie abiotique, (g) pucerons, (h) rouille, (i) striure).

In this study, we employed a carefully designed data-splitting strategy to effectively train, validate, and test our machine learning models. The dataset was divided into three subsets: a training set, a validation set, and a testing set. The training set, comprising approximately 70% of the total data, was used to train the models' parameters and learn the underlying patterns in the data. The validation set, representing around 15% of the data,

played a crucial role in fine-tuning model hyperparameters and evaluating performance during the training to prevent overfitting. Finally, the testing set, consisting of the remaining 15% of the data, was kept completely separate and used only once to assess the final model's generalization ability on unseen data. This careful data partitioning ensured robust model development.



**Figure 2.** Sample images from the tomato dataset ((a) acarien, (b) alternariose acarien, (c) alternariose, (d) coup du soleil, (e) excès d'azote, (f) feuille saine, (g) fusariose, (h) helicoverpa armigera, (i) flétrissement bactérien, (j) fruit sain, (k) mildiou de la tomate, (l) pourriture apicale, (m) tuta absoluta, (n) virose).



**Figure 3.** Sample images from the Onion dataset ((a) alternariose, (b) chenilles, (c) feuille saine, (d) fusariose, (e) pourriture des bulbes, (f) virose).

### 3.2. Baseline Model (EfficientNetB3) Architecture

The EfficientNetB3 model, depicted in Figure 4, comprises two key components: a pre-trained base block featuring convolutional layers for feature extraction and a head block with dense layers for classification. The base block employs a  $3 \times 3$  kernel for convolution and incorporates a ReLU activation layer, followed by downsampling using a Maxpooling layer. Meanwhile, the head block includes a fully connected layer with several neurons representing the target classes. This is succeeded by a dropout layer and a batch normalization layer, enhancing the model's training stability. A SoftMax layer determines the model's output, and training is conducted utilizing the Adamax optimizer with a learning rate set at 0.01, coupled with the categorically cross-entropy loss function. The hyperparameters of the EfficientNetB3 deployed in this work are presented in Table 5.

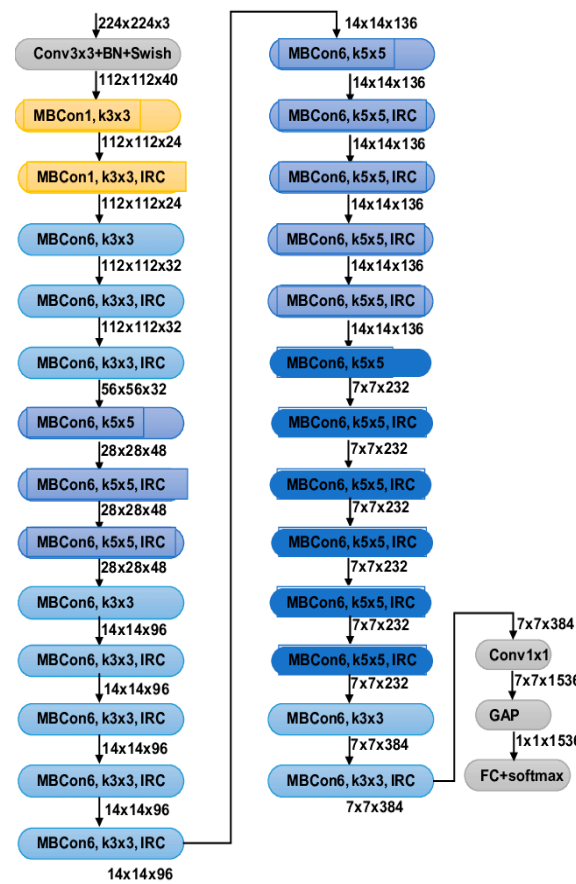


Figure 4. Baseline Architecture (EfficientNetB3).

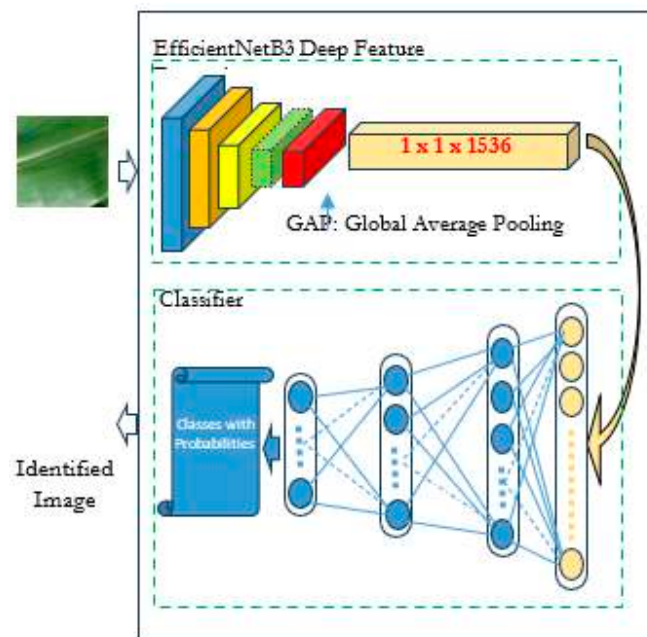
Table 5. Hyperparameter definition for the EfficientNetB3.

Training Parameters	Parameters Values
Loss Function	categorically_crossentropy
Optimizer	Adamax
Epsilon	0.001
Learning Rate	0.01
Batch Size	32
Epoch	40

An early stopping mechanism was implemented during the model training to prevent underfitting or overfitting. The minimum required improvement of 0.001 is set by the model’s min\_delta value to determine when to continue training. To control the training duration, we assigned a patience value of 5 to specify the number of consecutive epochs without significant improvement before halting the training. Additionally, we enabled the option to restore the best-performing model weights, ensuring that the optimal weights were retained throughout the training process.

### 3.3. Proposed Architecture

The proposed architecture utilizes two sequential steps for the classification tasks. Figure 5 presents the conceptual framework of the proposed architecture.



**Figure 5.** Proposed Framework.

In step one, the EfficientNetB3 model is employed as the backbone for deep feature extraction. EfficientNetB3, pre-trained on the ImageNet dataset, is known for its superior accuracy and efficiency, making it an ideal choice for feature extraction tasks. The model configuration is specifically tailored to optimize performance while maintaining computational efficiency. The base EfficientNetB3 model is instantiated with the following parameters:

**Weights:** The model is initialized with weights pre-trained on the ImageNet dataset, leveraging the extensive learning from millions of images.

