Towards Precision Agriculture for Sustainable Chili Pepper Production: A Deep Learning Approach to Crop Disease Detection in Benin

Mireille Gloria Founmilayo Odounfa
odounfemi1@yahoo.com

University of Abomey-Calavi

Castro Gbêmêmali Hounmenou
University of Abomey-Calavi

Valère Kolawolé Salako
University of Abomey-Calavi

Antoine Affokpon
University of Abomey-Calavi

Romain L. Glèlè Kakaï
University of Abomey-Calavi

Research Article

Keywords: Deep Learning, Real-Time Detection, Pepper Diseases, Capsicum, Agricultural Sustainability

Posted Date: July 15th, 2025

DOI: https://doi.org/10.21203/rs.3.rs-6994531/v1

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Discover Artificial Intelligence on November 7th, 2025. See the published version at https://doi.org/10.1007/s44163-025-00583-4.

Deep Learning Enables Precision Agriculture for Sustainable Chili Pepper Disease Detection in Benin

Mireille Gloria Founmilayo Odounfa[1], Castro Gbêmêmali Hounmenou[1], Valère Kolawolé Salako[1], Antoine Affokpon[2], and Romain L. Glèlè Kakaï[1]

[1]Laboratoire de Biomathématiques et d'Estimations Forestières (LABEF), Faculty of Agronomic Sciences, University of Abomey-Calavi, 04 BP 1525, Cotonou, Benin.

[2]Laboratoire d'Etude et de Gestion des Organismes Nuisibles des Végétaux (LaGON), Faculty of Agronomic Sciences, University of Abomey-Calavi, 04 BP 1525, Cotonou, Benin.

Abstract

Ensuring food security is a crucial priority for nations worldwide, but plant diseases significantly hinder this objective, through their impacts on agricultural productivity. Chili pepper (*Capsicum spp.*) is a major crop in West Africa including in Benin. However, its production is challenged by diseases such as anthracnose and Tomato Yellow Leaf Curl Virus (TYLCV) which severely impact its yields and farmer livelihoods. While traditional methods for disease detection and management have been commonly used, they are no longer sufficient to combat rising pest infestations and declining agricultural productivity. To tackle this issue, we built a comprehensive dataset of 213 images of anthracnose-affected leaves, 202 images of TYLCV-infected leaves, and 119 images of healthy leaves, collected under diverse environmental conditions in Benin. The study compared the performance of thirteen (13) deep learning models including among others YOLOv8, MobileNetV2, and DenseNet121 for the classification of chili diseases in Benin using transfer learning techniques to improve their accuracy, based on values close to one (1) of evaluation metrics such as Accuracy, Precision, Recall and F1-Score. Results show that YOLOv8 outperforms other models in detecting and localizing leaf diseases in real time. It achieved a mean Average Precision (mAP@0.5) of 0.995 and mAP@0.5-0.95 of 0.941, demonstrating strong generalization. With precision and recall exceeding 99%, YOLOv8 proves highly reliable for agricultural applications. Among CNN models, MobileNetV2 and DenseNet121 achieved 96.25% accuracy, confirming their effectiveness in disease classification. Our findings demonstrate that deep learning models, particularly YOLOv8, hold immense potential for real-time, automated detection of chili pepper diseases. Future research should focus on expanding datasets, integrating climatic variations, and improving disease severity assessment for enhanced agricultural sustainability.

keywords: Deep Learning, Real-Time Detection, Pepper Diseases, Capsicum, Agricultural Sustainability.

1 Introduction

Chili peppers (*Capsicum spp.*) are a crucial staple crop in Africa and particularly in Benin, playing a vital role in culinary traditions, local economies, and food security [20]. As the second most important vegetable crop after tomatoes in Benin, chili peppers are extensively cultivated by smallholder farmers across the country, contributing to the livelihoods of countless rural communities[47, 58]. The annual crop production at the national level was estimated at 176536 tons in 2016 [19]. The pungent, flavorful fruits are not only indispensable ingredients in Beninese cuisine but also generate significant income through both domestic and export markets[47]. Despite the economic and cultural significance of chili peppers in Benin, the cultivation of this crop faces numerous challenges, primarily due to the prevalence of devastating plant diseases. Two of the most troublesome diseases affecting chili peppers in the region are anthracnose, caused by the fungal pathogen *Colletotrichum spp.*, and Tomato Yellow Leaf Curl Virus (TYLCV), transmitted by whitefly vectors (*Bemisia tabaci.*) [47, 20, 23, 69]. These diseases can lead to substantial yield losses, often exceeding 50%, thereby threatening the livelihoods of chili pepper farmers and the food security of local communities[21]. Traditionally, Beninese farmers and agricultural extension workers have relied on visual inspection and manual sampling methods to identify disease symptoms in chili pepper plants[60]. However, this approach is labor-intensive, time-consuming, and heavily dependent on individual expertise, thus, frequently resulting in inaccurate diagnoses. Furthermore, the similar symptomology of various diseases can confound the accurate identification of the causal agents, hindering the implementation of targeted and effective interventions [71, 20]. These challenges pledge for an urgent need to implement an autonomous and automatic biometric system to disease diagnosis to ultimately improve chili production.

Precision agriculture is emerging as a promising solution, leveraging cutting-edge technologies and data-driven methodologies to increase agricultural productivity [14, 10]. With advancements in computer vision and graphics processing units, disease detection has primarily been categorized into two main approaches: traditional machine vision techniques and deep learning-based target detection methods [38]. Several tasks including classification tasks are carried out by different techniques. However, a series of steps must be followed to achieve classification using conventional machine learning techniques, including pre-processing, feature extraction, optimal feature selection, learning, and classification. It is important to note that feature selection plays a significant role in determining the performance of these machine-learning techniques. Biased feature selection can result in incorrect discrimination among classes. On the other hand, deep learning (DL), a subset of machine learning has the advantage of automating the learning of feature sets for various tasks, unlike conventional ML methods. This means that DL can automatically extract relevant features without the need for explicit feature selection [4]. With its many data technologies and high-performance computing capability, deep learning has become an indispensable technique widely exploited in the agricultural industry [59]. These models enable the analysis of huge image datasets[5]. Convolutional Neural Networks (CNN) are specially DL techniques designed to extract information from images, making them dynamic for disease detection. Deep learning-based systems can operate in real-time in the field, providing instant information to farmers. Automation makes it possible to continuously monitor large areas of crops[41]. Unlike visual methods, deep learning reduces reliance on human expertise, making disease detection and estimation more reliable and reproducible [37]. Deep learning models can also be trained to detect various diseases and adapted to different crops, making them very versatile[68]. These models can detect signs of disease long before they are visible to the naked eye. This allows for early intervention and more effective disease management [2]. The emergence of deep learning has fundamentally transformed approaches to detecting and estimating plant diseases in agriculture [2].

Recent studies have explored the use of deep learning techniques, such as convolutional neural networks (CNNs), for the detection and classification of diseases affecting chili peppers and other related vegetable crops[9]. Naresh et al. [45] introduce an enhanced squeeze and excitation densely connected convolutional neural network (SEDCNN) architecture for the early detection of chili leaf diseases. This model achieves a remarkable disease identification accuracy of 97%, which can be further increased to 98.86% through data augmentation. Pratap et al [51] presents a high-precision multiclass classification of chili leaf diseases using a customized EfficientNetB4 deep learning model, which outperformed other popular CNN architectures like ResNet-50, DenseNet-121, MobileNet-V2, and VGG-16 in detecting chili leaf diseases. Hang et al. [25] compares the performance of several widely used CNN architectures AlexNet, VGGNet, GoogLeNet, ResNet, and SENet on a plant leaf disease dataset. It proposes an improved CNN architecture that integrates the Inception module, Squeeze-and-Excitation module, and global average pooling to mitigate issues related to long training times and overfitting. This enhanced architecture achieves the highest classification accuracy of 91.7% on the dataset and demonstrates superior robustness compared to other CNN methods. Despite the promising results of traditional classifiers in previous studies, their effectiveness in real-time field conditions, especially in West African climates, remains underexplored. This study bridges this gap by assessing the feasibility of integrating other object detection models, such as YOLOv8 for precision agriculture in Benin. Naik et al.[44] compares classification and object detection techniques to recognize leaf diseases, highlighting that object detection also provides the location of the disease on the leaf. Using a dataset of 5100 chili leaf images spanning six different classes, the YOLOv5s object detection model was applied and evaluated with performance metrics [44]. The authors propose an enhanced single-stage object detection model that outperformed the YOLOv5s model in terms of performance metrics. Kanaparthi et al [32] used deep learning methods, specifically YOLOv5 and YOLOv7, to detect chilli pests. The researchers created a new dataset of 13,414 chili pest images across three classes (Black Thrips, Redmites, and White Fly) to improve classification accuracy. The study found that the YOLOv5 algorithm outperformed YOLOv7, VGG-16, and VGG-19 in detecting chili pests, achieving a mean average precision of 98.6%. However, the literature on the application of deep learning for the detection and management of diseases specifically affecting chili peppers cultivated in Benin remains limited.

This study aims to address these research gaps by developing and evaluating deep learning-based models for the early and accurate detection of anthracnose and TYLCV in chili pepper crops cultivated in Benin. The key contributions and innovations of this work can be summarized as follows: i) A comprehensive dataset of chili pepper leaf images, captured in complex backgrounds under various lighting conditions and angles, has been collected, thus complementing existing datasets. These images, including confirmed cases of anthracnose and TYLCV as well as healthy leaves, come from different agricultural regions of Benin. ii) The study employs state-of-the-art deep learning models, including VGG16, VGG19, ResNet50, MobileNetV2, DenseNet121, InceptionV3, Xception, EfficientNetB0, EfficientNetB1, EfficientNetB2, EfficientNetB3, NASNetMobile, and Yolov8, leveraging transfer learning to enhance model performance and adaptability. iii) A thorough comparison of various CNN architectures was conducted to determine their effectiveness in detecting and classifying chili pepper diseases. This comparison helps identify the most suitable models for this specific application.

