Journal of Computer Science

Original Research Paper

Agricultural Crop Disease Detection Using Convolutional Neural Networks

Atsu Alagah Komlavi, Harouna Naroua and Chaibou Kadri

Department of Mathematics and Computer Science, Abdou Moumouni University, Niamey, Niger

Article history
Received: 26-03-2024
Revised: 24-06-2024
Accepted: 16-09-2024

Corresponding Author:
Atsu Alagah Komlavi
Department of Mathematics and Computer Science, Abdou Moumouni University,
Niamey, Niger
Email: alagahdejesus@yahoo.fr

Abstract: Mitigating food insecurity in Niger is of paramount importance. According to some researchers, agricultural yield is declining. The permanent annual food shortage was estimated to be between 25 and 50%. Loss of cereal agricultural production is caused by pests and plant diseases. Knowing that ordinary techniques became inefficient. Automatic detection of plant diseases using artificial intelligence becomes the best solution. Several methods applying Convolutional Neural Networks (CNN) have been used in the recent literature. However, these networks are significantly affected by the vanishing gradient problem. Thus, to overcome this challenge, some CNN architectures have been proposed in recent literature. In this study, some properties of these CNN architectures, such as Densely Connected Convolutional Network (DenseNet) and AlexNet, were combined to propose a new efficient architecture called AlexNetDense. Some works prove that not all connections in DenseNet play positive roles for small datasets. And, reducing the connections between layers can improve the efficiency of the network model. Based on AlexNet, the proposed architecture adds some connections not only between consecutive layers. The proposed method was evaluated on PlantVillage. It produced average accuracy, F-measure, and MCC rates of 99.30, 99.12, and 98.90% respectively. Average accuracy, F-measure, and MCC rates of 99.94, 99.94, and 99.88% for this model with Millet Leaf Dataset were obtained. The result showed that the proposed model achieved a higher performance compared to most of the state-of-the-art models, such as LeNet-5 (LeNet), AlexNet, Visual Geometry Group (VGGNet), ResNet, DenseNet, and EfficientNet.

Keywords: Convolutional Neural Networks, Agricultural Crop Disease Detection, PlantVillage Dataset, Millet Leaf Dataset

Introduction

Pearl millet, forms the staple diet of Niger. Families eat these food crops three times a day. Millet is cultivated in a large part of this country. However, because of diseases, there is a decrease in the production of millet between 25-50% every year (Moussa Dit Kalamou *et al*., 2016; Hawey *et al*., 2020; Abdoul Habou *et al*., 2016). and, downy mildew is the most common disease affecting millet crops. There are many traditional solutions to this problem. Therefore, most of these methods are tedious, time-consuming, and least efficient. Hence, it is important to detect millet disease using artificial intelligence algorithms such as Convolution Neural Network (CNN).

The objective of this study is to develop a new efficient and accurate deep convolutional neural network architecture for millet crop disease identification. This model combines the simple structure of AlexNet to which some connections are added, in an efficient manner and takes into account the fact that the extreme connection method of DenseNet makes it high computing costs (Ju *et al*., 2022). Thereby, an optimal reduction of the number of connections leads to higher performance. So, this new model will help farmers to identify diseases in millet crops in order to increase agricultural production in Niger.

Literature Survey

For more than a decade, researchers in the agricultural sector have been interested in precision agriculture using artificial intelligence. Researchers have worked in the field of crop disease diagnosis using machine learning techniques. Several architectures of Convolutional Neural Networks have been evaluated by many researchers who have found satisfactory results in plant disease diagnosis. The LeNet architecture was used in 2017 by Amara *et al*. (2017) for the diagnosis of plant diseases. The researchers collected images from the banana farms. Each image in the dataset was then resized to 60×60 pixels and converted to grayscale. The authors used 80% of the data as the training set and the remainder as the test set. The trained model achieved test accuracies of 98.61 and 94.44% for RGB and grayscale images, respectively (Amara *et al.*, 2017). In 2019, (Arya and Singh, 2019) also used LeNet and other architectures, such as AlexNet, VGG19, VGG16 and ordinary CNN models, on the PlantVillage dataset. The results show that LeNet provides the best accuracy of 99.65%, whereas AlexNet, VGG16, ordinary CNN, and VGG19 achieved accuracies of 98.66, 97.49, 96.70, and 96.67%, respectively.

In 2018, AlexNet was used to propose an intelligent system for the detection of plant diseases (Rangarajan *et al.*, 2018). A tomato dataset of ten classes gathered from PlantVillage was used. A total of 13262 original images of size 256×256 were segmented and increased to 227×227. An accuracy rate of 97.49% was obtained. In 2020, (Matin *et al*., 2020) implemented an intelligent system to detect rice plant diseases using AlexNet architecture. They collected 120 images of healthy and diseased plants using Kaggle. Data augmentation operations were performed and the final size of the dataset was 900 images, of which 70% were used to form the training set and 30% to form the testing set. In the model created using AlexNet, the accuracy was 99.42% (Matin *et al.*, 2020) (Luaibi *et al*., 2021) used AlexNet for lemon plant leaf images, in 2021. They then compared their proposed model with the ResNet model. A dataset of 200 images was used. The latter was divided into three datasets: Training, testing, and validation sets, using 70%, 20%, and 10% split percentages. The images were resized to 227×227×3 and 224×224×3 for AlexNet and ResNet, respectively. The highest accuracy rates for ResNet and AlexNet were 95.83 and 97.92% respectively (Luaibi *et al*., 2021).

In 2018, (Suryawati *et al*., 2018) evaluated the performance of the VGG16 architecture on a tomato dataset of 18160 images from PlantVillage. The architectures highlighted are ordinary CNN, AlexNet, GoogleNet, and VGGNet, with accuracy rates of 84.58, 91.52, 89.68 and 95.24%, respectively (Rangarajan *et al.*, 2018; Kumar and Vani, 2019) selected 80% of the tomato database from PlantVillage as the training set and 20% as the testing set. They used the VGG16 architecture and compared it with LeNet, ResNet, and Xception. Some modifications were applied to VGG16 by adding three convolution layers and a max-pooling layer. The accuracy rates were calculated using a batch size of 30 samples and an epoch size of 30 iterations. The VGG16 model produced the highest accuracy rate of 99.25% on color images, whereas LeNet, ResNet, and Xception architectures produced 96.27, 98.65 and 98.13%, respectively. The VGG16 model provided an accuracy of 99.11% on segmented images, whereas LeNet, ResNet, and Xception yielded accuracy rates of 91.50, 97.55, and 97.11%, respectively. In 2021, (Rinu and Manjula, 2021) proposed a VGG16 architecture to diagnose 38 plant diseases using a dataset from PlantVillage. Their system achieved an accuracy rate of 94.80%.

A study based on ResNet architecture was proposed by Deeba and Amutha (2020). They collected a total of 30085 images of Apples, Corn, and tomatoes from PlantVillage, farms, and Google's website. Preprocessing consisted of removing the noise. The authors trained models using LeNet, AlexNet, VGG16, VGG19, and ResNet34 architectures with an epoch size of 50 and produced accuracy rates of 60.50, 85.50, 91.50, 94.20, and 94.70%, respectively (Deeba and Amutha, 2020). In 2020, (Marzougui *et al*., 2020) proposed a system based on ResNet. It was evaluated using 500 images collected using a Samsung Intelligent LCD camera. They applied common data augmentation techniques, such as rotation (range = 30), width and height shift (range = 0.2), horizontal flip, and fill (mode: Nearest). Two models were built using ResNet and ordinary CNN architectures, with an epoch size of 10. The classification accuracies obtained were 98.96 and 97.20% for ResNet and Ordinary CNN, respectively.