**Include Top:** The top layer is excluded to facilitate feature extraction rather than classification. This ensures that the output consists of high-dimensional feature vectors rather than class probabilities.

**Input Shape:** The input shape is set to (100, 100, 3), indicating that the model processes RGB images of size  $100 \times 100$  pixels.

**Pooling:** Global average pooling is applied to condense the spatial dimensions of the feature maps, resulting in a compact feature vector for each image.

The second phase, which performs the classification, is an Artificial Neural Network (ANN) model within the framework. The classifier's input shape is the same as the feature vector generated during the first stage. The classifier parameters are as follows:

**Input Layer:** The model starts with an input layer that accepts features with a shape of 1536, matching the output of the GAP layer from the EfficientNetB3 model. This ensures compatibility with the preprocessed feature representations.

**First Hidden Layer:** A dense layer with 256 neurons and ReLU activation (relu) is added. This layer is responsible for learning high-level feature representations. The choice of 256 neurons balances the trade-off between complexity and computational efficiency.

**Second Hidden Layer:** A subsequent dense layer with 64 neurons and ReLU activation further refines the feature representations. The smaller number of neurons helps capture more abstract features while preventing overfitting.

**Output Layer:** The final dense layer has several neurons equal to the number of target classes, with SoftMax activation. This layer outputs the class probabilities for multi-class classification tasks.

**Model Compilation:** The model is compiled using the Adam optimizer and categorical cross-entropy loss function. This compilation configuration ensures efficient training and robust performance.

### 3.3.1. Training of EfficientNetB3

The training process of the baseline model (EfficientNetB3) was meticulously designed to ensure optimal performance and generalization. Hyperparameters such as the learning rate, batch size, and optimization algorithm were carefully chosen to facilitate efficient model convergence. The learning rate was set within the range of 0.001 to 0.0001. This range allowed for a balance between rapid convergence and stability during training iterations. A batch size of 16 to 32 was selected, leveraging the scalable architecture of the EfficientNetB3 model to process multiple samples per iteration effectively. The optimization algorithm chosen for the experiment was Adam due to its adaptive learning rates and momentum capabilities. Adam's ability to handle sparse gradients and adjust learning rates dynamically contributed to faster convergence and improved training efficiency. Regularization techniques were employed to handle overfitting and enhance the model's generalization. Dropout regularization with a rate of 0.1 to 0.5 was applied to prevent the network from over-relying on specific neurons and encourage robust feature learning. Additionally, L2 regularization, also known as weight decay, was used to penalize large weight magnitudes and promote smoother weight distributions, thereby reducing model complexity and improving generalization.

### 3.3.2. Training of Classifier of the Proposed Framework

The Global Average Pooling (GAP) of the EfficientNetB3 is saved as descriptors of the images and subsequently used for training an Artificial Neural Network (ANN) model. The ANN has two hidden layers and ReLU activation functions were meticulously adopted to optimize performance and prevent overfitting. Hyperparameters, including the learning rate (ranging from 0.001 to 0.0001) and batch size (typically set at 16 to 32), were carefully chosen to strike the balance between rapid convergence and stability during training iterations. ReLU activation functions were applied in the hidden layers to introduce non-linearity and facilitate the model's ability to learn complex patterns within the data. Dropout regularization, with rates ranging from 0.1 to 0.5, was incorporated to prevent overfitting by randomly omitting units during training, encouraging the network to learn more robust features. The Adam optimization algorithm was employed due to its adaptive learning rates and momentum capabilities, aiding in faster convergence and improved training efficiency. Overall, these strategies contributed to achieving optimal performance and preventing overfitting during the training phase of the ANN model.

## 3.4. Evaluation Methods

In the context of this research, we employed two primary evaluation methodologies. The first pertains to assessing our proposed framework with the baseline EfficientNetB3 model. The second method involves evaluating the effectiveness of the image segmentation techniques employed to extract regions of interest, thereby enabling the models to classify instances of pests or diseases effectively.

Evaluating the EfficientNetB3 model and the Proposed Framework on maize, tomato, and onion datasets entails using various performance metrics, including Accuracy, Recall, Precision, and F-measure. These statistical measures were chosen due to their widespread use in deep learning research to assess model performance. During the testing phase, these evaluation metrics serve as the evaluative criteria to gauge the efficiency and effectiveness of the generated models when tested with previously unobserved data.

**Accuracy (A):** Ratio of correctly classified samples to total samples. The accuracy metric quantifies the proportion of correctly classified instances by a machine learning model, offering a straightforward evaluation of its overall performance. The Accuracy formula is presented in Equation (1)

$$\text{Accuracy} = (\text{True Positive} + \text{True Negative}) / (\text{Total Sample}) \quad (1)$$



**Precision:** Precision is used to quantify the accuracy of positive predictions made by classification models. It focuses on the proportion of true positive predictions (correctly identified positive cases) relative to all positive predictions made by the model. In other words, precision assesses how many of the positive predictions made by the model were correct. The Precision formula is presented in Equation (2).

$$\text{Precision} = (\text{True Positive}) / (\text{True Positive} + \text{False Positive}) \quad (2)$$

**Recall:** This determines the proportion of correctly predicted positive observations made by the proposed models. The Recall formula is presented in Equation (3).

$$\text{Recall} = (\text{True Positive}) / (\text{True Positive} + \text{False Negative}) \quad (3)$$

**F-Measure** calculates the arithmetic or harmonic mean of precision and recall. This ensures an equal class distribution of the model performance. The F-Measure formula is presented in Equation (4).

$$\text{F1-Score} = 2 \times (\text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall}) \quad (4)$$

### 3.5. *PlanteSaine Mobile App*

The *PlanteSaine* app enables users to capture images using the device's camera. Camera functionality is implemented to allow real-time image capture for analysis. Additionally, the application supports image segmentation, allowing users to submit the entire image or specific sections for analysis. Image processing techniques are employed to extract and preprocess image segments. This functionality permits users to preprocess images containing complex backgrounds by selecting sections in images they want the model to diagnose. This unique functionality, unlike bounding boxes or cropping, enables users to submit pieces of sub-images of the captured image in any possible way for the model to identify the problem. This novel approach reduces the chances of submitting noise to the model and enhances the application's performance. The three (3) models (*MaizeClassifier*, *TomatoClassifier*, and *OnionClassifier*) developed using the proposed classifier are integrated into the application by converting them into tensorflow lite (.tflite) objects for the Android application. The implementation focuses on optimizing the user experience by ensuring fast and responsive model predictions and providing informative feedback. The app was tested by using segmented and non-segmented images to determine which makes it easier for classification by the app.