The rest of this paper is structured as follows: Section 2 (Deep Learning Algorithms Explored) introduces the deep learning models compared in this study. Section 3 (Methods) details the processes for collecting and preparing the chili pepper leaf dataset, including image acquisition under various conditions, hardware and software configurations, training parameters, hyperparameter optimization strategies, and validation and testing procedures, along with the evaluation metrics used to measure model performance. Section 4 (Results) presents the study's findings, providing a comprehensive analysis of the performance of different architectures. Section 5 (Discussion) explores the implications of the findings for improving disease monitoring and management in chili pepper crops in Benin, addressing practical challenges faced by farmers. Finally, Section 6 (Conclusion and Future Work) summarizes the study's main contributions and innovations, highlighting their significance for advancing precision agriculture in Benin, and discusses the potential for integrating the proposed deep learning models into real-world agricultural practices to enhance the sustainability and productivity of chili pepper cultivation. It also outlines future research directions, such as developing more sophisticated models and exploring their applicability to other crops and agricultural contexts.

2 Deep learning algorithms explored

Several state-of-the-art Convolutional Neural Networks (CNNs) were employed to process and analyze image data, enabling real-time detection and disease monitoring in a more efficient and scalable manner. These CNN models include VGG16, VGG19, ResNet50, MobileNetV2, DenseNet121, InceptionV3, Xception, EfficientNet (B0, B1, B2, B3), NASNetMobile, and YOLOv8, each selected for their strengths in image classification and object detection tasks. Comparative analyses have shown that YOLOv8 outperforms earlier YOLO versions and other object detection algorithms, including Faster R-CNN, in both accuracy and processing speed [18, 57].

These architectures vary in terms of complexity, computational efficiency, and performance. Transfer learning was applied across models, utilizing pre-trained weights from the ImageNet dataset to enhance their ability to recognize disease patterns in chili pepper leaves. The study employed fine-tuning techniques to tailor each model to the specific characteristics of chili pepper disease detection. Table 1 providing a detailed description of deep learning models, their advantages, disadvantages, as well as the hyperparameters and tested values for each algorithm, offering a comparative overview of the configurations used in this study.

Table 1: Deep Learning Model Descriptions and Configurations

Algorithms	**Description**	**Advantages**	**Disadvantages**	**Hyperparameters and Possible Values**	**Hyperparameter Values Tested**
VGG16	A 16-layer CNN with 13 convolutional layers and 3 fully connected layers, using small 3×3 filters for progressive feature extraction [65, 61].	Strong feature extraction due to stacked 3×3 convolutions: $y^{(l)} = f(W^{(l)} * x^{(l-1)} + b^{(l)})$, where $y^{(l)}$ is the output of layer l; $W^{(l)}$ is the weight matrix; $*$ denotes convolution; $x^{(l-1)}$ is the input from the previous layer; $b^{(l)}$ is the bias; and $f(\cdot)$ is the activation function [65]. Effective for fine-grained classification.	High computational cost, large number of parameters[61].	Optimizer (SGD, Adam, RMSprop, Adagrad, AdamW), Learning Rate ($\eta \in [10^{-6}, 10^{-1}]$), Batch Size (16, 32, 64, 128, 256), Epochs(10, 20, 30, 50, 100, 200, 300, 500, 1000), Loss Function (Softmax Loss, Log Loss, Sparse Categorical Cross-Entropy).	Optimizer: Adam. Learning Rate:0.0001, 0.001, 0.01. Batch Size: 32, Epochs: 50, 100. Loss Function: Softmax Loss
VGG19	A deeper version of VGG16 with 19 layers, improving feature extraction [55]. Uses small 3×3 convolutions with stride and padding techniques.	Enhanced spatial control via stride and padding formula: $O = \left(\frac{W-F+2P}{S}\right) + 1$, where O is the output size; W is the input size; F is the filter size; P is padding; and S is stride [62]. Better feature refinement than VGG16.	Increased memory usage, slower inference[55].	Optimizer (SGD, Adam, RMSprop, Adagrad, AdamW), Learning Rate ($\eta \in [10^{-6}, 10^{-1}]$), Batch Size (16, 32, 64, 128, 256), Epochs(10, 20, 30, 50, 100, 200, 300, 500, 1000), Loss Function (Softmax Loss, Log Loss, Sparse Categorical Cross-Entropy).	Optimizer: Adam. Learning Rate:0.0001, 0.001, 0.01. Batch Size: 32, Epochs: 50, 100. Loss Function: Softmax Loss

Table 1 – *(Continued)*

Algorithms	**Description**	**Advantages**	**Disadvantages**	**Hyperparameters and Possible Values**	**Hyperparameter Values Tested**
ResNet50	A 50-layer deep CNN with residual connections to allow deeper networks to train effectively without vanishing gradients [7].	Residual learning with skip connections mitigates gradient vanishing: $y^{(l)} = f(x^{(l)}) + x^{(l)}$, where $y^{(l)}$ is the output of layer l, allowing direct paths for gradients during backpropagation [7]. Allows very deep architectures.	Requires more training time, can overfit on small datasets[7].	Optimizer (SGD, Adam, RMSprop, Adagrad, AdamW), Learning Rate ($\eta \in [10^{-6}, 10^{-1}]$), Batch Size (16, 32, 64, 128, 256), Epochs(10, 20, 30, 50, 100, 200, 300, 500, 1000), Loss Function (Softmax Loss, Log Loss, Sparse Categorical Cross-Entropy).	Optimizer: Adam. Learning Rate:0.0001, 0.001, 0.01. Batch Size: 32, Epochs: 50, 100. Loss Function: Softmax Loss
MobileNetV2	A lightweight CNN using depthwise separable convolutions to reduce computation, making it suitable for mobile deployment [31].	Computational efficiency through depthwise separable convolutions: $y = (W_D * x) * W_P$, where W_D represents depthwise filters applied to each input channel separately; and W_P represents pointwise filters that combine features across channels [31]. Suitable for edge devices.	Slightly lower accuracy on complex datasets[31].	Optimizer (SGD, Adam, RMSprop, Adagrad, AdamW), Learning Rate ($\eta \in [10^{-6}, 10^{-1}]$), Batch Size (16, 32, 64, 128, 256), Epochs(10, 20, 30, 50, 100, 200, 300, 500, 1000), Loss Function (Categorical Cross-Entropy, Sparse Categorical Cross-Entropy, Binary Cross-Entropy).	Optimizer: Adam. Learning Rate:0.0001, 0.001, 0.01. Batch Size: 32, Epochs: 50, 100. Loss Function: Categorical Cross-Entropy

Table 1 – *(Continued)*

Algorithms	**Description**	**Advantages**	**Disadvantages**	**Hyperparameters and Possible Values**	**Hyperparameter Values Tested**
DenseNet121	Uses dense connections where each layer receives input from all preceding layers, improving gradient flow and reducing parameters [28].	Dense connections enhance feature reuse: $x^{(l)} = H_l([x^{(0)}, x^{(1)}, ..., x^{(l-1)}])$, where $H_l(\cdot)$ denotes a composite function that concatenates inputs from all previous layers to layer l [28]. Reduces overfitting.	High memory consumption, requires careful tuning [28].	Optimizer (SGD, Adam, RMSprop, Adagrad, AdamW), Learning Rate ($\eta \in [10^{-6}, 10^{-1}]$), Batch Size (16, 32, 64, 128, 256), Epochs(10, 20, 30, 50, 100, 200, 300, 500, 1000), Loss Function (Softmax Loss, Log Loss, Sparse Categorical Cross-Entropy).	Optimizer: Adam. Learning Rate:0.0001, 0.001, 0.01. Batch Size: 32, Epochs: 50, 100. Loss Function: Softmax Loss
InceptionV3	Uses inception modules with multiple filter sizes to capture multi-scale features and factorized convolutions to reduce computational complexity [53].	Efficient factorized convolutions optimize performance: $O(n \cdot c \cdot k^2 \cdot h \cdot w)$, where n is the number of filters; c is the number of input channels; k is filter size; and (h, w) are height and width of input feature maps respectively [53]. Excellent for detecting fine details.	The computational complexity of Xception hinders its deployment on resource-constrained edge devices[26].	Optimizer (SGD, Adam, RMSprop, Adagrad, AdamW), Learning Rate ($\eta \in [10^{-6}, 10^{-1}]$), Batch Size (16, 32, 64, 128, 256), Epochs(10, 20, 30, 50, 100, 200, 300, 500, 1000), Loss Function (Softmax Loss, Log Loss, Sparse Categorical Cross-Entropy).	Optimizer: Adam. Learning Rate:0.0001, 0.001, 0.01. Batch Size: 32, Epochs: 50, 100. Loss Function: Softmax Loss
Xception	Improves upon InceptionV3 by replacing inception modules with depthwise separable convolutions [16].	Enhances computational efficiency using depthwise separable convolutions: $y = (W_D * x) * W_P$, similar to MobileNetV2 but optimized further for performance while maintaining high accuracy [16].	Requires fine-tuning and sensitive to hyperparameter choices [53].	Optimizer (SGD, Adam, RMSprop, Adagrad, AdamW), Learning Rate ($\eta \in [10^{-6}, 10^{-1}]$), Batch Size (16, 32, 64, 128, 256), Epochs(10, 20, 30, 50, 100, 200, 300, 500, 1000), Loss Function (Softmax Loss, Log Loss, Sparse Categorical Cross-Entropy).	Optimizer: Adam. Learning Rate:0.0001, 0.001, 0.01. Batch Size: 32, Epochs: 50, 100. Loss Function: Softmax Loss