In Too *et al*. (2019) used transfer learning on VGG16, ResNet, InceptionV4, and DenseNet for plant disease detection. Leaf images from PlantVillage were distributed across 38 classes. The images were resized to 224×224 pixels for the VGG16, ResNet, and DenseNet models and to 299×299 pixels for InceptionV4. This dataset was divided into two sets, 80% for training and 20% for testing. DenseNet achieved the best classification rate of 99.75% with an epoch size of 30.

In Khatoon *et al*. developed a system for plant disease detection using DenseNet architecture (Khatoon *et al.*, 2021). They used a dataset of tomatoes from PlantVillage, enriched by images from farms at King Faycal University in Saudi Arabia. The first training and test datasets respectively received 21345 and 2371 images. The second training and test datasets received 17076 and 4269 images, respectively. The VGGNet, ResNet, and DenseNet architectures were used for data standardization and augmentation. In this study, the best success rate was 95.31%, as determined using DenseNet.

In 2022, a plant disease detection approach was proposed by Anak Entuni and Zulcaffle (2022) using the DenseNet201 architecture on a corn dataset from PlantVillage. The images were preprocessed by applying a color space transformation from RGB to YCbCr (a family of color spaces used as a part of the color image pipeline in video and digital photography). Subsequently,

DOI: 10.3844/jcssp.2025.455.468

a data augmentation technique, such as vertical flipping, was used to expand the dataset to 4354 images. After the evaluation, a good accuracy of 95.11% was obtained.

In 2023 (Wang and Shabrina, 2023), EfficientNetB0 architecture was proposed in plant leaf disease detection using images of tomato leaves taken from The PlantVillage dataset. The combination of rotation, brightness, flip, shifting, shear, and zoom is used to apply data augmentation on the original dataset. The results obtained in the test dataset showed that EfficientNetB0 achieved an average accuracy of 91.4%.

Some authors (Rajeena *et al*., 2023) use images from the PlantVillage and PlantDoc dataset to perform a model based on EfficientNetB0 architecture. The used dataset includes 1306 Common Rust images, 574 Gray Leaf Spot images, 1146 Blight images, and 162 Healthy images.

All leaf disease images are resized to 224×224 dimension and are preprocessed using greyscale conversion, Otsu thresholding, and other morphological operations such as smoothing. Then, all preprocessed images are augmented and the features are extracted using the GLCM feature extraction method. The extracted feature trains the EfficientNetB0 for better classification accuracy of 98.85%.

In 2023, a transfer learning-based EfficientNetB0 has been proposed to identify and classify the plant leaf disease for pepper, potato, and tomato plants taken from PlantVillage (Dheeraj and Chand, 2023). A total of 24313 images of three plants having 12 leaf diseases along with 3 classes with healthy leaves have been used by authors. These images are preprocessed and normalized to the same size (224×224) and remove noises. Then, they are trained on the EfficientNetB0 architecture. The testing accuracy of this proposed model is 99.79%.

Atila *et al*. (2021), in a study, used EfficientNet architecture to propose models for plant disease detection. They trained their models using a transfer learning approach, with original and augmented PlantVillage datasets having 55448 and 61486 images, respectively. The results obtained in the test dataset showed that EfficientNetB5 and EfficientNetB4 models achieved the highest performances in original and augmented datasets respectively. The EfficientNetB5 model achieved 99.91% accuracy in the original dataset and 99.93% accuracy in the augmented dataset, while the EfficientNetB4 model achieved 99.84% accuracy in the original dataset and 99.97% accuracy in the augmented dataset.

Tables (1-6) and Fig. (1) summarize the performance comparison of some state-of-the-art deep learning architectures on the PlantVillage dataset.

As seen in Fig. (1), through comparative analysis with LeNet, AlexNet, VGGNet, ResNet, DenseNet, and EfficientNet, it is observed that ResNet has better classification accuracy which proves the utility of the ResNet. In addition, EfficientNet and DenseNet also achieved good accuracies. Some advantages of these architectures will be used to propose a new architecture.

Table 1: Performance of LeNet architecture

Authors/ Year	Database	Average
Arya and Singh (2019)	PlantVillage	99.65
Kumar andVani (2019)	PlantVillage	96.27
Deeba and Amutha (2020)	PlantVillage	60.50
Average accuracy	-	85.47

Table 2: Performance of AlexNet architecture

Authors/ Year	Database	Average
Rangarajan *et al.* (2018)	PlantVillage	97.49
Arya and Singh (2019)	PlantVillage	98.66
Suryawati *et al.* (2018)	PlantVillage/Tomato	91.52
Maeda-Gutiérrez *et al.* (2020)	PlantVillage/Tomato	98.93
Deeba and Amutha (2020)	PlantVillage	85.50
Average accuracy	-	94.42

Table 3: Performance of VGG Net architecture

Authors/ Year	Database	Average
Arya and Singh (2019)	PlantVillage	97.49
Arya and Singh (2019)	PlantVillage	96.67
Suryawati *et al.* (2018)	PlantVillage/Tomato	95.24
Kumar and Vani (2019)	PlantVillage	99.25
Kumar and Vani (2019)	PlantVillage/Segmented	99.11
Rinu and Manjula (2021)	PlantVillage	94.80
Deeba and Amutha (2020)	PlantVillage	91.50
Deeba and Amutha (2020)	PlantVillage	94.20
Average accuracy	-	96.03

Table 4: Performance of ResNet architecture

Authors/ Year	Database	Average
Deeba and Amutha (2020)	PlantVillage	94.70
Kumar and Vani (2019)	PlantVillage	98.65
Kumar and Vani (2019)	PlantVillage	97.55
Maeda-Gutiérrez *et al.* (2020)	PlantVillage/Tomato	99.06
Maeda-Gutiérrez *et al.* (2020)	PlantVillage/Tomato	99.15
Average accuracy	-	97.82

Table 5: Performance of DenseNet architecture

Authors/ Year	Database	Average
Too *et al*. (2019)	PlantVillage	99.75
Khatoon *et al.* (2021)	PlantVillage	95.31
Anak Entuni and Zulcaffle (2022)	PlantVillage/Corn	95.11
Average accuracy	-	96.72

Table 6: Performance of EfficientNet architecture

Authors/ Year	Database	Average
Atila *et al.* (2021)	PlantVillage	99.91
Kumar and Vani (2019)	PlantVillage	99.79
Rajeena *et al.* (2023)	PlantVillage	98.85
Wang and Shabrina (2023)	PlantVillage	91.40
Average accuracy	-	97.48

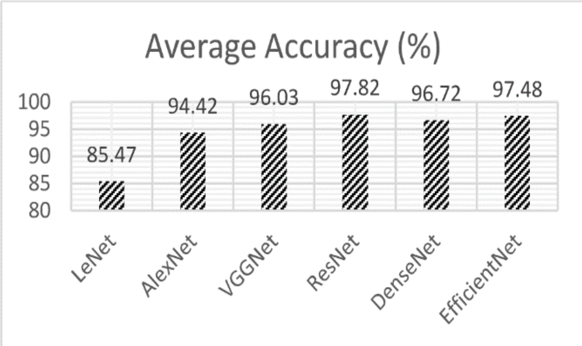

Fig. 1: Performance of some CNN architectures in the recent literature.

Convolutional Neural Networks (CNN)

Convolutional Neural Networks (CNN) are powerful image classification techniques (Jahan *et al*., 2020; Guo *et al*., 2016; Michelucci, 2018). Several CNN architectures have been studied in this study.

LeNet Architecture

LeNet was the first Convolutional Neural Network (CNN) algorithm proposed by Lecun *et al*. (1998) in 1998. It was used mainly to recognize digits and was applied to the recognition of handwritten numbers on bank checks. It is composed of seven layers: Three convolution layers, two pooling layers, and two fully connected layers (Arya and Singh, 2019; (Deeba and Amutha, 2020).