## 4. Results and Discussion

The results Section presents a detailed examination of the performance evaluation conducted on the EfficientNetB3 model and the Proposed Framework across maize, tomato, and onion crops. The performance metrics, encompassing precision, recall, and F1 score, attained by each model in detecting pests and diseases affecting these crops are presented in this section. The Proposed Framework consistently outperforms EfficientNetB3 across all crop types, demonstrating superior accuracy and reliability in identifying pests and diseases. Additionally, evaluating the *PlanteSaine* mobile app showcases its innovative features, such as the ability to designate regions of interest within images, contributing to enhanced classification accuracy and user experience. Overall, these results highlight the effectiveness of AI-driven approaches in agricultural pest and disease management.

### 4.1. *Pest and Disease Detection Model's Performance*

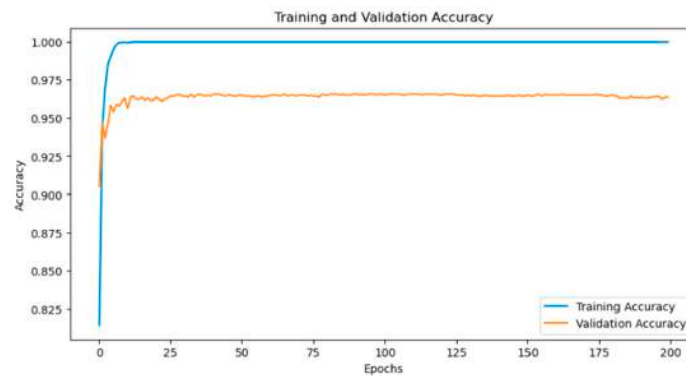
#### 4.1.1. Maize Pest and Diseases Detection Performance

The results in Table 6 illustrate the performance of EfficientNetB3 and the Proposed Framework in detecting pests and diseases affecting maize plants. The Proposed Framework demonstrates superior performance across all evaluated metrics, achieving higher accuracy, precision, recall, and F1 scores when compared with EfficientNetB3. Specifically,

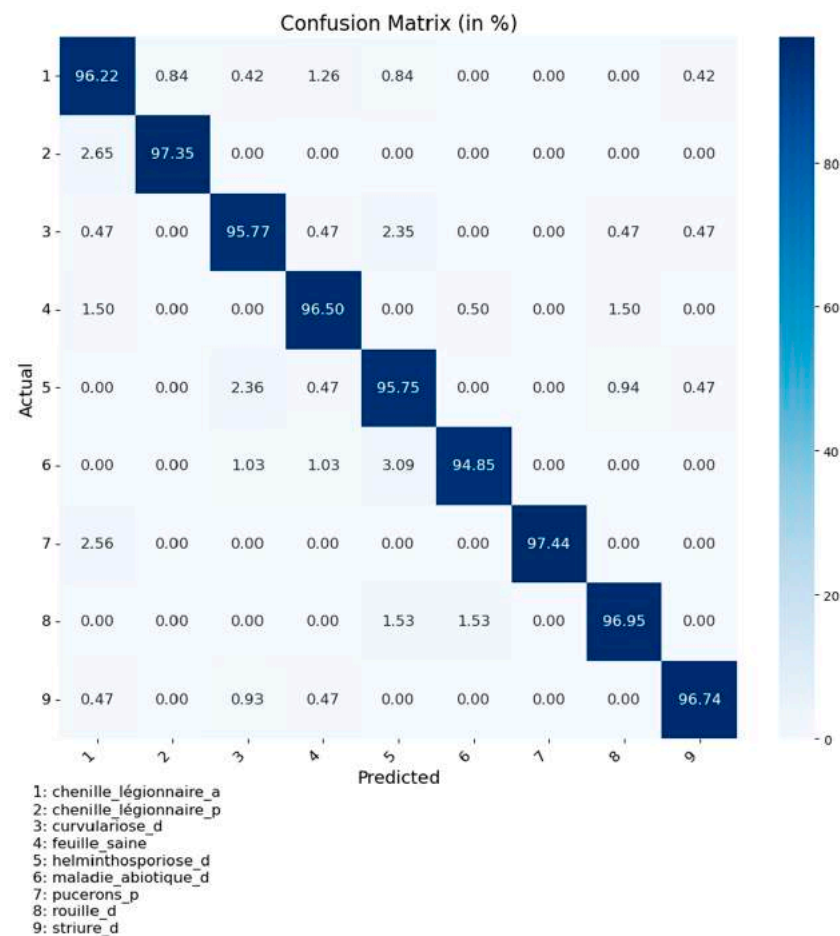
the Proposed Framework attained an accuracy of 96.5, precision of 96.71%, recall of 96.39%, and F1 score of 96.54%, indicating its efficacy in accurately identifying maize pests and diseases from imagery data. The Training and Validation Accuracies and the Confusion Matrix for evaluating the MaizeClassifier are illustrated in Figures 6 and 7, respectively.

**Table 6.** Maize Pest and Disease Detection Performance.

Maize Model	Accuracy	Precision	Recall	F1
EfficientNetB3	89.3	86.8	91.9	88.3
Proposed Framework	96.5	96.7	96.4	96.5



**Figure 6.** Training and Validation Accuracy for Maize Classifier.



**Figure 7.** Testing accuracies for each class in the Maize dataset (Average accuracy = 0.96).

The EfficientNetB3 model achieved an accuracy of 89.3%, while the Proposed Framework significantly improved this metric, reaching 95.5% accuracy, as illustrated in Table 6. This substantial increase underscores the enhanced performance and reliability of the Proposed Framework in classifying maize pests and diseases. Additionally, the precision of the Proposed Framework (96.71%) far exceeds that of EfficientNetB3 (86.8%), indicating a remarkable reduction in false positives. This higher precision is crucial in pest and disease management to avoid misclassification.

Although EfficientNetB3 achieves a relatively high recall rate of 91.9%, the Proposed Framework attains an even higher recall rate of 96.39%, highlighting its ability to identify a greater proportion of actual maize pest and disease instances. This minimizes the risk of overlooking potential threats during monitoring and management activities. The F1 score further supports the superiority of the Proposed Framework, achieving 96.54% compared to EfficientNetB3's 88.3%. This superior F1 score demonstrates that the Proposed Framework maintains a stronger overall performance, making it a robust and reliable model for maize pest and disease detection tasks.