Table 1 – *(Continued)*

Algorithms	**Description**	**Advantages**	**Disadvantages**	**Hyperparameters and Possible Values**	**Hyperparameter Values Tested**
EfficientNet (B0-B3)	A family of CNNs that scale depth, width and resolution in a balanced way to optimize accuracy with fewer parameters [63].	Compound scaling optimizes model size while maintaining accuracy: $d = \alpha^{\phi}$, $w = \beta^{\phi}$, $r = \gamma^{\phi}$, where α, β, γ are scaling coefficients for depth (d), width (w), and resolution (r), respectively [63].	Requires extensive fine-tuning and large datasets, training EfficientNets can be time-consuming [67].	Optimizer (SGD, Adam, RMSprop, Adagrad, AdamW), Learning Rate ($\eta \in [10^{-6}, 10^{-1}]$), Batch Size (16, 32, 64, 128, 256), Epochs(10, 20, 30, 50, 100, 200, 300, 500, 1000), Loss Function (Softmax Loss, Log Loss, Sparse Categorical Cross-Entropy).	Optimizer: Adam. Learning Rate:0.0001, 0.001, 0.01. Batch Size: 32, Epochs: 50, 100. Loss Function: Softmax Loss
NASNetMobile	A lightweight CNN optimized using neural architecture search (NAS), balancing accuracy and efficiency designed for mobile applications [72].	NAS-based architecture search finds the best model automatically: $S = O_1, O_2, \ldots, O_n$, where S represents a set of operations chosen from a search space during model optimization [72].	High efficiency but requires extensive pretraining and computationally expensive during model search [15].	Optimizer (SGD, Adam, RMSprop, Adagrad, AdamW), Learning Rate ($\eta \in [10^{-6}, 10^{-1}]$), Batch Size (16, 32, 64, 128, 256), Epochs(10, 20, 30, 50, 100, 200, 300, 500, 1000), Loss Function (Softmax Loss, Log Loss, Sparse Categorical Cross-Entropy).	Optimizer: Adam. Learning Rate:0.0001, 0.001, 0.01. Batch Size: 32, Epochs: 50, 100. Loss Function: Softmax Loss

Table 1 – *(Continued)*

Algorithms	**Description**	**Advantages**	**Disadvantages**	**Hyperparameters and Possible Values**	**Hyperparameter Values Tested**
YOLOv8	The latest YOLO model for real-time object detection and classification uses self-attention mechanisms and anchor-free detection [64].	Fully convolutional network enables real-time detection: $y = f(W * x)$, where y is the output prediction; $f(\cdot)$ represents a function mapping inputs through network weights (W) [64]. Can detect multiple disease symptoms at once. Improves localization, especially for overlapping objects. Helps detect small objects or underrepresented classes.	Computationally demanding requires GPU for training,YOLOv8 still faces limitations, particularly in balancing model complexity with detection accuracy for edge computing applications[29].	Optimizer (SGD, Adam, RMSprop, Adagrad, AdamW), Confidence Threshold(0.25,0.5,0.75), Anchor Boxes(Default, Auto-generated), Learning Rate ($\eta \in [10^{-6}, 10^{-1}]$), Batch Size (16, 32, 64, 128, 256), Epochs(10, 20, 30, 50, 100, 200, 300, 500, 1000), Loss Function (CIoU Loss, Focal Loss, Binary Cross-Entropy).	Optimizer: Adam. Learning Rate:0.0001, 0.001, 0.01. Batch Size: 16, Epochs: 50, 100. Loss Function: Focal Loss

3 Methods

3.1 Study site and data collection

The dataset was assembled using a Techno 20 camera (resolution photos: 3840x2160) and an iPhone 15 Pro Max (resolution photos: 24MP and 48MP), capturing a total of 534 images of chili pepper leaves from various localities across Benin, specifically within the Oueme, Atlantique, and Mono regions. The dataset comprises a total of 415 images of sick leaves (including 213 images affected by anthracnose and 202 images showing symptoms of the Tomato Yellow Leaf Curl Virus (TYLCV)) and 119 images of healthy leaves, representing the plant's different states of health (healthy, diseased, and severely affected).

To ensure a comprehensive and representative dataset, images were collected under diverse conditions, including different lighting, angles, and backgrounds. Data collection occurred between December 2023 and June 2024, spanning three distinct periods of the day: morning (7:00-11:00), lunchtime (12:00-14:00), and afternoon (15:00-18:00).

The dataset was developed in collaboration with a phytopathologist, whose expertise ensured accurate identification and classification of plant health states, as well as precise documentation of the symptoms of each disease.

3.2 Data Preprocessing

The ImageDataGenerator and flow_from_dataframe functions from the TensorFlow/Keras library were employed to preprocess and prepare the image data for training. For the majority of models, images are resized to a standard input size of 224x224 pixels, ensuring uniformity across the dataset. However, certain architectures necessitate tailored adjustments: for instance, the NASNetLarge model performs optimally with images resized to 331x331 pixels, while the YOLOv8 model is designed to accommodate larger inputs of 640x640 pixels. To further enhance model performance, pixel values are normalized to a range of 0 to 1, a critical preprocessing step that standardizes the data and accelerates convergence during training. Beyond preprocessing, data augmentation techniques are applied to artificially diversify the dataset. These techniques include random rotations, horizontal and vertical flips, zooms, and brightness adjustments. These augmentations were applied to simulate real-world environmental variations, ensuring robustness against changes in lighting, leaf positioning, and occlusion.

3.3 Model implementaions

The experimental setup leveraged Google Colab's cloud platform with GPU acceleration to efficiently train deep learning models for plant disease detection. Key Python libraries, including TensorFlow and Keras, were utilized for model development, while Scikit-learn supported data processing and evaluation tasks. OpenCV was used for image preprocessing, such as resizing and augmentation, and Matplotlib and Seaborn were employed to generate visualizations of the results.

The methodology integrates image classification using convolutional neural network (CNN) models, following a structured workflow from image preprocessing to performance evaluation, as illustrated in Figure 1. The process begins with resizing the image dataset and applying data augmentation techniques such as rescaling, rotations, shifts, zooming, and shearing to improve model robustness and avoid overfitting. The CNN models, shown in the figure 1, consist of convolution and pooling layers to extract low- and high-level image features, followed by fully connected dense layers for the final classification, which could be used to identify plant leaf diseases. Models such as VGG16, VGG19, ResNet50, MobileNetV2, DenseNet121, InceptionV3, Xception, EfficientNet (B0, B1, B2, B3), NASNetMobile and YOLOv8 are used through transfer learning. Hyperparameter optimization further improves the model accuracy. The data set is divided into 70% for training, 15% for validation, and 15% for testing to ensure reliable results. During training, performance was tracked using metrics such as accuracy and loss, visualized as curves to monitor the progress of the model, with the test data set used for the final evaluation.

Figure1 summarizes the entire process, from data preparation and training of the CNN model to evaluation and testing.

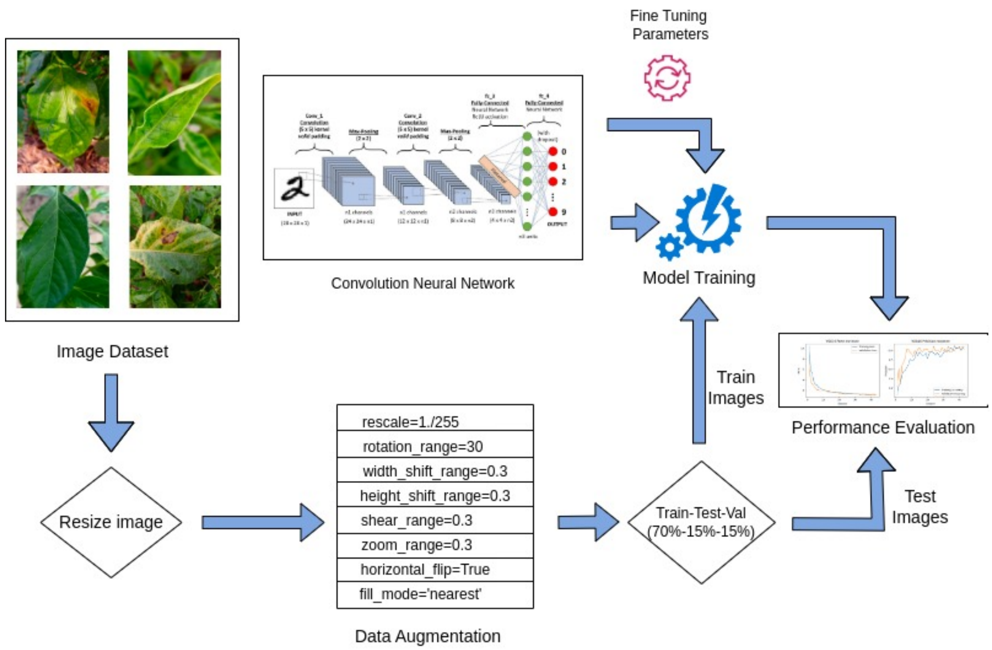


Figure 1: System Architecture

3.4 Training and Hyperparameter Tuning

CNN-based classification models, including VGG16, VGG19, ResNet50, MobileNetV2, DenseNet121, InceptionV3, Xception, EfficientNet (B0, B1, B2, B3), and NASNetMobile, were trained using the Adam optimizer with a learning rate of 0.001 and categorical cross-entropy as the loss function. The training process spanned 100 epochs with a batch size of 32. In contrast, YOLOv8 was trained using the Adam optimizer with a higher learning rate of 0.01 and a reduced batch size of 16 to accommodate its computational complexity. To prevent overfitting and enhance model generalization, early stopping and learning rate reduction techniques were applied, ensuring optimal convergence across all models.