AlexNet Architecture

As proposed by Krizhevsky *et al*., AlexNet architecture is close to LeNet but is deeper, with more filters per layer and stacked convolutional layers directly on top of each other (Krizhevsky *et al*., 2017). It won the ImageNet visual recognition competition (ILSVRC-2012) in 2012 by achieving a top 5 error of 15.30% (Arya and Singh, 2019). The AlexNet architecture consists of one input layer, five convolutional layers, seven activation function (ReLU) layers, three max-pooling layers, two normalization layers, two fully connected layers, one softmax layer, and one output layer (Matin *et al*., 2020)

VGGNet Architecture

VGG (Visual Geometry Group) pioneered the creation of a small filter (3×3) for convolution and deeper network. It is constructed by using a stack of 3 convolutional layers. The max-pooling layers have a filter size of 2×2. VGG16 is proposed by Zisserman and Simonyan at the University of Oxford (Arya and Singh, 2019; Rangarajan *et al*., 2018; Montalbo and Hernandez, 2020). It consists of 13 convolutional layers and three fully connected layers (Luaibi *et al*., 2021; Marzougui *et al*., 2020). VGG16 achieved an accuracy of 92.70% in ImageNet and became one of the top models in the ILSVRC2014 competition.

ResNet Architecture

The vanishing gradient problem makes deep networks difficult to train. Residual Network (ResNets) introduced the concept of skip connection to solve this problem. It was proposed in 2015 by Maeda-Gutiérrez *et al*. (2020); He *et al*. (2016) and won first place in the ILSVRC 2015 competition, with a top-5 error rate of 3.57%. The basic building block of ResNet is the residual block, as shown in Fig. (2). This block is a stack of layers such that input X is directly added to the output of the block: $Y = (X) + X$.

EfficientNet Architecture

EfficientNet was proposed by some Google researchers (Tan and Le, 2019) It has eight models starting from EfficientNet B0 to EfficientNet B7. The difference between each model is the total number of their parameters. The objective of EfficientNet architecture is to provide good accuracy results with appropriate scaling of width, and depth of the deep network and improvement in the resolution of an image. These dimensions can be calculated by the following formulas:

$$\begin{cases} depth: d = \alpha^{\varphi} \\ width: w = \beta^{\varphi} \\ resolution: r = \gamma^{\varphi} \\ \alpha.\beta^2.\gamma^2 \approx 2 \\ \alpha \geq 1, \beta \geq 1, \gamma \geq 1 \end{cases} \quad (1)$$

where, α, β, and γ are the constants. The main building block for EfficientNet is the inverted bottleneck MBConv.

DenseNet Architecture

In 2017, Huang *et al.*, introduced a densely connected convolutional network (DenseNet) to solve the vanishing gradient problem (Huang *et al*., 2017). DenseNet comprises a stack of dense and transition layers. A dense block is a group of layers connected to all previous layers. The feature maps of all preceding layers are used as inputs to the current layer, whose feature maps are used as inputs for subsequent layers. As shown in Fig. (3), a single layer consists of Batch Normalization, a ReLU activation, and a 3×3 convolution (BN–ReLU–Conv). A transition layer is composed of Batch Normalization, a 1×1 convolution layer, and an average pooling layer.

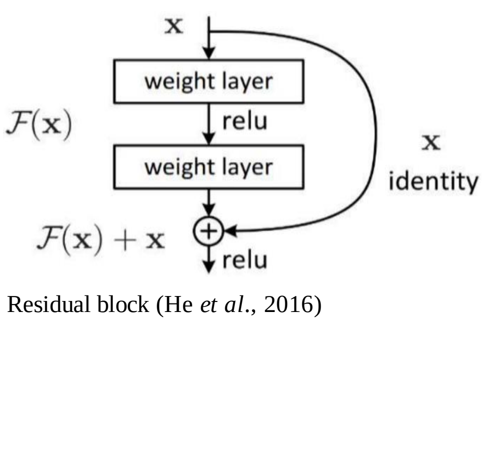

Fig. 2: Residual block (He *et al*., 2016)

DOI: 10.3844/jcssp.2025.455.468

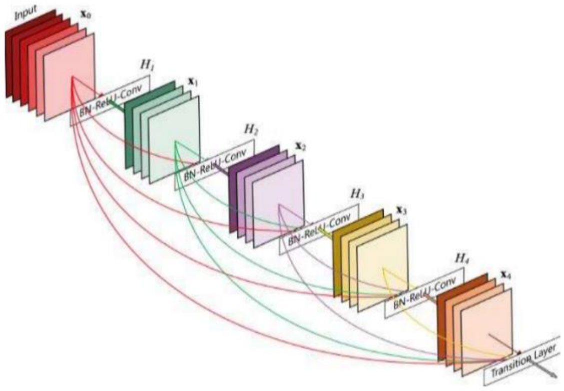

Fig. 3: 5-layer dense block (Huang *et al*., 2017)

The Proposed Architecture: AlexNetDense

It is proved that not all connections in Dense Net play positive roles for small datasets (Ju *et al*., 2022). and, this extreme connection method of DenseNet makes it high computing costs. The new proposed architecture replaces the dense connection between layers with a less dense connection in order to make the model more efficient and lower computing costs. A DenseNet's modified principle is used. It is known that the input of a layer inside the original DenseNet is the concatenation of the feature maps from all the previous layers. Inside AlexNetDense architecture, each input of a layer is a concatenation of the feature maps from the previous two layers.

Several techniques can be used to increase the performance of artificial neural network models. Additionally, in the proposed architecture, some of these techniques were used. There are:

➢ Batch normalization

One of the most common problems in artificial neural networks is overfitting. In machine learning, the known solution to avoid this problem is regularization technique.

One of the regularization tools is batch normalization, which is mostly used in the proposed architecture. Batch Normalization was proposed by Sergey and Christian (Lecun *et al*., 1998) to improve the generalization ability of the model and to allow the model to converge faster. In AlexNetDense architecture, overfitting is prevented by using a Batch Normalization operation on the feature maps of all convolutional layers.

➢ The ReLU Activation Function

An activation function is basically a simple function that transforms inputs into outputs that have a certain range. There are various types of activation functions. However, the ReLU activation function is simple, yet better than its predecessor (Glorot *et al*., 2011). The function returns 0 if it receives any negative input, but returns back any positive value x (Glorot *et al*., 2011). This function f is mathematically expressed as follows:

$$f(x) = ReLU(x) = max(0,x) \tag{2}$$

The main advantage of the ReLU function is that it does not activate all neurons simultaneously. The neurons are only deactivated if the output of the linear transformation is less than 0. In addition, the ReLu function is less computationally expensive than the other functions because it involves simpler mathematical operations. Finally, the ReLU function avoids and rectifies the vanishing gradient problem. In AlexNetDense architecture, the ReLU is used and each neuron activates this function.

AlexNetDense architecture is built with the following convolutional neural network (CNN) layers:

- Convolutional Layer CONV (),
- Batch Normalization (BN),
- Rectified Linear Unit (ReLU).

It contains 12 blocks divided into four parts, as shown in Fig. (4). Each part has 64, 128, 256, or 512 filters for convolution. Depending on the convolutional layer, the kernel size of the convolution can be three, two, or one. Therefore, the input of each block is the concatenation of the last two previous layers. At the end of this proposal, there are flattened, dense, ReLU layers.