#### 4.1.2. Tomato Pest and Diseases Detection Performance

Table 7 presents the performance metrics for tomato pest and disease detection achieved by EfficientNetB3 and the Proposed Framework. Like the maize detection results, the Proposed Framework outperforms EfficientNetB3's recall and F1 score. With an accuracy of 97.7, precision of 97.68%, recall of 97.72%, and F1 score of 97.69%, the Proposed Framework demonstrates its effectiveness in accurately detecting tomato pests and diseases. Training and Validation Accuracy and the Confusion Matrix for testing the TomatoClassifier are illustrated in Figures 8 and 9, respectively.

**Table 7.** Tomato Pest and Diseases Detection Performance.

Tomato Model	Accuracy	Precision	Recall	F1
EfficientNetB3	83.8	83.3	84.4	82
Proposed Framework	97.7	97.7	97.7	97.6

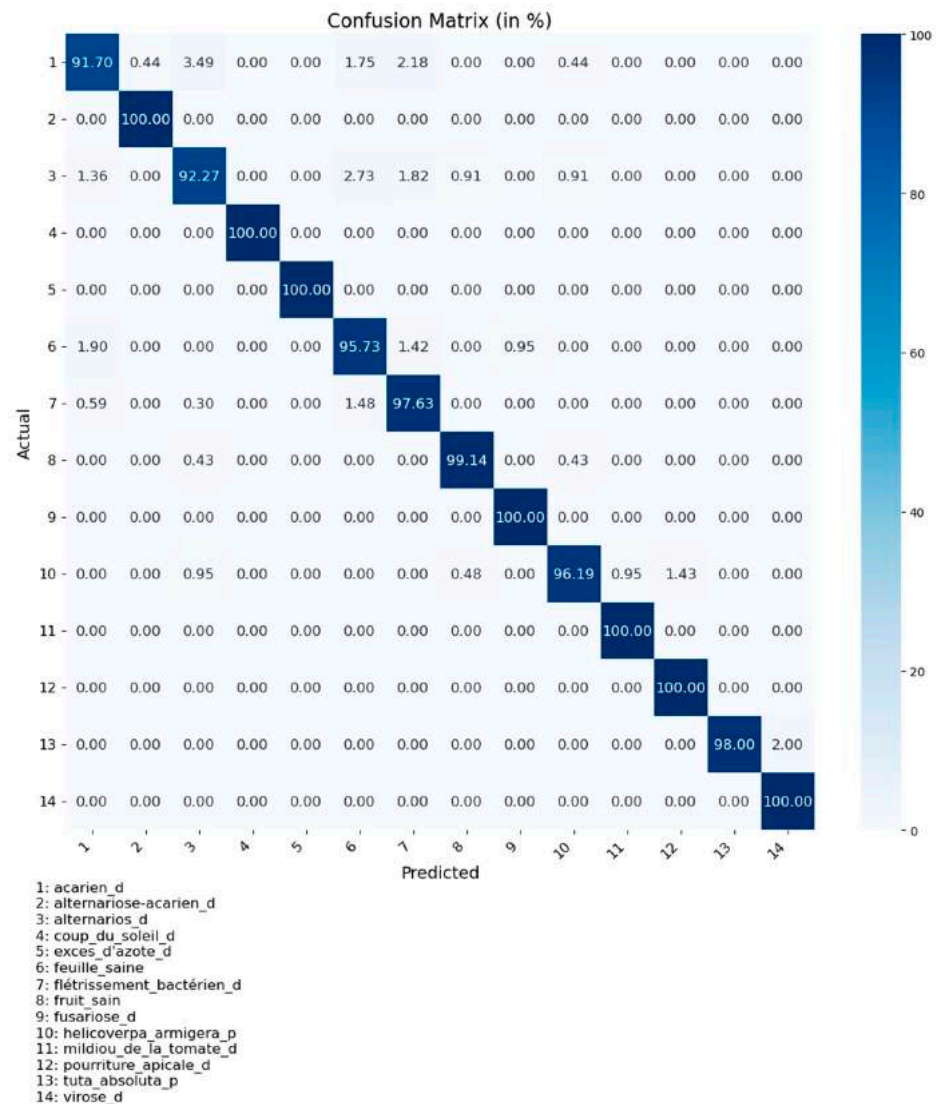


**Figure 8.** Training and Validation Accuracy for Tomato Classifier.

The EfficientNetB3 model achieved an accuracy of 83.8%, with precision, recall, and F1 scores of 83.3%, 84.4%, and 82%, respectively. In contrast, the Proposed Framework demonstrated remarkable improvements, achieving an accuracy of 97.7%, with precision, recall, and F1 scores all at or above 97.7%. This showcases the superior performance of the Proposed Framework in accurately identifying tomato-related tasks, significantly reducing false positives, and enhancing decision-making in pest and disease management.

The Proposed Framework's exceptional recall rate of 97.72% indicates its ability to identify most actual tomato pest and disease instances, ensuring comprehensive threat detection and minimizing the risk of overlooking potential outbreaks. Its high F1 score of 97.69% highlights the robust balance between precision and recall, far surpassing EfficientNetB3's F1 score of 82%. These metrics underscore the Proposed Framework's reliability

and effectiveness, making it a compelling choice for managing pests and diseases in tomato cultivation, ultimately preserving tomato yields and quality.



**Figure 9.** Testing accuracies for each class in the Tomato dataset (Average accuracy = 0.98).

#### 4.1.3. Onion Pest and Diseases Detection Performance

The performance of EfficientNetB3 and the Proposed Framework in onion pest and disease detection is summarized in Table 8. Once again, the Proposed Framework shows superior performance metrics compared to EfficientNetB3. With an accuracy of 96.3, precision of 96.48%, recall of 96.23%, and F1 score of 96.35%, the Proposed framework exhibits enhanced accuracy in classifying onion pests and diseases. Training and Validation Accuracies and the Confusion Matrix for testing the OnionClassifier are illustrated in Figures 10 and 11, respectively.

**Table 8.** Onion Pest and Disease Detection Performance.

Onion Model	Accuracy	Precision	Recall	F1
EfficientNetB3	92.8	93	92.7	92.5
Proposed Framework	96.3	96.5	96.2	96.4



Figure 10. Training and Validation Accuracy for Onion Classifier.