To further optimize the models, hyperparameters such as learning rate, batch size, and number of layers to be fine-tuned were tuned using grid search. This process involved evaluating the models on a validation set comprising 15% of the total dataset, while the remaining 15% was used for testing. 10-fold cross-validation was applied to ensure that the models were robust and generalized well across different data partitions.

3.5 Evaluation Metrics

The performance of the models was evaluated using several key metrics, including precision, recall, F1 score, Matthews correlation coefficient (MCC), confusion matrix, and mean average precision (mAP):

True Positive (TP): Number of samples correctly identified as belonging to class C by the classifier.
True Negative (TN): Number of samples correctly identified as not belonging to class C.
False Positives (FP): Instances incorrectly classified as class C when they do not belong to it.
False Negatives (FN): Number of samples that should have been classified as class C but were categorized into a different class [46],[27].

Accuracy: The proportion of correctly classified instances out of the total cases[46].

$$\text{Accuracy} = \frac{\text{TP + TN}}{\text{TN+TP+FN+FP}}. \tag{1}$$

Precision: The ability of the model to return only relevant results, calculated as the number of true positives divided by the sum of true positives and false positives [39].

$$\text{Precision} = \frac{\text{TP}}{\text{TP + FP}}. \tag{2}$$

Key Properties of Precision

- **Range:** Precision values range from 0 to 1.
- **Precision = 1:** No false positives; all predicted positives are correct.
- **Precision = 0:** All predicted positives are incorrect.

Recall: The ability of the model to identify all relevant instances, calculated as the number of true positives divided by the sum of true positives and false negatives [39].

$$\text{Recall} = \frac{\text{TP}}{\text{TP + FN}}. \tag{3}$$

Key Properties of Recall

- **Range:** Recall values range from 0 to 1.
- **Recall = 1:** No false negatives; all actual positives are detected.
- **Recall = 0:** No actual positives are detected.

F1-Score: The harmonic mean of precision and recall, providing a balance between the two [1],[36].

$$\text{F1-Score} = \frac{2 \times (\text{Precision} \times \text{Recall})}{\text{Precision} + \text{Recall}}. \tag{4}$$

Key Properties of F1 Score

- **Range:** F1 Score values range from 0 to 1.
- **F1 = 1:** Perfect balance between precision and recall.
- **F1 = 0:** Worst possible classification.

Matthews Correlation Coefficient (MCC):

The Matthews Correlation Coefficient (MCC) is a robust metric that evaluates the quality of a classification model, especially in imbalanced datasets. It considers all four elements of the confusion matrix—True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN)—providing a balanced assessment of the model's performance. MCC is computed using the formula:

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}. \tag{5}$$

Key Properties of MCC

- **Range:** MCC values range from -1 to 1.
- **MCC = 1:** Perfect classification.
- **MCC = 0:** No better than random guessing.
- **MCC = -1:** Completely incorrect predictions.

MCC is particularly useful in scenarios with imbalanced classes, as it provides a more informative evaluation than accuracy, precision, or recall alone.

Confusion Matrix: Used to analyze the classification errors and provide a visual representation of the true positives, false positives, true negatives, and false negatives for each class[30].

$$\begin{bmatrix} \mathrm{TP} & \mathrm{FP} \\ \mathrm{FN} & \mathrm{TN} \end{bmatrix}. \tag{6}$$

Mean Average Precision (mAP): This is a common metric in object detection that summarizes the precision-recall curve for each class and averages it across all classes[46].

$$\mathrm{mAP} = \frac{1}{N} \sum_{i=1}^{N} \mathrm{AP}_i$$

,

where:

- N = Number of classes
- AP_i = Average Precision for class i, calculated as the area under the Precision-Recall curve.

Key Properties of Mean Average Precision (mAP)

- **Range:** mAP values range from 0 to 1.
- **mAP = 1:** Perfect detection and classification of all objects.
- **mAP = 0:** No correct detections.

4 Results

4.1 CNN-based classification models

MobileNetV2 and DenseNet121 emerged as the top performers, both achieving 96.25% accuracy. MobileNetV2 demonstrated a precision of 95.69%, recall of 96.25%, F1-score of 96.26%, and an MCC of 92.45%, while DenseNet121 slightly outperformed it with a precision of 96.37%, an F1-score of 96.28%, and the highest MCC of 93.48%. Figures 2c and 2d show a low validation loss, while Figures 2a and 2b highlight high validation accuracy, confirming their robustness in distinguishing diseased and healthy leaves with minimal false positives and negatives. Their superior performance is attributed to MobileNetV2's efficient depthwise separable convolutions and DenseNet121's densely connected layers, which enhance feature reuse and gradient flow.

InceptionV3 also performed well, achieving 90.00% accuracy, with a precision of 90.15%, recall of 90.00%, an F1-score of 90.01%, and an MCC of 80.60%. Its performance benefits from multi-scale feature extraction via inception modules, allowing it to identify fine-grained details in diseased leaves.

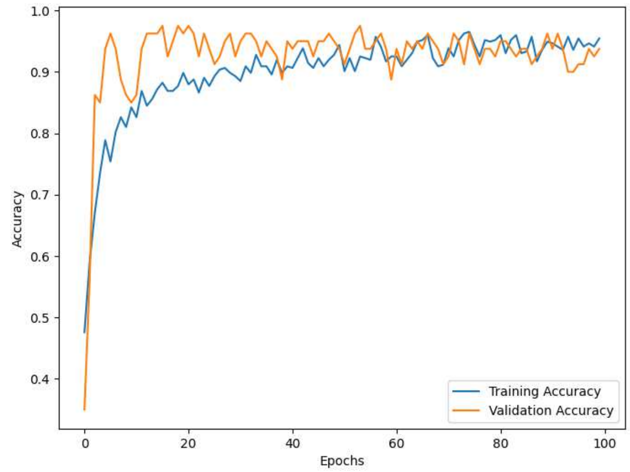
VGG16, NASNetMobile, and VGG19 followed closely, each attaining 88.75% accuracy. VGG16 recorded a precision of 89.76%, recall of 88.75%, an F1-score of 88.50%, and an MCC of 82.72%. NASNetMobile exhibited a precision of 89.90%, recall of 88.75%, an F1-score of 88.73%, and an MCC of 83.20%. VGG19 performed slightly lower, with a precision of 88.31%, recall of 87.50%, an F1-score of 87.40%, and an MCC of 82.70%, showing that these models effectively classify plant diseases but lack the efficiency of MobileNetV2 and DenseNet121.

Xception achieved 83.75% accuracy, with a precision of 84.90%, recall of 83.75%, an F1-score of 83.63%, and an MCC of 78.36%, making it a moderate performer with decent generalization capabilities.

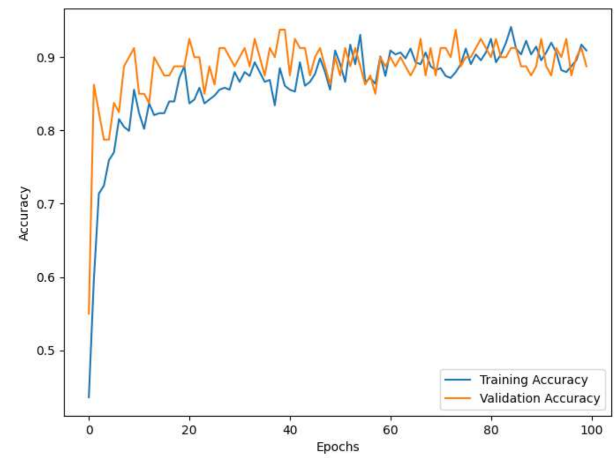
On the other hand, ResNet50 performed poorly, with only 46.25% accuracy, precision of 60.90%, recall of 46.25%, F1-score of 40.55%, and an MCC of 11.06%, indicating high misclassification rates and a lack of generalizability to this dataset.

The EfficientNet models (B0-B3) recorded the worst performance, each obtaining 35.00% accuracy, precision of 77.25%, recall of 35.00%, an F1-score of 18.14%, and an MCC of 0.10%, showing a clear inability to properly classify leaf disease severity. The poor performance suggests that EfficientNet models require significant fine-tuning or alternative scaling strategies to improve their robustness on this dataset.

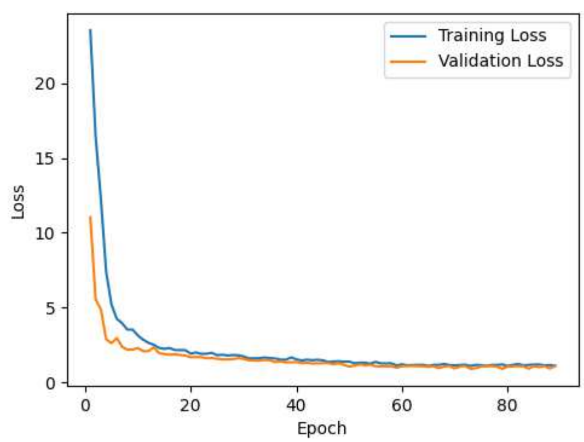
Precision and recall values were highest for MobileNetV2 and DenseNet121, indicating minimal misclassification in the confusion matrices (Figures 2e and 2f). In contrast, ResNet50 and EfficientNet models displayed considerable misclassification, leading to low F1-scores and MCC values. The architectural efficiency of MobileNetV2 and DenseNet121 explains their superior performance, while models like ResNet50 and EfficientNetB0-B3 may require additional adaptation, better to handle the complexity of chili pepper leaf diseases.