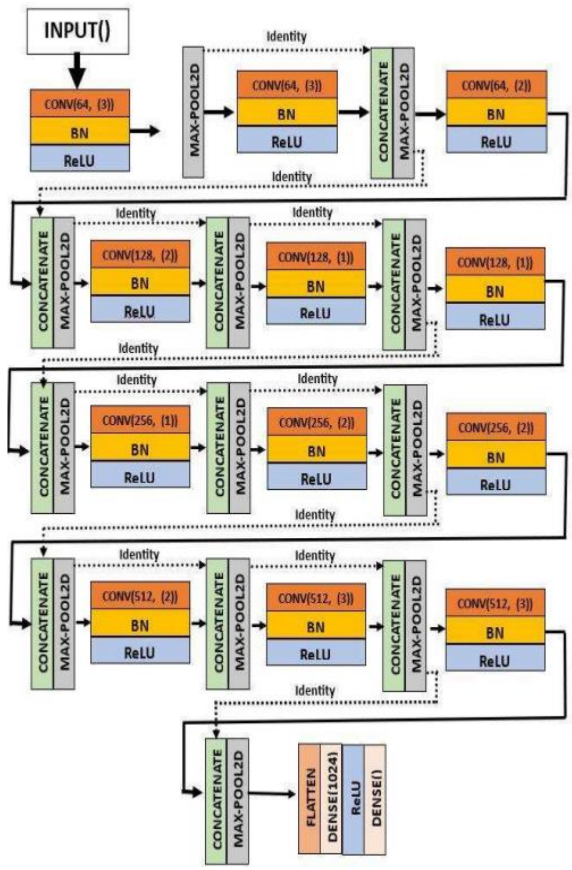

Fig. 4: The proposed AlexNetDense architecture

DOI: 10.3844/jcssp.2025.455.468

Materials and Methods

Image Databases

All CNN Models were applied on two main datasets: PlantVillage and Leaf dataset. Five plants are chosen from PlantVillage (Hughes and Salathé, 2015): Apple dataset of 3172 images and three diseases (Gymnosporangium juniperi-virginianae (276), Venturia insequalis (630), Botryospaeria obtuse (621)), Corn dataset of 3852 images and three diseases (Cercospora zeae-maydis (513), Puccinia sorghi (1192), Exserohilum turcicum (985)), Grape dataset of 4062 images and three diseases (Guignardia bidwellii (1180), Phaeomoniella (1384), Pseudocercospora vitis (1076)), potato dataset of 2152 images and two diseases (Alternaria solani (1000), Phytophthora Infestans (1000)) and tomato dataset of 9270 images and nine diseases as shown in the Fig. (5). PlantVillage is a repository of images of plant introduced in 2015 by David Hughes and Marcel Salathé. It is often used to improve agricultural practices and food security.

The Millet Leaf dataset is collected in three rural communes of Niger named: Sabon Machi, Sherkin Haoussa, and Saé Saboua. This dataset of 7222 images consists of two classes of millet leaves: A class of unhealthy images containing 3312 diseased leaves collected from plants infected by downny mildew and a class of 3910 healthy leaf images. Thus, each image was labeled under the supervision of an agricultural expert. Figure (6) displays sample diseased images of this dataset.

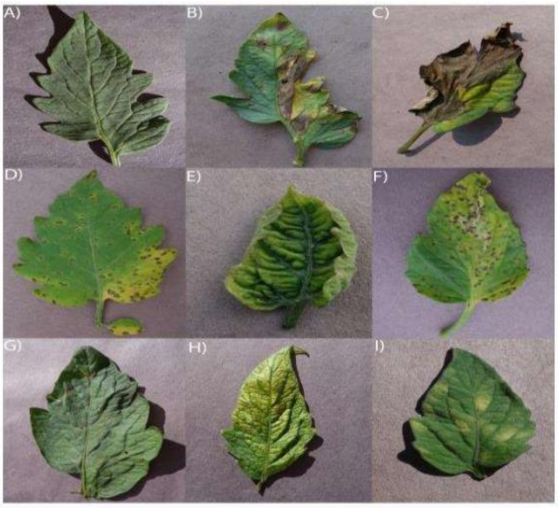

Fig. 5: Examples of different phenotypes of tomato plants. A) Healthy leaf; B) Early Blight; C) Late Blight; D) Septoria Leaf Spot; E) Yellow Leaf Curl Virus; F) Bacterial Spot; G) Target Spot; H) Spider Mite Damage; I) Leaf Mold (Hughes and Salathe, 2015)

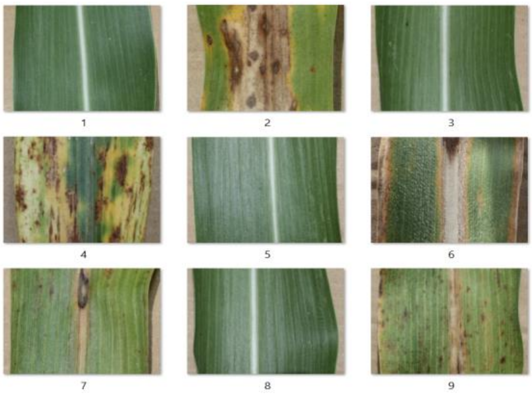

Fig. 6: Examples of different phenotypes of millet plants. Images 1, 3, 5, 8) Healthy leaf. And, images 2, 4, 6, 7, 9) Downy mildew disease.

Data Preprocessing

Before models are trained, image preprocessing is done to improve image consistency in the dataset. One of the most significant operations is the normalization of image size and format. In this study, all leaf images are resized to 100×100 dimensions so that the length and width of the image are the same. The second step of data preprocessing is a sample-wise standardization. It is achieved by subtracting a pixel from its mean value and dividing the result by the standard deviation of the pixel so that the individual features are distributed normally.

Methodology

For experiments, datasets will be divided into two sets: Train and test with the ratio of 80:20. As shown in Fig. (7), to train all the models, three steps were followed: Acquisition of images, image preprocessing, and classification using the CNN algorithms such as LeNet, AlexNet, VGGNet, ResNet, DenseNet, EfficientNet, and AlexNetDense. During the training process, the Adam optimizer was used for all models, except for VGGNet on which a Stochastic Gradient Descent (SGD) optimizer was applied. This study used batch size values of 32 and 64.

All models used a learning rate value of 0.0001, except for The VGGNet and EfficientNet models, on which learning rate values of 0.02 and 0.001 are used. Each model was trained for 1000 epochs, which defined the number of times the model learned the training samples.

Development Environments

The experiments were performed on a Lambda Razer Tensorbook Computer. The specifications of this machine are: Lambda × Razer Tensorbook 2022, graphics (GPU): NVIDIA Corporation, Processor (CPU): 12th Gen Intel® CoreTM i7-12800H×20, System Memory: 32 GiB, Disk Capacity: 2.0 TB, OS Name: Ubuntu 22.04.3 LTS, OS Type: 64-bit. All models used were compiled with GPU support, in Python using Keras backend with TensorFlow.

DOI: 10.3844/jcssp.2025.455.468

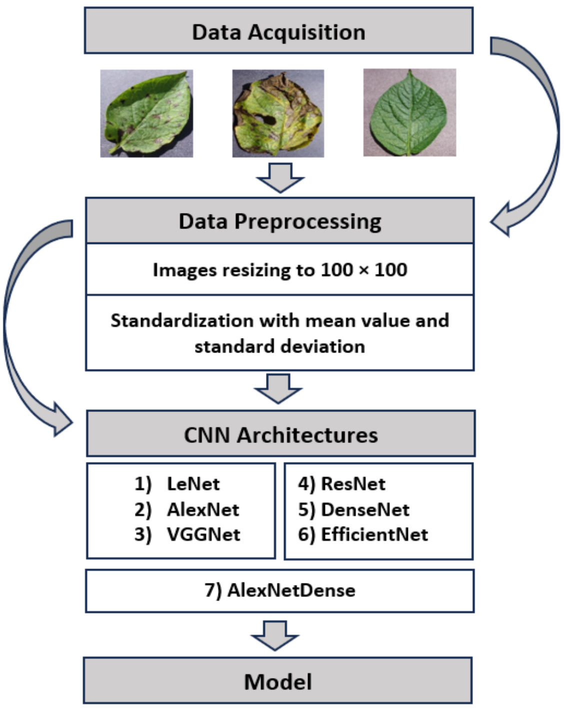

Fig. 7: Methodology diagram

Results and Discussion

Three performance measures are used to evaluate these deep learning models. First, the accuracy rate is calculated by dividing the number of correct predictions by the total number of leaf images tested:

$$Accuracy = \frac{Number\ of\ Correct\ Predivtion}{Total\ Number\ of\ Samples Used\ For\ Prediction} \times 100 \quad (3)$$

Second, the confusion matrix in which several measures are defined based on the confusion matrix.