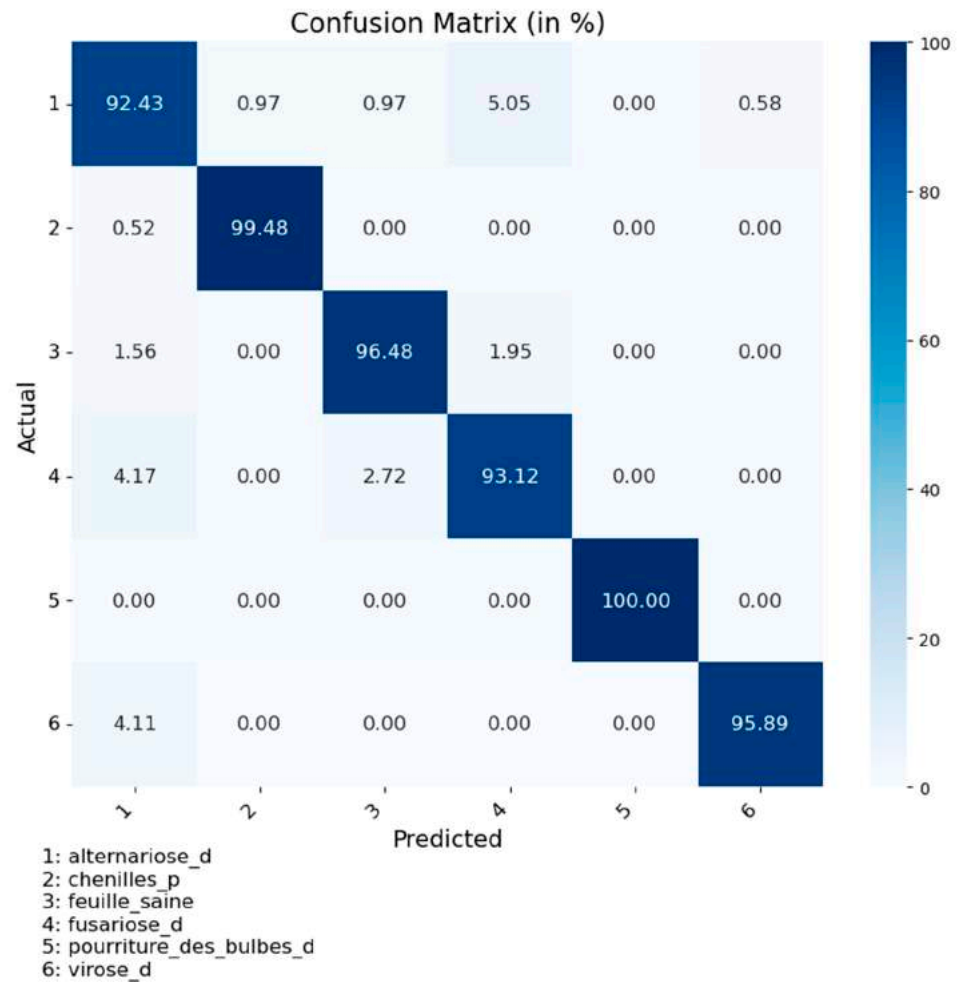
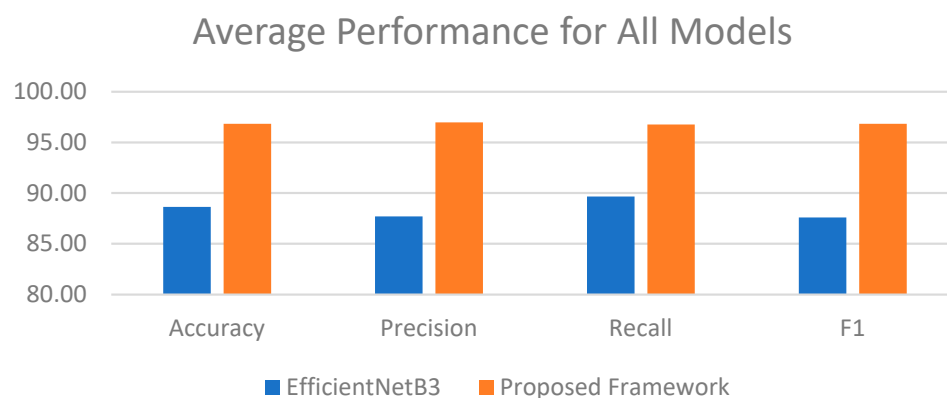


Figure 11. Testing accuracies for each class in the Onion dataset (Average accuracy = 0.96).

The EfficientNetB3 model achieved an accuracy of 92.8%, with precision, recall, and F1 scores of 93%, 92.7%, and 92.5%, respectively. In contrast, the Proposed Framework demonstrated notable improvements, achieving an accuracy of 96.3%, with precision, recall, and F1 scores all exceeding 96%. This highlights its superior performance in accurately identifying onion-related tasks. The Proposed Framework’s precision of 96.48% significantly outperforms EfficientNetB3’s 93%, underscoring its exceptional ability to minimize false positives in identifying onion pests and diseases.



The Proposed Framework achieves a remarkable recall rate of 96.23%, ensuring comprehensive threat detection by identifying most actual onion pest and disease instances, which is crucial for effective monitoring and management. Its high recall rate minimizes the risk of overlooking potential outbreaks. Additionally, with an impressive F1 score of 96.35%, the Proposed Framework maintains a strong balance between precision and recall, surpassing EfficientNetB3's F1 score of 92.5%. These metrics underscore the robustness and reliability of the Proposed Framework, making it a compelling choice for pest and disease management in onion cultivation. Figure 12 presents the overall average performance of all the classifiers developed in this work. The exceptional performance of the Proposed Framework across all evaluated crop types highlights its potential to revolutionize pest and disease management in agriculture, offering a reliable solution for accurate pest and disease identification.



**Figure 12.** Average accuracy, precision, recall, and F1 performance for all models.

#### 4.2. *PlanteSaine*

Figure 13 illustrates *PlanteSaine*'s innovative capability to empower users to designate non-continuous sections within an image to create regions of interest (ROI) for submission. This novel feature significantly augments the model's performance by enabling users to focus on specific areas of interest within an image, thereby enhancing the accuracy and relevance of the classification process. This distinctive functionality gives users greater flexibility and control over the input data and optimizes the model's ability to discern key features and characteristics within the designated regions.