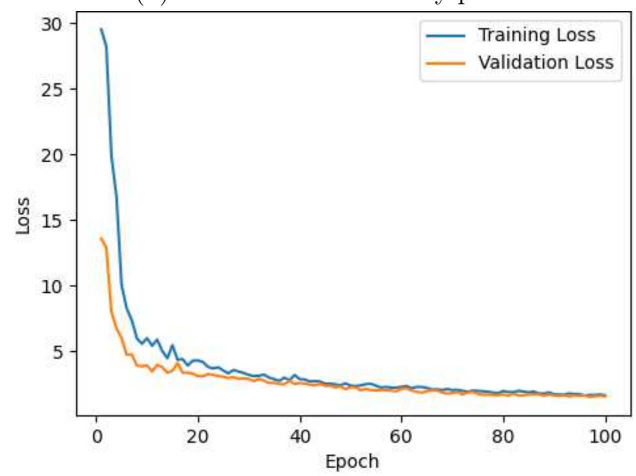
(a) DenseNet121 Accuracy plot



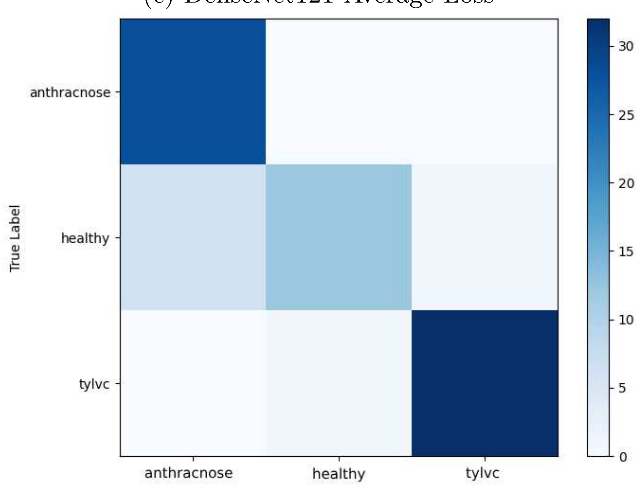
(b) MobileNetV2 Accuracy plot



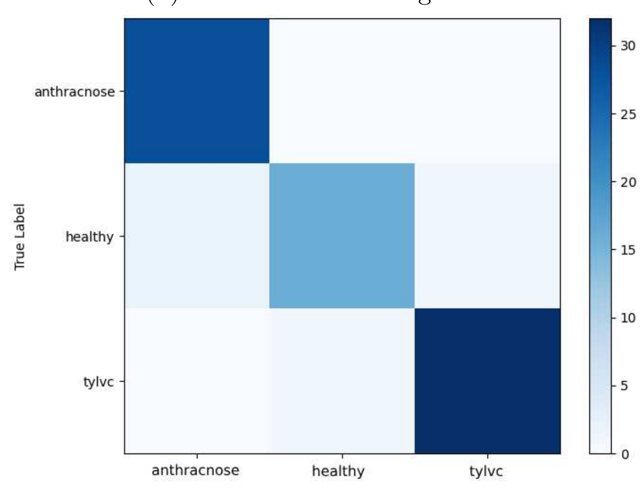
(c) DenseNet121 Average Loss



(d) MobileNetV2 Average Loss



(e) DenseNet121 Confusion Matrix



(f) MobileNetV2 Confusion Matrix

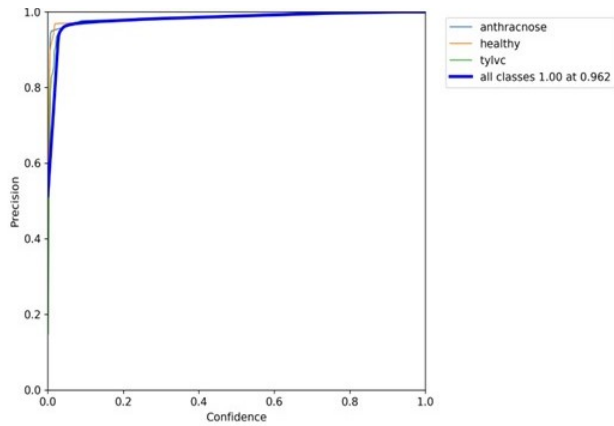
Figure 2: Comparison of DenseNet121 and MobileNetV2 across evaluation metrics: (1) Accuracy plots (top row), (2) Average Loss (middle row), and (3) Confusion Matrices (bottom row).

Table 2: Performance comparison of different models

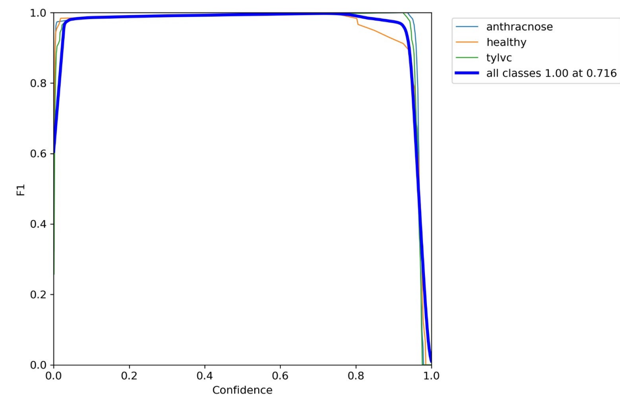
Model	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)	MCC(%)
VGG16	88.75	89.76	88.75	88.5	82.72
VGG19	87.50	88.31	87.5	87.40	82.70
NASNetMobile	88.75	89.90	88.75	88.73	83.20
ResNet50	46.25	60.90	46.25	40.55	11.06
MobileNetV2	96.25	95.69	96.25	96.26	92.45
DenseNet121	96.25	96.37	96.25	96.28	93.48
InceptionV3	90.00	90.15	90.00	90.01	80.60
Xception	83.75	84.90	83.75	83.63	78.36
EfficientNetB0	35.00	77.25	35.00	18.14	00.10
EfficientNetB1	35.00	77.25	35.00	18.14	00.10
EfficientNetB2	35.00	77.25	35.00	18.14	00.10
EfficientNetB3	35.00	77.25	35.00	18.14	00.10

4.2 YOLOv8 architecture

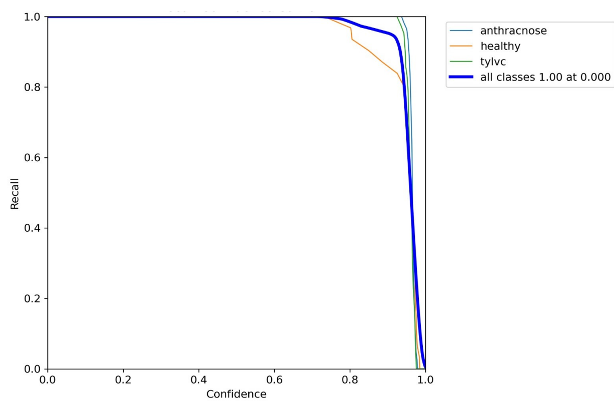
Figure illustrates the results of YOLOv8 in detecting and localizing chili leaf diseases, including anthracnose, Tomato Yellow Leaf Curl Virus (TYLCV), and healthy leaves. Each detected region is marked with a bounding box and a confidence score, showcasing the model's accuracy in real-time disease identification. The recall-confidence curve shows high recall values across all confidence thresholds, with recall remaining at 0.98 for anthracnose, 0.99 for healthy leaves, and 0.97 for TYLCV, indicating consistent detection performance across all categories (figure 3c). The precision-recall curve demonstrates an almost perfect average precision (AP) for all classes, with 0.996 for anthracnose, 0.995 for healthy, and 0.994 for TYLCV, and an overall mAP@0.5 of 0.995, confirming minimal false positives (figure 3d). The training and validation loss curves (figure 4) show a steady decline over the 100 training epochs, with final values of 0.010 for box loss, 0.002 for class loss, and 0.005 for DFL loss, indicating effective learning (figure 4). The F1-Confidence Curve shows an F1-score near 1.00 across all confidence thresholds for anthracnose, healthy, and TYLCV-affected leaves (figure 3b). All classes combined reach an F1-score of 1.00 at a confidence of 0.716, indicating YOLOv8's reliable detection with minimal misclassifications across all categories. Metrics such as precision and recall improved significantly throughout the training, with the final precision reaching 0.996 for anthracnose, 0.995 for healthy, and 0.994 for TYLCV, and the final recall reaching 0.98 for anthracnose, 0.99 for healthy, and 0.97 for TYLCV. The mAP@0.5-0.95 of 0.941 further confirms the model's ability to generalize across various intersection-over-union thresholds (figure 3d). The confusion matrix highlights YOLOv8's exceptional performance in classifying chili pepper leaf diseases. It accurately identified 56 out of 57 anthracnose cases, with only 1 misclassification, and achieved perfect classification for both 31 healthy leaves and 42 TYLCV-affected leaves (figure 3f). These results demonstrate YOLOv8's strong capability in accurately detecting and localizing diseases in chili pepper leaves, providing reliable performance for real-world applications.