One is the recall, also called sensitivity, which is the True Positive rate. Another measure often associated with this matrix, the precision, which is the rate of correct predictions among the positive predictions. The simultaneous use of these two indicators gives a good measure called *F-measure*: Harmonic mean of precision and recall. It is defined as:

$$F - measure = \frac{Precision \times Recall}{Precision + Recall} \times 100 \quad (4)$$

Third, the Matthews Correlation Coefficient (MCC) is used in machine learning as a measure of the quality of binary classifications. It takes into account true and false positives and negatives and is generally regarded as a balanced measure that can be used even if the classes are of very different sizes. It is in essence a correlation coefficient between the observed and predicted binary classifications. It is useful when classes are imbalanced. This value is somewhat close to one, which indicates that the model does a decent job of predicting. It can be calculated from the confusion matrix using the formula:

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

In this equation, *TP* is the number of true positives, *TN* is the number of true negatives, *FP* is the number of false positives, and *FN* is the number of false negatives.

In this study, the models are evaluated based on their average Accuracy, F-measure, and *MCC*. The proposed model has been compared with six state-of-the-art deep learning-based models.

From Tables (7-27), the models are evaluated based on PlantVillage dataset.

Table 7: Accuracy rate of LeNet on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	96.37	94.42	95.38	96.05	85.60
2	93.84	94.55	92.76	96.74	86.35
3	94.47	94.16	94.88	96.05	87.38
4	93.05	92.99	93.26	97.44	86.08
5	96.21	91.95	93.88	97.44	84.84
6	92.10	94.68	94.13	95.12	88.13
7	91.94	92.21	87.64	97.44	84.14
8	92.73	91.30	91.76	93.72	87.27
9	94.47	93.64	88.64	96.28	86.46
10	94.00	91.17	92.76	97.91	87.81
Average	93.91	93.10	92.50	96.41	86.40

Table 8: Accuracy rate of AlexNet on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	97.95	96.88	96.63	98.14	95.20
2	97.47	96.88	97.63	97.91	94.61
3	97.31	97.01	96.75	98.60	94.88
4	97.79	96.75	97.25	98.60	94.93
5	97.47	97.53	97.88	98.37	94.82
6	98.10	96.10	97.63	97.91	94.01
7	97.79	95.84	97.25	98.37	95.25
8	97.63	97.14	97.88	98.14	94.82
9	97.95	96.10	97.50	97.91	94.55
10	97.47	96.49	96.38	98.60	95.42
Average	97.69	96.67	97.27	98.18	94.84

Table 9: Accuracy rate of VGGNet on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	96.84	95.97	98.38	98.84	91.37
2	96.37	96.10	98.00	98.84	92.02
3	96.68	96.10	98.25	98.60	90.56
4	95.89	95.06	98.38	99.07	91.15
5	95.89	95.84	98.38	98.84	91.37
6	95.58	95.84	97.38	98.84	90.94
7	95.89	96.23	98.63	99.07	92.13
8	96.21	95.71	98.25	97.44	89.75
9	97.16	96.10	98.00	98.60	90.29
10	97.16	96.23	98.50	98.14	91.26
Average	96.36	95.91	98.21	98.62	91.08

Table 10: Accuracy rate of ResNet on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	97.00	96.75	95.88	98.14	94.07
2	97.31	94.68	96.38	97.91	93.31
3	97.31	96.36	97.63	96.74	93.53
4	97.47	94.68	96.88	96.98	93.15
5	97.16	95.58	96.50	97.21	93.42
6	97.00	97.01	97.63	97.44	93.85
7	97.00	95.45	95.76	97.91	92.77
8	97.00	96.88	96.75	97.67	94.44
9	97.16	96.23	95.76	98.37	93.85
10	97.63	94.94	96.38	99.07	94.39
Average	97.20	95.85	96.55	97.74	93.67

Table 11: Accuracy rate of DenseNet on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	98.74	98.18	99.13	99.07	96.01
2	99.21	97.66	99.50	99.53	96.33
3	98.89	97.79	99.25	99.07	96.39
4	99.53	97.79	99.50	99.30	95.85
5	99.05	97.53	97.75	98.37	96.82
6	99.21	98.57	99.13	99.07	96.82
7	98.26	98.05	99.25	97.91	97.20
8	99.21	97.40	99.25	98.14	97.46
9	99.37	96.88	99.00	98.84	96.49
10	99.53	97.79	99.13	98.37	97.52
Average	99.10	97.76	99.08	98.76	96.68

Table 12: Accuracy rate of EfficientNet on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	97.95	93.51	98.25	98.84	90.02
2	98.42	94.03	98.75	97.67	88.67
3	98.10	93.51	98.25	98.37	89.27
4	98.26	92.73	98.25	96.98	89.10
5	97.47	92.86	99.00	98.37	90.99
6	97.47	93.25	98.75	97.44	90.94
7	97.63	93.90	98.50	98.84	88.89
8	96.05	93.90	98.13	97.67	87.67
9	97.47	93.25	98.50	97.21	89.16
10	97.47	93.12	98.50	97.91	88.67
Average	97.62	93.40	98.48	97.93	89.33

Table 13: Accuracy rate of AlexNetDense on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	99.84	98.18	99.25	100.0	99.03
2	100.0	98.83	99.75	99.07	98.44
3	100.0	98.31	99.50	100.0	98.92
4	99.68	97.53	99.38	99.77	98.60
5	99.84	98.57	99.13	100.0	98.76
6	99.68	97.92	99.88	100.0	98.60
7	99.84	97.79	99.63	99.77	98.60
8	100.0	98.18	99.38	97.67	98.98
9	100.0	98.57	99.50	100.0	98.27
10	100.0	98.05	99.50	100.0	99.44
Average	99.88	98.19	99.49	99.62	98.66

Table 14: F-measure of LeNet on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	96.43	94.37	95.41	96.10	85.64
2	93.95	94.50	92.78	96.78	86.40
3	94.53	94.25	94.91	96.05	87.43
4	93.20	92.93	93.25	97.45	86.07
5	96.24	91.89	93.89	97.45	84.89
6	94.11	94.63	94.15	95.13	88.14
7	91.81	92.12	87.66	97.47	84.14
8	92.86	91.15	91.75	93.75	87.27
9	94.50	93.59	88.61	96.29	86.46
10	94.11	91.02	92.74	97.94	87.80
Average	94.17	93.04	92.51	96.44	86.42

Table 15: F-measure of AlexNet on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	97.97	96.88	96.66	98.16	95.20
2	97.50	96.88	97.68	97.92	94.63
3	97.33	97.01	96.82	98.61	94.89
4	97.80	96.74	97.32	98.61	94.93
5	97.51	97.53	97.93	98.38	94.81
6	98.14	96.11	97.68	97.92	94.03
7	97.81	95.81	97.31	98.39	95.29
8	97.65	97.14	97.90	98.16	94.83
9	97.95	96.11	97.56	97.92	94.56
10	97.52	96.49	96.42	98.61	95.44
Average	97.71	96.67	97.32	98.26	94.86

Table 16: F-measure of VGGNet on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	96.86	96.10	98.41	98.84	91.37
2	96.39	96.14	98.04	98.84	92.01
3	96.72	96.11	98.26	98.61	90.54
4	95.90	95.05	98.39	99.07	91.17
5	95.89	95.86	98.40	98.84	91.37
6	95.60	95.83	97.42	98.61	90.93
7	95.97	96.21	98.65	99.07	92.14
8	96.22	95.70	98.28	97.46	89.75
9	97.18	96.11	98.04	98.62	90.29
10	97.17	96.21	98.53	98.15	91.25
Average	96.39	95.93	98.24	98.61	91.08