An image to be classified by the app is submitted wholly or partly to the AI model. In Figure 13, a tomato fruit is submitted wholly and partly. *PlanteSaine* classified the wholly submitted image as *Tuta absoluta*, while the segmented one is *Helicoverpa armigera*. The app can consider the variation in both instances good results since the possibility of a fruit being infected with diverse diseases can be seen. The app allows users to pinpoint and prioritize relevant portions of an image. This innovative feature reflects *Plantesaine*'s commitment to user-centric design and its continual pursuit of refining and advancing the capabilities of AI-driven plant classification systems.

Upon rigorous testing and analysis of *PlanteSaine*, the app demonstrated robust performance across various metrics, showcasing its capability to leverage AI technology for enhanced functionality and user experience. Specifically, the app exhibits commendable accuracy in its predictive capabilities, consistently delivering reliable results in real-time scenarios. Users reported a seamless interaction with the app, highlighting its intuitive interface and responsiveness.

Overall, the results underscore the successful integration of AI technology into the Android mobile app, offering users a powerful tool for diverse applications. The app's performance and user-friendly interface position it as a promising solution with vast potential for widespread adoption.

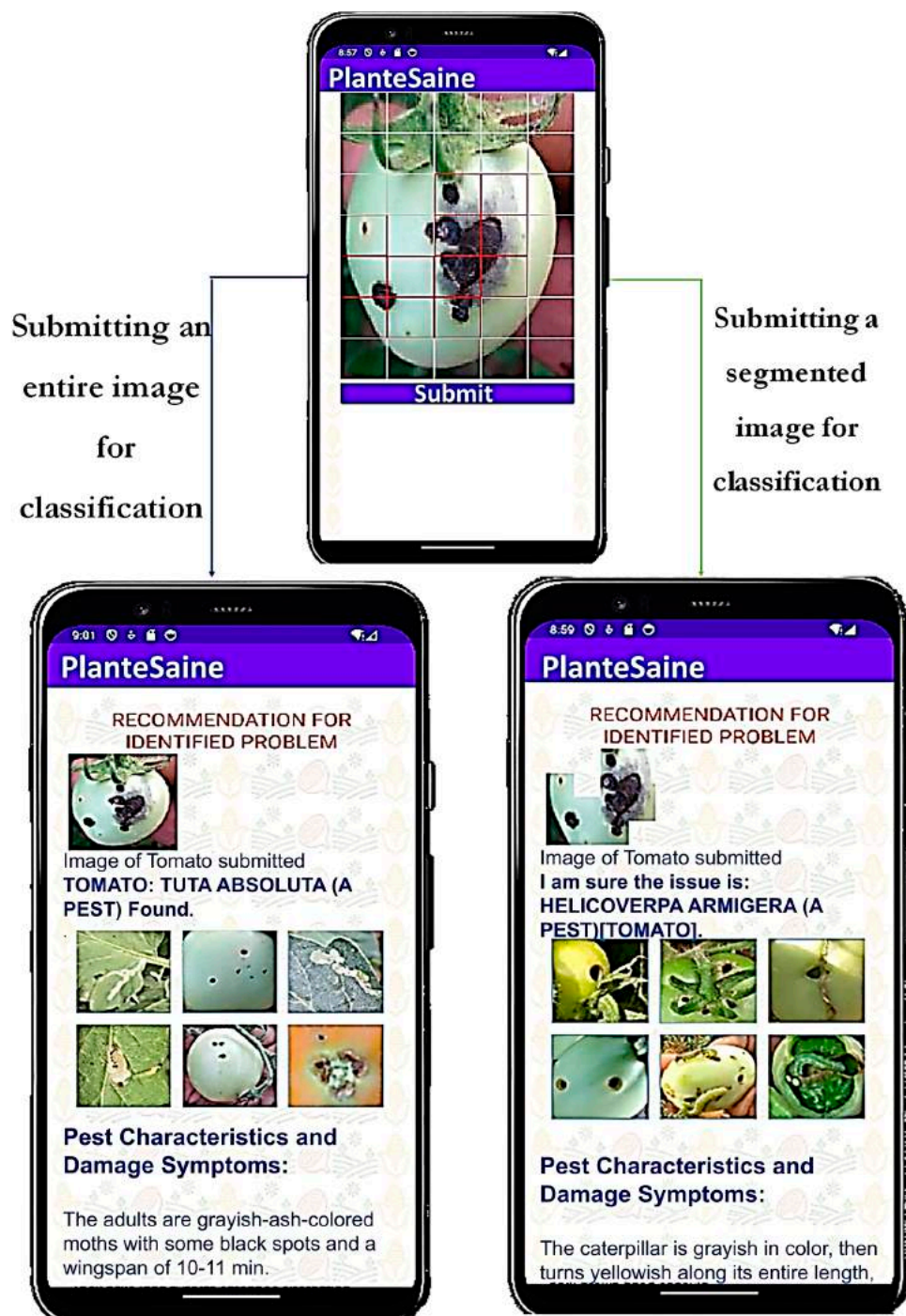


Figure 13. A demonstration of the two modes of submitting images for PlanteSaine to classify.

### 5. Conclusions

In this study, we proposed a comprehensive framework for classifying pests and diseases affecting crops, specifically maize, tomato, and onion. Our approach integrated state-of-the-art deep learning techniques with innovative mobile application development to create a user-friendly and effective solution for farmers, extension officers, and other agricultural stakeholders.

Using the EfficientNetB3 architecture for deep feature extraction, coupled with a customized ANN classifier, we demonstrated significant improvements in accuracy and performance compared to the baseline model. Through rigorous evaluation using accuracy, precision, recall, and F1-score metrics, our proposed framework consistently outperformed

the EfficientNetB3 model across all crop types, showcasing its efficacy in accurately identifying and classifying pests and diseases.

Furthermore, the development of the PlanteSaine mobile application represents a pivotal step toward translating our research findings into practical tools for real-world applications. PlanteSaine offers users a convenient and intuitive platform for capturing, analyzing, and addressing plant health issues in the field by leveraging image segmentation techniques and integrating trained classifiers. The app's innovative features, including the ability to designate regions of interest within images, enhance its usability and effectiveness in plant disease and pest identification. This contributes to the advancement of agricultural technology in Burkina Faso by providing farmers and extension officers with accessible and reliable tools for pest and disease management.