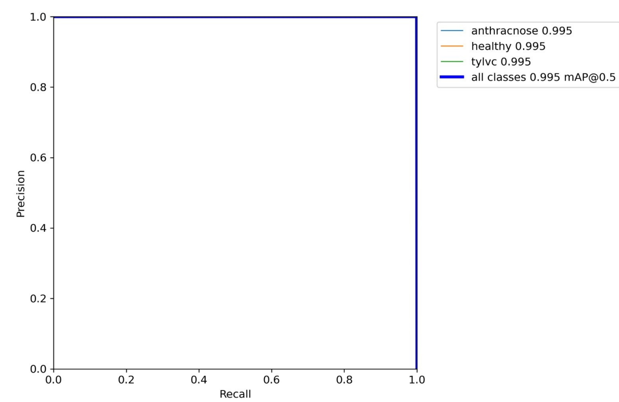
(a) Precision Confidence Curve



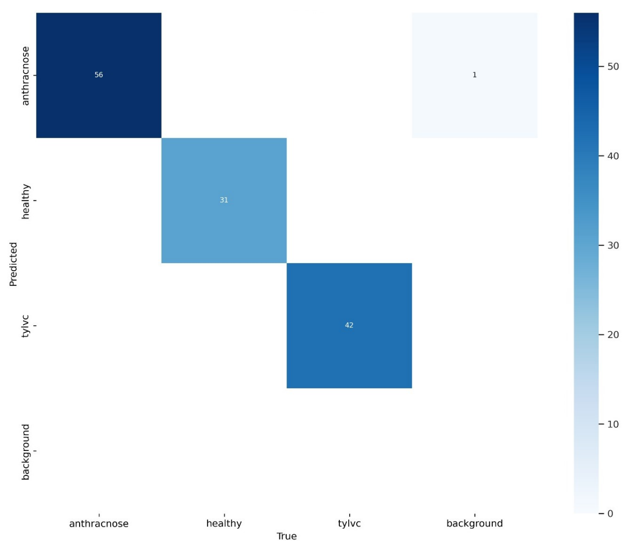
(b) F1 Confidence Curve



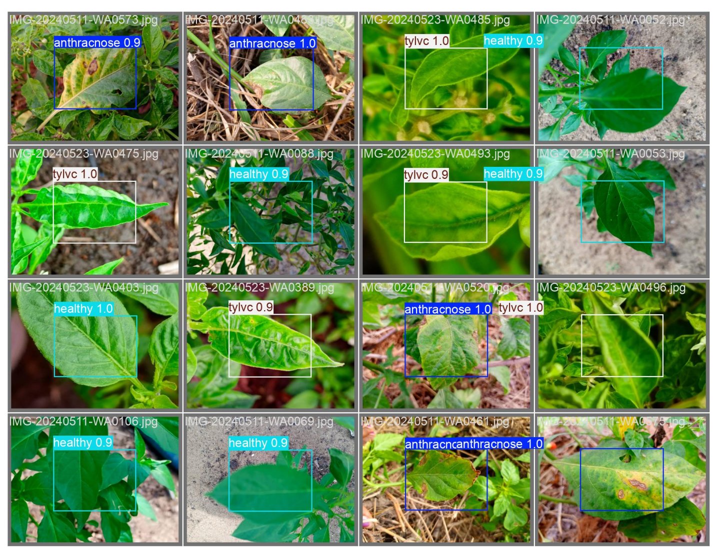
(c) Recall Confidence Curve



(d) Precision-Recall Curve



(e) Confusion Matrix



(f) YOLOv8 Detection of Chili Leaf Diseases

Figure 3: Performance Metrics for the YOLO Model: (a) Precision Confidence Curve, (b) F1 Confidence Curve, (c) Recall Confidence Curve, (d) Precision-Recall Curve, (e) Confusion Matrix, and YOLOv8 Detection of Chili Leaf Diseases.

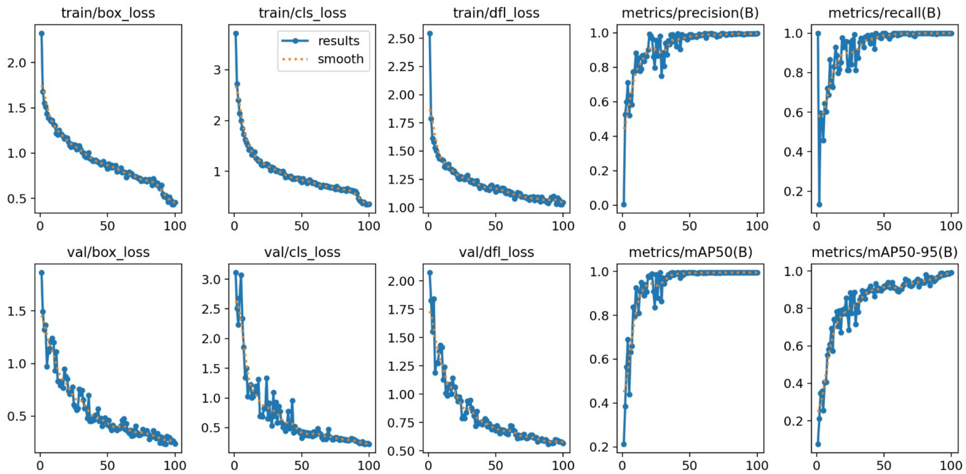


Figure 4: Training and Validation Loss Curves

5 Discussion

This study evaluated both classic deep learning model (GG16, VGG19, ResNet50, MobileNetV2, DenseNet121, InceptionV3, Xception, EfficientNet (B0, B1, B2, B3) and NASNetMobile) and the YOLOv8 architecture for their effectiveness in detecting and localizing chili pepper leaf diseases, specifically anthracnose and Tomato Yellow Leaf Curl Virus (TYLCV). The results show the strengths and limitations of each approach, with significant implications for real-world agricultural applications.

The results obtained align with, and in some cases outperform, previous studies in which similar models achieved high accuracies in plant disease datasets. MobileNetV2 and DenseNet121 demonstrated exceptional classification precision at 96.25%, consistent with the findings of Sobitha et al. [54], who reported that their dual deep learning architecture (DDLA), which combined MobileNet and DenseNet121, achieved classification accuracies of up to 98.71% for the classification of plant species. Similarly, Zheng et al. [70] identified MobileNet as the most efficient architecture for classifying tomato leaf disease, achieving a precision of 91%. Pillai et al. [50] employed DenseNet121 with fine-tuning to classify diseases in bell pepper, potato, and tomato plants, attaining a precision of 97.38%. Moreover, Khultsum et al. [35] highlighted MobileNet's superior performance, achieving 98.00% precision in potato leaf disease classification. InceptionV3, which achieved 90% accuracy, is consistent with findings by Peyal et al. [49] who reported the InceptionV3 model achieved 90% accuracy in detecting tomato leaf diseases. However, VGG16, VGG19, and NASNetMobile demonstrated slightly lower accuracies in this study (88.75%, 87.55%, and 88.75%, respectively), aligning with the findings of Amatullah et al. [43], who utilized a novel dataset of 87,570 images spanning 32 plant species and 74 diseases. Their research highlighted the superior performance of VGG16 and VGG19, which achieved approximately 86% accuracy and an F1-score of 87%, outperforming other models. Similarly, Bhavansh et al. [24] explored automated systems for the detection of crop disease and the identification of weeds using image-based approaches. They found NASNetMobile to be the best performer for crop disease detection in a dataset of three crops with 12 disease classes. Xception, with 83.75% accuracy, mirrors the findings of Barley et al.[56], who indicated that while Xception performed well alongside other models like MobileNet and InceptionV3, it still faced limitations in accurately classifying diseases with overlapping symptoms or those that manifest in early stages. ResNet50 has shown significant performance challenges, achieving only 46.25% accuracy. This poor performance can be attributed to several factors, including the complexity of the dataset and the inherent difficulties in distinguishing between similar disease symptoms. Liang et al. [40] demonstrate that although ResNet50 effectively uses deep learning techniques, it can be difficult to extract characteristics when diseases have overlapping characteristics or subtle variations. On the other hand, YOLOv8 showcased superior results, particularly in localization tasks. The model achieved recall values of 0.98, 0.99, and 0.97 for anthracnose, healthy leaves, and TYLCV, respectively, with precision-recall curve values close to 1. These findings are consistent with research by Usman et al. [3] and Brucal et al. [12] who demonstrated the effectiveness of YOLO in real-time detection of plant disease. The overall mAP@0.5 of 0.995 outperforms previous studies, such as Bebarta et al. [8] where YOLOv5 achieved mAP@0.5 values of 0.641 for the detection of plant disease.

Although the study demonstrates YOLOv8's advantages in detection and localization, certain limitations persist. The relatively small dataset used may lead to overfitting, particularly for traditional deep learning models like ResNet50 and EfficientNet, as emphasized by Vandana et al. [33] and Hongwen et al. [52], who highlighted the critical role of large-scale datasets for robust model training. This issue persists even when using transfer learning techniques [34]. Future research should address this by expanding the dataset and incorporating greater variability in leaf samples. Moreover, YOLOv8's high performance comes at the cost of significant computational demands during training. To overcome this, exploring lightweight YOLO variants or implementing model quantization could provide practical solutions [66, 42]. While this study primarily focused on detection and localization, it did not address disease severity estimation, a key factor for actionable insights in agricultural management, as pointed out by Demba et al. [17]. Future work should integrate regression models into YOLOv8 or similar architectures to estimate severity levels. The high accuracy and recall rates of YOLOv8 make it a promising tool for real-time disease detection in chili crops. By incorporating this model into drones or IoT devices, farmers could efficiently monitor large fields and promptly address issues, as proposed by Pratyay et al. [13] and Soukaina et al. [11]. Furthermore, precise early detection reduces the dependency on chemical pesticides, fostering sustainable agricultural practices, a benefit highlighted by Grella et al. [22]. This approach also minimizes crop losses due to diseases, leading to increased yields and better economic outcomes for farmers, as reported by Smitha et al. [48]. Lastly, YOLOv8's exceptional performance in classification and localization tasks underscores its potential applicability to other crops beyond chili peppers, provided it undergoes proper retraining with new datasets. Chetan et al. [6] also emphasized the versatility of YOLO-based models across diverse agricultural domains. The implementation of YOLOv8 in smart farming tools, such as mobile applications or drones, could enable farmers to detect diseases in real-time, reducing reliance on manual inspection and minimizing pesticide overuse.