Table 17: F-measure of ResNet on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	97.03	96.75	95.89	98.14	94.10
2	97.34	94.65	96.38	97.92	93.32
3	97.33	96.38	97.63	96.76	93.57
4	97.49	94.64	96.89	96.97	93.18
5	97.17	95.56	96.52	97.21	93.41
6	97.05	97.01	97.65	97.47	93.86
7	97.65	95.42	96.43	97.95	92.77
8	97.01	97.00	96.80	97.69	94.44
9	97.19	96.25	95.83	98.39	93.86
10	97.65	94.93	96.43	99.07	94.39
Average	97.29	95.85	96.64	97.80	93.69

Table 18: F-measure of DenseNet on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	98.74	98.17	99.13	99.07	96.03
2	99.21	97.66	99.63	99.53	96.33
3	98.91	97.79	99.26	99.07	96.43
4	99.53	97.81	99.50	99.30	95.90
5	99.06	97.52	97.81	98.38	96.83
6	99.22	98.58	99.13	99.08	96.83
7	98.29	98.04	99.26	97.94	97.21
8	99.22	97.47	99.26	98.16	97.48
9	99.37	96.87	99.03	98.84	96.51
10	99.53	97.85	99.14	98.39	97.54
Average	99.10	97.77	99.11	98.77	96.70

Table 19: F-measure of EfficientNet on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	97.94	93.44	98.26	98.84	90.08
2	98.43	94.08	98.75	97.69	88.71
3	98.10	93.43	98.26	98.38	89.29
4	98.26	92.71	98.27	97.01	89.13
5	97.51	92.76	99.01	98.38	91.09
6	97.47	93.23	98.77	97.46	88.87
7	97.63	93.94	98.51	98.85	88.95
8	96.15	93.84	98.14	97.68	87.79
9	97.49	93.15	98.51	97.27	89.18
10	97.52	93.09	98.51	97.92	88.90
Average	97.65	93.36	98.49	97.94	89.19

Table 20: F-measure of AlexNetDense on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	99.84	98.18	99.26	100.0	99.03
2	100.0	98.83	99.75	100.0	98.44
3	100.0	98.30	99.50	100.0	98.92
4	99.68	98.57	99.38	99.77	98.62
5	99.84	97.53	99.13	100.0	98.77
6	99.69	97.92	99.88	99.77	98.60
7	99.84	97.78	99.63	99.77	98.60
8	100.0	98.18	99.38	97.72	98.98
9	100.0	98.57	99.50	100.0	98.29
10	100.0	98.05	99.50	100.0	98.45
Average	99.88	98.19	99.45	99.61	98.49

Table 21: MCC of LeNet on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	94.41	92.35	93.59	93.00	83.94
2	90.61	92.53	89.95	94.24	84.80
3	91.49	92.04	92.91	92.99	85.93
4	89.37	90.39	90.65	95.46	84.49
5	94.17	88.97	91.51	95.46	83.13
6	87.74	92.70	91.86	91.33	86.76
7	87.24	89.32	82.84	95.46	82.30
8	88.89	88.08	88.55	88.86	85.80
9	91.44	91.31	84.22	93.37	84.90
10	90.87	87.91	89.95	96.29	86.41
Average	90.62	90.56	89.60	93.64	84.84

Table 22: MCC of AlexNet on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	96.85	95.73	95.33	96.71	94.64
2	96.12	95.73	96.73	96.28	93.99
3	95.85	95.91	95.54	97.52	94.29
4	96.60	95.55	96.23	97.52	94.35
5	96.13	96.62	97.09	97.10	94.23
6	97.10	94.47	96.74	96.28	93.33
7	96.60	94.31	96.22	97.12	94.71
8	96.35	96.09	97.06	96.71	94.22
9	96.82	94.67	96.73	96.29	93.93
10	96.15	95.20	94.99	97.52	94.89
Average	96.45	95.42	96.26	96.90	94.25

Table 23: MCC of VGGNet on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	95.12	94.54	97.77	97.93	90.38
2	94.39	94.68	97.25	97.93	91.10
3	94.90	94.67	97.57	97.52	89.47
4	93.64	93.24	97.76	98.35	90.14
5	93.64	94.31	97.76	97.93	90.38
6	93.18	94.31	96.38	97.52	89.90
7	93.70	94.84	98.11	98.35	91.22
8	94.13	94.13	97.59	95.47	88.57
9	95.61	94.67	97.25	97.54	89.17
10	95.60	94.84	97.94	96.69	90.26
Average	94.39	94.42	97.53	97.52	90.05

Table 24: MCC of ResNet on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	95.12	95.56	94.28	96.69	93.39
2	95.84	97.21	94.97	96.29	92.54
3	95.84	95.03	96.71	94.21	92.79
4	96.12	92.71	95.68	96.41	92.37
5	95.59	93.95	95.15	95.03	92.66
6	95.40	95.91	96.71	95.48	93.14
7	95.38	93.78	94.12	96.34	91.94
8	95.36	95.91	95.52	95.88	93.80
9	95.62	94.85	94.14	97.11	93.14
10	96.35	93.06	95.00	98.35	93.74
Average	95.66	94.79	95.22	96.17	92.95

Table 25: MCC of DenseNet on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	98.04	97.51	98.79	98.35	95.55
2	98.78	96.80	99.48	99.17	95.91
3	98.30	96.98	98.97	98.35	95.98
4	99.27	96.98	99.31	98.76	95.38
5	98.53	96.62	96.92	97.12	96.45
6	98.78	98.05	98.79	98.36	96.45
7	97.33	97.33	98.97	96.32	96.87
8	98.78	96.48	98.96	96.71	97.17
9	99.02	95.74	98.62	97.93	96.09
10	99.27	97.01	98.80	97.13	97.24
Average	98.61	96.95	98.76	97.82	96.30

Table 26: MCC of EfficientNet on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	97.55	91.11	97.58	97.93	88.91
2	97.55	91.85	98.27	95.85	87.39
3	97.05	91.14	97.58	97.10	88.05
4	97.31	90.04	97.58	94.64	87.87
5	96.10	90.24	98.62	97.11	89.96
6	96.07	90.83	98.28	95.47	87.62
7	96.33	91.66	97.92	97.94	87.63
8	93.92	91.64	97.40	95.88	86.37
9	96.10	90.76	97.93	95.10	87.92
10	96.13	90.58	97.93	96.30	87.41
Average	96.41	90.98	97.90	96.33	87.91

Table 27: MCC of AlexNetDense on 5 datasets

Test no	Apple	Corn	Grape	Potato	Tomato
1	99.76	97.51	98.96	100.0	98.92
2	100.0	98.48	99.65	98.35	98.25
3	100.0	97.69	99.31	100.0	98.80
4	99.51	98.04	99.14	99.59	98.44
5	99.76	96.62	98.79	100.0	98.62
6	99.51	97.16	99.83	99.59	98.92
7	99.76	96.98	99.48	99.59	98.44
8	100.0	97.51	99.14	95.94	98.86
9	100.0	98.04	99.31	100.0	98.08
10	100.0	97.33	99.31	100.0	98.26
Average	99.83	97.53	99.29	99.30	98.55

Table 28: Accuracy, F-measure, and MCC rates of CNN models on PlantVillage dataset

CNN	Accuracy	F-measure	MCC	Average
LeNet	92.46	92.51	89.85	91.60
AlexNet	96.93	96.96	95.85	96.58
VGGNet	96.03	96.05	94.78	95.62
ResNet	96.20	96.25	94.95	95.80
DenseNet	98.28	98.29	97.68	98.08
EfficientNet	95.35	95.32	93.90	94.85
AlexNetDense	99.30	99.12	98.90	99.10