**Author Contributions:** O.A.: Writing—review and editing, Writing—original draft, Methodology, Investigation, Formal analysis, Conceptualization. K.O.H.: Writing—review and editing, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization. B.A.A.D.: Supervision, Methodology. K.O.O.: Funding acquisition, Supervision, Writing—review and editing. S.D.: Methodology—pest identification. O.V.: Conceptualization, Methodology. D.A.-K.: Methodology—data collection. G.D.: Methodology—disease identification. All authors have read and agreed to the published version of the manuscript.

**Funding:** The research was conducted as part of the implementation of the Precision Pest and Disease Management System based on the Multidimensional Big Data (PPeDMaS) project. The project is one of the AGriDI (Accelerating inclusive green growth through agri-based digital innovation in West Africa) projects. Funded through the ACP Innovation Fund of the Organization of African, Caribbean and Pacific States (OACPS) financed by the European Union, AGriDI is implemented by the International Centre of Insect Physiology and Ecology (ICIPE), under its Coordination Unit of the Regional Scholarship and Innovation Fund (RSIF), in collaboration with the University of Abomey Calavi (Benin), Agropolis Foundation (France), and Gearbox Pan African Network (Kenya). The Grant Number is AGriDI/Grants/004.

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** Data used for the training of models can be found at [www.ppedmas.org](http://www.ppedmas.org) (1 January 2024), Download the app at [www.ppedmas.org](http://www.ppedmas.org) or use the Google Play Store.

**Acknowledgments:** This work benefited from the support of other experts and institutions. Notable among them are the extension officers (particularly those from the Centre-Ouest, Centre-Sud, and Plateau Centrale regions) and experts of the General Directorate of Plant Production (DGPV) under the Ministry of Agriculture. Special thanks also go to the staff of WASCAL including Jesse Naab, Momo Bebe, Ivan Bessin, Audrey Codjia, Kisito Gandji, and Melika Vodounhessi who contributed in different ways.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Ferreira, V.; Almazán-Gómez, M.Á.; Nechifor, V.; Ferrari, E. The role of the agricultural sector in Ghanaian development: A multiregional SAM-based analysis. *J. Econ. Struct.* **2022**, *11*, 6. [[CrossRef](#)] [[PubMed](#)]
2. Angon, P.B.; Mondal, S.; Jahan, I.; Datto, M.; Antu, U.B.; Ayshi, F.J.; Islam, M.S. Integrated Pest Management (IPM) in agriculture and its role in maintaining ecological balance and biodiversity. *Adv. Agric.* **2023**, *1*, 5546373. [[CrossRef](#)]
3. Stathas, I.G.; Sakellariadis, A.C.; Papadelli, M.; Kapolos, J.; Papadimitriou, K.; Stathas, G.J. The effects of insect infestation on stored agricultural products and the quality of food. *Foods* **2023**, *12*, 2046. [[CrossRef](#)] [[PubMed](#)]
4. Jiang, T.; Zhou, T. Unraveling the mechanisms of virus-induced symptom development in plants. *Plants* **2023**, *12*, 2830. [[CrossRef](#)] [[PubMed](#)]
5. Ahmad, A.; Saraswat, D.; el Gamal, A. A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools. *Smart Agric. Technol.* **2023**, *3*, 100083. [[CrossRef](#)]
6. Akinyemi, A.O.; Fadele, E.; Ojeleye, A.E. Exploring a mobile application for pest and disease symptomatic diagnosis in food crops in Nigeria: Implications of its use by smallholder farmers in sub-Saharan Africa. *Ife J. Sci.* **2023**, *25*, 1–2. [[CrossRef](#)]
7. Alpysov, A.; Uzakkyzy, N.; Talgatbek, A.; Moldasheva, R.; Bekmagambetova, G.; Yessekeyeva, M.; Kenzhaliev, D.; Yerzhan, A.; Tolstoy, A. Assessment of plant disease detection by deep learning. *East. Eur. J. Enterp. Technol.* **2023**, *121*, 41–48. [[CrossRef](#)]