6 Conclusion

This study demonstrates the potential of advanced deep learning models, particularly YOLOv8, to detect and localize pepper leaf diseases such as anthracnose and the yellow leaf curl virus (TYLCV). The results highlight YOLOv8's strengths in classification accuracy and precise localization, making it a powerful tool for real-time agricultural applications. CNN-based classification models like MobileNetV2 and DenseNet121 also perform well in classification, achieving high accuracy, precision, and recall. However, YOLOv8 consistently outperforms them by minimizing false positives and providing superior precision and recall metrics in all disease categories. Integrating AI-based solutions into traditional farming practices has immense potential to transform agricultural disease management. By providing farmers with accurate and timely information on plant health, these technologies help mitigate the risks posed by plant diseases, improve crop yields, and support more sustainable farming practices. Future research should expand the application of these models to other crops, integrate climate variables for dynamic disease forecasting, and develop user-friendly tools for direct deployment in farming communities. Additionally, integrating climate variables into these models becomes increasingly important as climate change impacts agriculture. Efforts to improve the interpretability of these AI models are crucial to ensuring that they are both scientifically robust and user-friendly for farmers. By making these technologies more accessible, we promote wider adoption, ultimately creating more resilient and sustainable agricultural systems that are better equipped to handle the challenges posed by plant diseases and climate change.

Author Contributions

M.G.F.O. conceived the study, developed the methodology, implemented the deep learning models, and wrote the main manuscript text. C.G.H. contributed to model evaluation and data analysis. V.K.S. assisted in the design of experiments and manuscript revision. A.A. led the field data collection and disease classification with phytopathological validation. R.L.G.K. supervised the overall research design, provided critical revisions, and ensured scientific rigor. All authors reviewed and approved the final manuscript.

Funding

This work was supported by the German Academic Exchange Service (DAAD), with deepest gratitude extended to the organization. Acknowledgment is also given to the staff members of the Laboratoire de Biomathématiques et d'Estimation Forestière (LABEF/UAC) for their support.

Ethics approval and consent to participate

Not applicable.

Consent to publication

Not applicable.

Competing interest

The authors declare no competing interests.

References

[1] Amreen Abbas, Sweta Jain, Mahesh Gour, and Swetha Vankudothu. Tomato plant disease detection using transfer learning with c-gan synthetic images. *Computers and Electronics in Agriculture*, 187:106279, 2021.

[2] Aanis Ahmad, Dharmendra Saraswat, and Aly El Gamal. A survey on using deep learning techniques for plant disease diagnosis and recommendations for development of appropriate tools. *Smart Agricultural Technology*, 3:100083, 2023.

[3] Usman Ali, Maizatul Akmar Ismail, Riyaz Ahamed Ariyaluran Habeeb, and Syed Roshaan Ali Shah. Performance evaluation of yolo models in plant disease detection. *Journal of Informatics and Web Engineering*, 2024.

[4] Laith Alzubaidi, Jinshuai Bai, Aiman Al-Sabaawi, Jose Santamaría, A S Albahri, Bashar Sami Nayyef Al-dabbagh, Mohammed A Fadhel, Mohamed Manoufali, Jinglan Zhang, and Ali H Al-Timemy. A survey on deep learning tools dealing with data scarcity: definitions, challenges, solutions, tips, and applications. *Journal of Big Data*, 10:46, 2023.

[5] Laith Alzubaidi, Jinglan Zhang, Amjad J Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, José Santamaría, Mohammed A Fadhel, Muthana Al-Amidie, and Laith Farhan. Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions. *Journal of big Data*, 8:1–74, 2021.

[6] Chetan M. Badgujar, Alwin Poulose, and Hao Gan. Agricultural object detection with you look only once (yolo) algorithm: A bibliometric and systematic literature review. *ArXiv*, abs/2401.10379, 2024.

[7] Kola Balavani, Dasari Sriram, Malla Bhavani Shankar, and Devavarapu Sri Charan. An optimized plant disease classification system based on resnet-50 architecture and transfer learning. In *2023 4th International Conference for Emerging Technology (INCET)*, pages 1–5. IEEE, 2023.

[8] Dwiti Krishna Bebarta. Automated plant disease detection using deep learning. *INTERANTIONAL JOURNAL OF SCIENTIFIC RESEARCH IN ENGINEERING AND MANAGEMENT*, 2024.

[9] Yohannes Agegnehu Bezabh, Ayodeji Olalekan Salau, Biniyam Mulugeta Abuhayi, Abdela Ahmed Mussa, and Aleka Melese Ayalew. Cpd-ccnn: classification of pepper disease using a concatenation of convolutional neural network models. *Scientific Reports*, 13:15581, 2023.

[10] Ishita Bhakta, Santanu Phadikar, and Koushik Majumder. State-of-the-art technologies in precision agriculture: a systematic review. *Journal of the Science of Food and Agriculture*, 99:4878–4888, 2019.

[11] Soukaina Bouarourou, El Habib Nfaoui, Abdelhak Boulalaam, and Abderrahim Zannou. A predictive model for abnormal conditions in smart farming using iot sensors. *Procedia Computer Science*, 2023.

[12] Stanley Glenn E. Brucal, Luigi Carlo M. de Jesus, Sergio R. Peruda, Leonardo A. Samaniego, and Einstein D. Yong. Development of tomato leaf disease detection using yolov8 model via roboflow 2.0. *2023 IEEE 12th Global Conference on Consumer Electronics (GCCE)*, pages 692–694, 2023.

[13] Pratyay Chattopadhyay, Homak P. Patel, and Viral Parmar. Internet of things (iot) in smart agriculture. *2022 3rd International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 536–540, 2022.

[14] Fernando Auat Cheein, Daniel Herrera, Javier Gimenez, Ricardo Carelli, Miguel Torres-Torriti, Joan R Rosell-Polo, Alexandre Escolà, and Jaume Arnó. Human-robot interaction in precision agriculture: Sharing the workspace with service units. In *2015 IEEE International conference on industrial technology (ICIT)*, pages 289–295. IEEE, 2015.

[15] Shoufa Chen, Yunpeng Chen, Shuicheng Yan, and Jiashi Feng. Efficient differentiable neural architecture search with meta kernels. *ArXiv*, abs/1912.04749, 2019.

[16] François Chollet. Xception: Deep learning with depthwise separable convolutions. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, 2016.

[17] Demba Faye, Idy Diop, Nalla Mbaye, Doudou Dione, and Marius Mintu Diedhiou. Plant disease severity assessment based on machine learning and deep learning: A survey. *Journal of Computer and Communications*, 2023.

[18] Abdul Ghafar, Caikou Chen, Syed Atif Ali Shah, Zia Ur Rehman, and Gul Rahman. Visualizing plant disease distribution and evaluating model performance for deep learning classification with yolov8. *Pathogens*, 13(12):1032, 2024.

[19] Nicéphore M Glodjinon, Agossou P Noumavo, Brice AM Ohin, Noel S Tovide, Fatiou Toukourou, Lamine Baba-Moussa, Aly Savadogo, and Farid Baba-Moussa. Chilli pepper (capsicum spp.) diversity, production and fungal contamination management in benin. *American Journal of Plant Sciences*, 12(12):1859–1879, 2021.

[20] Nicéphore M Glodjinon, Agossou P Noumavo, Brice A M Ohin, Noel S Tovide, Fatiou Toukourou, Lamine Baba-Moussa, Aly Savadogo, and Farid Baba-Moussa. Chilli pepper (capsicum spp.) diversity, production and fungal contamination management in benin. *American Journal of Plant Sciences*, 12:1859–1879, 2021.

[21] Nicéphore M Glodjinon, Pacôme A Noumavo, Kifouli Adeoti, Kamal Garba, Sonangnon H S Kouhounde, Brice M Ohin, Hama Cisse, Noel S Tovide, Marcellin C Fainou, and Fatiou Toukourou. The technical production, storage and conservation routes of chilli peppers (capsicum spp.) produced in benin and constraints impeding the development of the sector. *J. Anim. Plant Sci*, 42:7279–7295, 2019.

[22] Marco Grella, Paolo Marucco, Jordi Llop, and Fabrizio Gioelli. Special issue on precision technologies and novel farming practices to reduce chemical inputs in agriculture. *Applied Sciences*, 2023.

[23] Yeong Hyeon Gu, Helin Yin, Dong Jin, Jong-Han Park, and Seong Joon Yoo. Image-based hot pepper disease and pest diagnosis using transfer learning and fine-tuning. *Frontiers in plant Science*, 12:724487, 2021.

[24] Bhavansh Gupta, Shubham Chalekar, Saurabh Bomble, Prof. S. R. Vispute, Onkar Gaikar, and Prof. Dr. K. Rajeswari. Convolutional neural networks for detection of crop diseases and weed. *2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA*, pages 1–5, 2022.

[25] Jie Hang, Dexiang Zhang, Peng Chen, Jun Zhang, and Bing Wang. Classification of plant leaf diseases based on improved convolutional neural network. *Sensors*, 19:4161, 2019.

[26] Md. Arid Hasan and Krishno Dey. Depthwise separable convolutions with deep residual convolutions. *ArXiv*, abs/2411.07544, 2024.

[27] Sèton Calmette Ariane Houetohossou, Vinasetan Ratheil Houndji, Castro Gbêmêmali Hounmenou, Rachidatou Sikirou, and Romain Lucas Glele Kakaï. Deep learning methods for biotic and abiotic stresses detection and classification in fruits and vegetables: State of the art and perspectives. *Artificial Intelligence in Agriculture*, 2023.

[28] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[29] Muhammad Hussain. Yolov5, yolov8 and yolov10: The go-to detectors for real-time vision. *ArXiv*, abs/2407.02988, 2024.