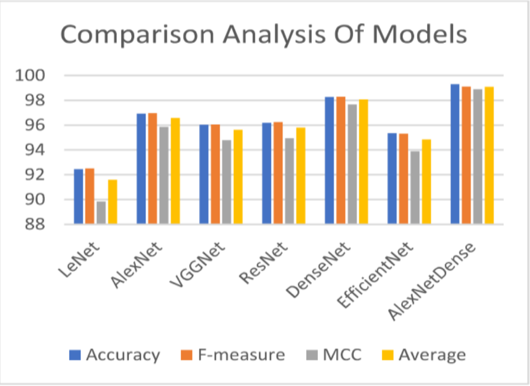

Fig. 8: Comparison analysis of the proposed model under Accuracy, F-measure, and MCC on PlantVillge dataset

Table (28) and Fig. (8) summarize the performance comparison of different models with AlexNetDense on the PlantVillage dataset. As seen from the results, the AlexNetDense model has a testing accuracy of 99.30%, a testing F-measure of 99.12%, and a testing MCC of 98.90%. From calculation, the determined average value of these three performance measures is 99.10%. So, this result shows that the proposed architecture obtains higher performance than the remaining models: LeNet (91.60%), AlexNet (96.58%), VGGNet (95.62%), ResNet (95.80%), DenseNet (98.08%), EfficientNet (94.85%). In this investigation, images of these plants have been taken from the well-known plant disease dataset named PlantVillage. It has been observed that most studies in the literature use this dataset. Also, it was proven to provide accurate, robust, and successful results. In the next part of the study, the effectiveness of the proposed model and the six models (LeNet, AlexNet, VGGNet, ResNet, DenseNet, EfficientNet) are evaluated on the Milled Leaf dataset and results are reported in Tables (29-35).

Table 29: Accuracy, F-measure, and MCC rates of LeNet on Millet dataset

Test N°	Accuracy (%)	F-measure (%)	MCC (%)
1	99.24	99.24	98.47
2	99.65	99.66	99.31
3	99.58	99.59	99.17
4	99.65	99.66	99.31
5	99.45	99.45	98.89
6	99.65	99.66	99.31
7	99.45	99.45	98.89
8	99.72	99.72	99.44
9	99.65	99.65	99.30
10	99.52	99.52	99.03
Average	99.55	99.56	99.11

Table 30: Accuracy, F-measure, and MCC rates of AlexNet on Millet dataset

Test N°	Accuracy (%)	F-measure (%)	MCC (%)
1	99.86	99.86	99.72
2	99.86	99.86	99.72
3	99.65	99.66	99.31
4	99.86	99.86	99.72
5	99.79	99.79	99.58
6	99.79	99.79	99.58
7	99.72	99.72	99.44
8	99.79	99.79	99.58
9	99.86	99.86	99.72
10	99.72	99.72	99.44
Average	99.79	99.79	99.58

Table 31: Accuracy, F-measure, and MCC rates of VGGNet on Millet dataset

Test Nº	Accuracy (%)	F-measure (%)	MCC (%)
1	99.65	99.66	99.31
2	99.65	99.66	99.31
3	99.79	99.79	99.58
4	99.58	99.59	99.17
5	99.86	99.86	99.72
6	99.86	99.86	99.72
7	99.79	99.79	99.58
8	99.72	99.72	99.44
9	99.65	99.65	99.30
10	99.79	99.79	99.58
Average	99.73	99.73	99.47

Table 32: Accuracy, F-measure, and MCC rates of ResNet on Millet dataset

Test Nº	Accuracy (%)	F-measure (%)	MCC (%)
1	99.79	99.79	99.58
2	99.86	99.86	99.72
3	99.72	99.72	99.44
4	99.86	99.86	99.72
5	99.79	99.79	99.58
6	99.79	99.79	99.58
7	99.65	99.65	99.30
8	99.86	99.86	99.72
9	99.65	99.66	99.31
10	99.93	99.93	99.86
Average	99.79	99.79	99.58

Table 33: Accuracy, F-measure, and MCC rates of DenseNet on Millet dataset

Test Nº	Accuracy (%)	F-measure (%)	MCC (%)
1	99.86	99.86	99.72
2	99.93	99.93	99.86
3	99.79	99.79	99.58
4	99.72	99.72	99.44
5	99.93	99.93	99.86
6	99.93	99.93	99.86
7	99.86	99.86	99.72
8	99.86	99.86	99.72
9	99.86	99.86	99.72
10	99.79	99.79	99.58
Average	99.85	99.85	99.70

Table 34: Accuracy, F-measure, and MCC rates of EfficientNet on Millet dataset

Test Nº	Accuracy (%)	F-measure (%)	MCC (%)
1	99.52	99.52	99.58
2	99.79	99.79	99.58
3	99.72	99.72	99.44
4	99.72	99.72	99.44
5	99.65	99.65	99.30
6	99.79	99.79	99.58
7	99.72	99.72	99.44
8	99.86	99.86	99.72
9	99.79	99.79	99.58
10	99.58	99.59	99.17
Average	99.71	99.71	99.48

Table 35: Accuracy, F-measure, and MCC rates of AlexNetDense on Millet dataset

Test Nº	Accuracy (%)	F-measure (%)	MCC (%)
1	99.93	99.93	99.86
2	100.0	100.0	100.0
3	99.93	99.93	99.86
4	100.0	100.0	100.0
5	100.0	100.0	100.0
6	99.93	99.93	99.86
7	99.79	99.79	99.58
8	100.0	100.0	100.0
9	99.93	99.93	99.86
10	99.93	99.93	99.86
Average	99.94	99.94	99.88

Table 36: Accuracy, F-measure, and MCC rates of AlexNetDense and some CNN models

CNN	Accuracy	F-measure	MCC	Average
LeNet	99.55	99.56	99.11	99.40
AlexNet	99.79	99.79	99.58	99.72
VGGNet	99.73	99.73	99.47	99.64
ResNet	99.79	99.79	99.58	99.72
DenseNet	99.85	99.85	99.70	99.80
EfficientNet	99.71	99.71	99.48	99.63
AlexNetDense	99.94	99.94	99.88	99.92

On millet dataset, from the results observed (Table 36) and Fig. (9), it is noticed that the proposed architecture AlexNetDense recorded a higher average rate of 99.92% calculated from measurements of means accuracy (99.94%), F-measure (99.94%) and MCC (99.88%). Result analysis demonstrates that the remaining architectures behave well for the taken dataset. And, the average rate for DenseNet, ResNet, AlexNet, VGGNet, EfficientNet, and LeNet is achieved as 99.80, 99.72, 99.72, 99.64, 99.63, and 99.40%, respectively. The Millet dataset has only two classes. For this reason, testing rates are close to 100% for AlexNetDense architecture. This proposed system can be extended to recognize several other types of diseases found in Millet at an early stage. This can additionally prove the reliability of this architecture on millet crops cultivated in the Sahelian region like Niger where climatic conditions and disease symptoms on plant leaves are different.

According to all results obtained, the proposed architecture performs better than those proposed in the state-of-the-art on both PlanVillage and Millet Leaf datasets. In addition, the execution time for all the models was analyzed with the PlantVillage and 1000 epochs. Table (37) and Fig. (10) also compare the proposed model with other state-of-the-art CNN networks using computational time.