8. Guan, H.; Fu, C.; Zhang, G.; Li, K.; Wang, P.; Zhu, Z. A lightweight model for efficient identification of plant diseases and pests based on deep learning. *Front. Plant Sci.* **2023**, *14*, 1227011. [[CrossRef](#)] [[PubMed](#)]
9. Guiam, A.C.; Gutierrez, R.D.; Gapasin, C.V.D.; Matalog, R.P.; Ebuenga, M.D. Smarter Pest Identification Technology (SPIDTECH): A Mobile Application for Digital Identification and Remote Monitoring of Insect Pests and Diseases of Major Crops in the Philippines. *Philipp. J. Sci.* **2021**, *150*, 6B. [[CrossRef](#)]
10. Liang, C.; Shah, T. IoT in Agriculture: The Future of Precision Monitoring and Data-Driven Farming. *Eig. Rev. Sci. Technol.* **2023**, *7*, 85–104.
11. Singh, V.; Sharma, N.; Singh, S. A review of imaging techniques for plant disease detection. *Artif. Intell. Agric.* **2020**, *4*, 229–242. [[CrossRef](#)]
12. Abade, A.; Ferreira, P.A.; Vidal, F.B. Plant diseases recognition on images using convolutional neural networks: A systematic review. *Comput. Electron. Agric.* **2021**, *185*, 106125. [[CrossRef](#)]
13. Geetharamani, G.; Pandian, A.J. Identification of plant leaf diseases using a nine-layer deep convolutional neural network. *Comput. Electr. Eng.* **2019**, *76*, 323–338.
14. Hassan, S.M.; Maji, A.K.; Jasiński, M.; Leonowicz, Z.; Jasińska, E. Identification of plant-leaf diseases using cnn and transfer-learning approach. *Electronics* **2021**, *10*, 1388. [[CrossRef](#)]
15. Chen, R.; Qi, H.; Liang, Y.; Yang, M. Identification of plant leaf diseases by deep learning based on channel attention and channel pruning. *Front. Plant Sci.* **2022**, *13*, 1023515. [[CrossRef](#)] [[PubMed](#)]
16. Ferentinos, K.P. Deep learning models for plant disease detection and diagnosis. *Comput. Electr. Eng.* **2018**, *145*, 311–318. [[CrossRef](#)]
17. Agarwal, M.; Gupta, S.K.; Biswas, K.K. Development of Efficient CNN model for Tomato crop disease identification. *J. Sustain. Comput. Inform. Syst.* **2020**, *28*, 100407. [[CrossRef](#)]
18. Hughes, D.; Salathé, M. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv* **2015**, arXiv:1511.08060.
19. Ullah, Z.; Alsubaie, N.; Jamjoom, M.; Alajmani, S.H.; Saleem, F. EffiMob-Net: A Deep Learning-Based Hybrid Model for Detection and Identification of Tomato Diseases Using Leaf Images. *Agriculture* **2023**, *13*, 737. [[CrossRef](#)]
20. Li, Z.; Wang, D.; Zhu, T.; Tao, Y.; Ni, C. Review of deep learning-based methods for non-destructive evaluation of agricultural products. *Biosyst. Eng.* **2024**, *245*, 56–83. [[CrossRef](#)]
21. Astani, M.; Hasheminejad, M.; Vaghefi, M. A diverse ensemble classifier for tomato disease recognition. *Comput. Electron. Agric.* **2022**, *198*, 107054. [[CrossRef](#)]
22. El Akhal, H.; Yahya, A.B.; Moussa, N.; El Alaoui, A.E.B. A novel approach for image-based olive leaf diseases classification using a deep hybrid model. *Ecol. Inform.* **2023**, *77*, 102276. [[CrossRef](#)]
23. Hossain, M.I.; Jahan, S.; Al Asif, M.R.; Samsuddoha, M.; Ahmed, K. Detecting tomato leaf diseases by image processing through deep convolutional neural networks. *Smart Agric. Technol.* **2023**, *5*, 100301. [[CrossRef](#)]
24. Badiger, M.; Mathew, J.A. Tomato plant leaf disease segmentation and multiclass disease detection using hybrid optimization enabled deep learning. *J. Biotechnol.* **2023**, *374*, 101–113. [[CrossRef](#)] [[PubMed](#)]
25. Wayama, R.; Sasaki, Y.; Kagiwada, S.; Iwasaki, N.; Iyatomi, H. Investigation to answer three key questions concerning plant pest identification and development of a practical identification framework. *Comput. Electron. Agric.* **2024**, *222*, 109021. [[CrossRef](#)]
26. Sathiyar, V.; Josephine, M.S.; Jeyabalaraja, V. An automatic classification and early disease detection technique for herbs plant. *Comput. Electr. Eng.* **2022**, *100*, 108026. [[CrossRef](#)]
27. Shewale, M.V.; Daruwala, R. High performance deep learning architecture for early detection and classification of plant leaf disease. *J. Agric. Food Res.* **2023**, *14*, 100675. [[CrossRef](#)]
28. Huang, M.L.; Chuang, T.C.; Liao, Y.C. Application of transfer learning and image augmentation technology for tomato pest identification. *Sustain. Comput. Inform. Syst.* **2022**, *33*, 100646. [[CrossRef](#)]
29. Wani, J.A.; Sharma, S.; Muzamil, M.; Ahmed, S.; Sharma, S.; Singh, S. Machine Learning and Deep Learning Based Computational Techniques in Automatic Agricultural Diseases Detection: Methodologies, Applications, and Challenges. *Arch. Comput. Methods Eng.* **2022**, *29*, 641–677. [[CrossRef](#)]
30. Ahila Priyadharshini, R.; Arivazhagan, S.; Arun, M.; Mirmalini, A. Maize leaf disease classification using deep convolutional neural networks. *Neural Comput. Appl.* **2019**, *31*, 8887–8895. [[CrossRef](#)]
31. Esgario, J.G.M.; de Castro, P.B.C.; Tassis, L.M.; Krohling, R.A. An app to assist farmers in the identification of diseases and pests of coffee leaves using deep learning. *Inf. Process. Agric.* **2022**, *9*, 38–47. [[CrossRef](#)]
32. Plantix—Your Crop Doctor. Available online: <https://play.google.com/store/apps/details?id=com.peat.GartenBank&hl=en> (accessed on 5 January 2024).
33. Janarthan, S.; Thuseethan, S.; Rajasegarar, S.; Yearwood, J. P2OP—Plant Pathology on Palms: A deep learning-based mobile solution for in-field plant disease detection. *Artif. Intell. Agric.* **2022**, *202*, 107371. [[CrossRef](#)]
34. Kaushik, H.; Khanna, A.; Singh, D.; Kaur, M.; Lee, H.-N. TomFusioNet: A tomato crop analysis framework for mobile applications using the multi-objective optimization based late fusion of deep models and background elimination. *J. Appl. Soft Comput.* **2023**, *133*, 109898. [[CrossRef](#)]
35. Nag, A.; Chanda, P.R.; Nandi, S. Mobile app-based tomato disease identification with fine-tuned convolutional neural networks. *Comput. Electr. Eng.* **2023**, *112*, 108995. [[CrossRef](#)]

36. Shin, J.; Chang, Y.K.; Heung, B.; Nguyen-Quang, T.; Price, G.; Al-Mallahi, A. A deep learning approach for RGB image-based powdery mildew disease detection on strawberry leaves. *Comput. Electron. Agric.* **2021**, *183*, 106042. [[CrossRef](#)]
37. Wang, A.R.; Shabrina, N.H. A deep learning-based mobile app system for visual identification of tomato plant disease. *Int. J. Electr. Comput. Eng.* **2023**, *13*, 6992–7004. [[CrossRef](#)]
38. Picon, A.; Alvarez-Gila, A.; Seitz, M.; Ortiz-Barredo, A.; Echazarra, J.; Johannes, A. Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild. *Comput. Electron. Agric.* **2019**, *161*, 280–290. [[CrossRef](#)]
39. Novtahaning, D.; Shah, H.A.; Kang, J.-M. Deep Learning Ensemble-Based Automated and High-Performing Recognition of Coffee Leaf Disease. *Agriculture* **2022**, *12*, 1909. [[CrossRef](#)]
40. Ngugia, L.C.; Abdelwahaba, M.; Abo-Zahhad, M. Tomato leaf segmentation algorithms for mobile phone applications using deep learning. *Comput. Electron. Agric.* **2020**, *178*, 105788. [[CrossRef](#)]
41. Türkoğlu, M.; Hanbay, D. Plant disease and pest detection using deep learning-based features. *Turk. J. Electr. Eng. Comput. Sci.* **2019**, *27*, 1636–1651. [[CrossRef](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.