[30] Yaser AbdulAali Jasim. High-performance deep learning to detection and tracking tomato plant leaf predict disease and expert systems. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 10(2), 2021.

[31] KC Kamal, Zhendong Yin, Mingyang Wu, and Zhilu Wu. Depthwise separable convolution architectures for plant disease classification. *Computers and electronics in agriculture*, 165:104948, 2019.

[32] Kantha Raju Kanaparthi and S Sudhakar Ilango. Deep learning methods for detecting chilli pests-a novel performance analysis. *Ecological Engineering Environmental Technology (EEET)*, 25, 2024.

[33] Vandana Khobragade, Milan Patnaik, and Srinivasa Rao Sura. Revaluating pretraining in small size training sample regime. *International Journal of Electrical and Electronics Research*, 2022.

[34] Vandana Khobragade, Milan Patnaik, and Srinivasa Rao Sura. Revaluating pretraining in small size training sample regime. *International Journal of Electrical and Electronics Research*, 2022.

[35] Umi Khultsum and Ghofar Taufik. Komparasi kinerja densenet 121 dan mobilenet untuk klasifikasi citra penyakit daun kentang. *JURIKOM (Jurnal Riset Komputer)*, 2023.

[36] Hareem Kibriya, Rimsha Rafique, Wakeel Ahmad, and SM Adnan. Tomato leaf disease detection using convolution neural network. In *2021 International Bhurban Conference on Applied Sciences and Technologies (IBCAST)*, pages 346–351. IEEE, 2021.

[37] Lukas Wiku Kuswidiyanto, Hyun-Ho Noh, and Xiongzhe Han. Plant disease diagnosis using deep learning based on aerial hyperspectral images: A review. *Remote Sensing*, 14:6031, 2022.

[38] Jiawei Li, Yongliang Qiao, Sha Liu, Jiaheng Zhang, Zhenchao Yang, and Meili Wang. An improved yolov5-based vegetable disease detection method. *Computers and Electronics in Agriculture*, 202:107345, 2022.

[39] Rujia Li, Yiting He, Yadong Li, Weibo Qin, Arzlan Abbas, Rongbiao Ji, Shuang Li, Yehui Wu, Xiaohai Sun, and Jianping Yang. Identification of cotton pest and disease based on cfnet-vov-gcsp-lsknet-yolov8s: a new era of precision agriculture. *Frontiers in Plant Science*, 15:1348402, 2024.

[40] Jin Liang and Wenping Jiang. A resnet50-dpa model for tomato leaf disease identification. *Frontiers in Plant Science*, 14:1258658, 2023.

[41] Andreea Roxana Luca, Tudor Florin Ursuleanu, Liliana Gheorghe, Roxana Grigorovici, Stefan Iancu, Maria Hlusneac, and Alexandru Grigorovici. Impact of quality, type and volume of data used by deep learning models in the analysis of medical images. *Informatics in Medicine Unlocked*, 29:100911, 2022.

[42] Shihang Luo, Tao He, Chao Xu, Jixiang Su, Shuqin Wang, Lifeng Zhang, Hu Liang, and Huipeng Li. Research on pedestrian detection based on lightweight yolov8. In *Conference on Computer Graphics, Artificial Intelligence, and Data Processing*, 2024.

[43] Amatullah Fatwimah Humairaa Mahomodally, Geerish Suddul, and Sandhya Armoogum. Machine learning techniques for plant disease detection: an evaluation with a customized dataset. *International Journal of Informatics and Communication Technology (IJ-ICT)*, 2023.

[44] Bhookya Nageswararao Naik, Malmathanraj Ramanathan, and Palanisamy Ponnusamy. Refined single-stage object detection deep-learning technique for chilli leaf disease detection. *Journal of Electronic Imaging*, 32:33039, 2023.

[45] V Naresh, G Yogeswararao, R Malmathanraj, and P Palanisamy. Empirical analysis of squeeze and excitation based densely connected cnn for chilli leaf disease identification. *IEEE Transactions on Artificial Intelligence*, 2024.

[46] Mireille Gloria Founmilayo Odounfa, Charlemagne DSJ Gbemavo, Souand Peace Gloria Tahi, and Romain L Glèlè Kakaï. Deep learning methods for enhanced stress and pest management in market garden crops: A comprehensive analysis. *Smart Agricultural Technology*, page 100521, 2024.

[47] A Orobiyi, Y L Loko, F Sanoussi, A Adjatin, A Gbaguidi, A Dansi, and A Sanni. Horticultural practices and varietal diversity of chili pepper (capsicum annuum l.) in central and northern benin. *Genetic resources and crop evolution*, 64:419–436, 2017.

[48] Smitha S P, Rakshitha D R, B Charan Singh, C H Vamsi Krishna, Rahul M Chavan, and M D Bhavana. Integrated crop protection management. *2024 3rd International Conference for Innovation in Technology (INOCON)*, pages 1–5, 2024.

[49] Hasibul Islam Peyal, Saleh Shahriar, Abida Sultana, Ifat Jahan, and Md. Halim Mondol. Detection of tomato leaf diseases using transfer learning architectures: A comparative analysis. *2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI)*, pages 1–6, 2021.

[50] Rudresh Pillai, N. Sharma, Rupesh Gupta, and Avinash Sharma. Classification of plant diseases using densenet 121 transfer learning model. *2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, pages 1–6, 2023.

[51] V Krishna Pratap and N Suresh Kumar. High-precision multiclass classification of chili leaf disease through customized effecientnetb4 from chili leaf images. *Smart Agricultural Technology*, 5:100295, 2023.

[52] Hongwen Pu and Keyi Yi. A comparative analysis of efficientnet and mobilenet models' performance on limited datasets: An example of american sign language alphabet detection. *Highlights in Science, Engineering and Technology*, 2024.

[53] Zhenping Qiang, Libo He, and Fei Dai. Identification of plant leaf diseases based on inception v3 transfer learning and fine-tuning. In *International Conference on Smart City and Informatization*, 2019.

[54] Anubha Pearline Sundara Sobitha Raj and Sathiesh Kumar Vajravelu. Ddla: dual deep learning architecture for classification of plant species. *IET Image Process.*, 13:2176–2182, 2019.

[55] Ramyea Ravindaran, N Kasthuri, S Preethi, B Adithya, Gunaranjan Sp, K Dharanidharan, and S Aravinth. Performance analysis of a vgg based deep learning model for classification of weeds and crops. In *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pages 1–5. IEEE, 2023.

[56] Masoud Rezaei, Sanjiv Gupta, Dean Diepeveen, Hamid Laga, Michael GK Jones, and Ferdous Sohel. Barley disease recognition using deep neural networks. *European journal of agronomy*, 161:127359, 2024.

[57] Annevena Sandhya, Pottapinjara Babu, U Mohan Srinivas, B Sravanthi, Ksv Phani Kumar, and N Rajeswaran. Enhanced detection of brinjal diseases using yolov8: A high-accuracy real-time model for precision agriculture. In *2024 8th International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 1075–1081. IEEE, 2024.

[58] A Sanni. Chili (capsicum annuum l.) in southern benin: production constraints, varietal diversity, preference criteria and participatory evaluation. 2015.

[59] Iqbal H Sarker. Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions. *SN Computer Science*, 2:420, 2021.

[60] R Sikirou, L Afouda, A Zannou, G Gbèhounou, and F Komlan. Diagnostic des problèmes phytosanitaires des cultures maraichères au sud bénin: cas de la tomate, du piment, de l'oignon et du gombo. *Acte de l'atelier scientifique Sud et Centre tenu du*, 12:102–105, 2001.

[61] Karl Stratos. Vision architectures. 2023.

[62] Karl Stratos. Vision architectures. 2023.

[63] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *ArXiv*, abs/1905.11946, 2019.

[64] Rejin Varghese and Sambath. M. Yolov8: A novel object detection algorithm with enhanced performance and robustness. *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, pages 1–6, 2024.

[65] Chien-Chih Wang, Kent Loong Tan, and Chih-Jen Lin. Newton methods for convolutional neural networks. *arXiv preprint arXiv:1811.06100*, 2018.

[66] Hansen Wang, Dongju Li, and Tsuyoshi Isshiki. A power-efficient end-to-end implementation of yolov8 based on risc-v. *2023 4th International Conference on Computers and Artificial Intelligence Technology (CAIT)*, pages 217–225, 2023.

[67] Arissa Wongpanich, Hieu Pham, James Demmel, Mingxing Tan, Quoc V. Le, Yang You, and Sameer Kumar. Training efficientnets at supercomputer scale: 83% imagenet top-1 accuracy in one hour. *2021 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 947–950, 2020.

[68] Shuli Xing and Hyo Jong Lee. Crop pests and diseases recognition using danet with tldp. *Computers and Electronics in Agriculture*, 199:107144, 2022.

[69] Zhe Yan, Anne-Marie A Wolters, Jesús Navas-Castillo, and Yuling Bai. The global dimension of tomato yellow leaf curl disease: Current status and breeding perspectives. *Microorganisms*, 9:740, 2021.

[70] Ji Zheng and Minjie Du. Study on tomato disease classification based on leaf image recognition based on deep learning technology. *International Journal of Advanced Computer Science and Applications*, 2023.

[71] Herbaud P F Zohoungbogbo, Judicael S O Ganta, Ricardo Oliva, Yuan-Li Chan, Appolinaire Adandonon, Aime H Bokonon-Ganta, Malick N Ba, Enoch G Achigan-Dako, and Derek W Barchenger. Farmers' perception of viral diseases and their management in pepper (capsicum spp.) production in benin. *HortScience*, 59:110–120, 2024.

[72] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8697–8710, 2017.