Table 37: Comparison analysis of the CNN models under computation time on the PlantVillage dataset

CNN	Time (s)
LeNet	492.31
AlexNet	867.81
VGGNet	7076.59
ResNet	4869.29
DenseNet	4622.73
EfficientNet	1716.49
AlexNetDense	1709.88

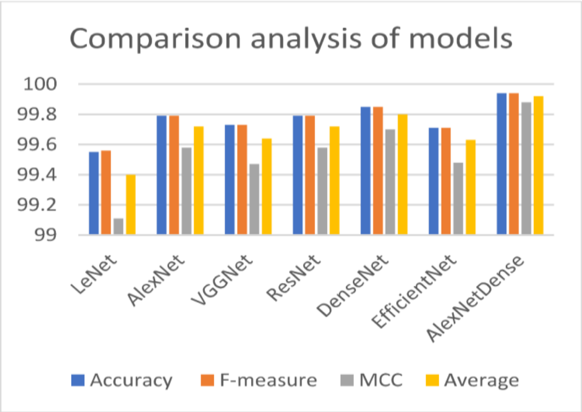

Fig. 9: Comparison of CNN models on millet dataset using accuracy and F-measure

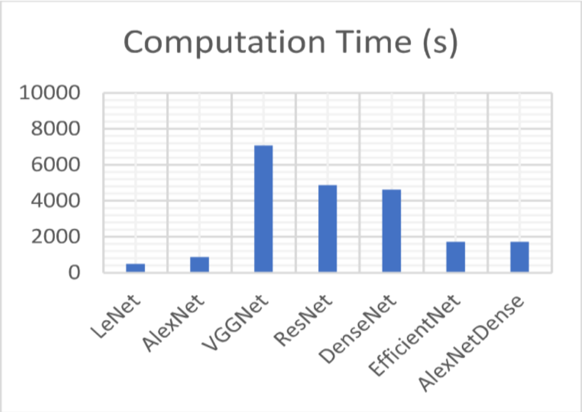

Fig. 10: Comparison analysis of the models under computation time

Table (37) and Fig. (10) show the evaluation of the models under computation time. On the other hand, when the training times of the models per epoch were analyzed, LeNet had the lowest computation time on the training dataset, with 492.31 s, but its average accuracy and Fmeasure of 92.46% and 92.51% respectively, were behind other models. Furthermore, the VGGNet approach performs well, with the longest computation time of 7076.59 s. In addition, it was very clear that the proposed AlexNetDense model achieved the highest average accuracy and F-measure during acceptable computation time which is 1709.88 s on training datasets for 1000 epochs. Nevertheless, the proposed model was quite faster than DenseNet, ResNet, EfficientNet, and VGGNet.

Conclusion

In recent years, the use of Convolutional Neural Network (CNN) algorithms for image analysis has gained significant attention. It can help identify plant leaf disease by enabling the development of accurate and efficient automated systems that can classify and diagnose plant diseases. Because of this, the CNN technique is an important branch of deep learning. The proposed work incorporates the novel CNN architecture for automated diagnosis and an accurate image-based classification approach. It replaced the dense connections of DenseNet with a technique in which the connections between layers are reduced.

And, this reduced the computational cost and effectively improved model performance for small datasets. The study consisted of four phases, namely phases of Data Collection, Data Processing, Data Training, and Testing. Experimental studies were conducted in both the PlantVillage dataset and the Millet Leaf dataset. Considering both the average accuracy and the average F-measure metric, the proposed model AlexNetDense was found to be superior to other CNN architectures. It achieved 99.30% accuracy, 99.12% F-measure, and 98.90% MCC on the PlantVillage dataset and 99.94% accuracy, 99.94% F-measure, and 99.88% MCC on the Millet Leaf dataset. In the future, the presented research work can be extended to enhance the robustness of the model by incorporating more disease-infected images for various plants at different disease severity levels on Millet Leaf dataset taken from Niger. The proposed approach could be further developed for mobile settings, allowing plant farmers to quickly and accurately recognize plant pathogens and take necessary precautions to mitigate the impact of plant diseases.

Acknowledgment

We are thankful to the African Centre for Technology Studies (ACTS), Nairobi (KENYA) and Abdou Moumouni University of Niamey (NIGER) for facilitating the necessary support.

Funding Information

Scholarship funders: International Development Research Centre (IDRC) and Swedish International Development Cooperation Agency (SIDA) Scholarship Programme - Artificial Intelligence for Development (AI4D) Africa Scholarship Fund Manager - Africa Center for Technology Studies (ACTS).

Author's Contributions

Atsu Alagah Komlavi: Made considerable contributions to the design, acquisition of data, analysis, and interpretation of data; Contributed to drafting the article and reviewing it critically for significant intellectual content; Prepare the final version to be submitted and any revised version.

Harouna Naroua: Coordinated the research activity and made considerable contributions to the design, analysis, and interpretation of data; Contributed to reviewing the article critically for significant intellectual content; Gave final approval of the version to be submitted and any revised version.

Chaibou Kadri: Made considerable contributions to the design, analysis, and interpretation of data; Contributed to reviewing the article critically for significant intellectual content; Gave final approval of the version to be submitted and any revised version.

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and that no ethical issues are involved.

Conflict of Interest

The author has disclosed no possible conflicts of interest.

References

Abdoul Habou, Z., Boubacar, M. K., & Adam, T. (2016). Les Systèmes De Productions Agricoles Du Niger Face Au Changement Climatique : Défis Et Perspectives. *International Journal of Biological and Chemical Sciences*, *10*(3), 1262–1272. https://doi.org/10.4314/ijbcs.v10i3.28

Amara, J., Bouaziz, B., & Algergawy, A. (2017). A Deep Learning-Based Approach for Banana Leaf Diseases Classification. In *(Hrsg.): BTW 2017 – Workshopband.*

Anak Entuni, C. J., & Zulcaffle, T. M. A. (2022). Identification of Corn Leaf Diseases Comprising of Blight, Grey Spot and Rust Using DenseNet-201. *Borneo Journal of Resource Science and Technology*, *12*(1), 125–134. https://doi.org/10.33736/bjrst.4224.2022

Arya, S., & Singh, R. (2019). An Analysis of Deep Learning Techniques for Plant Leaf Disease Detection. *International Journal of Computer Science and Information Security (IJCSIS)*, *17*(7), 73–80.

Atila, Ü., Uçar, M., Akyol, K., & Uçar, E. (2021). Plant Leaf Disease Classification Using Efficient Net Deep Learning Model. *Ecological Informatics*, *61*, 101182. https://doi.org/10.1016/j.ecoinf.2020.101182

Deeba, K., & Amutha, B. (2020). WITHDRAWN: ResNet - Deep Neural Network Architecture for Leaf Disease Classification. *Microprocessors and Microsystems*, 103364. https://doi.org/10.1016/j.micpro.2020.103364

Dheeraj, A., & Chand, S. (2023). Deep Learning Model for Automated Image Based Plant Disease Classification. *Proceedings in Adaptation, Learning and Optimization*, 21–32. https://doi.org/10.1007/978-3-031-31164-2_3

Glorot, X., Bordes, A., & Bengio, Y. (2011). Deep Sparse Rectifier Neural Networks. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 315–323.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S., & Lew, M. S. (2016). Deep Learning for Visual Understanding: A Review. *Neurocomputing*, *187*, 27–48. https://doi.org/10.1016/j.neucom.2015.09.116

Hawey, T., Sitou, L., Manssour, A. M., Soumana, I., Massaoudou, M., & Zoubeirou, A. M. (2020). Risques Pluviométriques, Source D'insécurité Alimentaire Et Nutritionnelle Au Niger. *International Journal of Biological and Chemical Sciences*, *14*(2), 452–464. https://doi.org/10.4314/ijbcs.v14i2.12

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770–778. https://doi.org/10.1109/cvpr.2016.90

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely Connected Convolutional Networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4700–4708. https://doi.org/10.1109/cvpr.2017.243

Hughes, D., & Salathe, M. (2015). An Open Access Repository of Images on Plant Health to Enable the Development of Mobile Disease Diagnostics. *ArXiv*, arXiv:1511.08060.

Jahan, N., Flores, P., Liu, Z., Friskop, A., Mathew, J. J., & Zhang, Z. (2020). Detecting and Distinguishing Wheat Diseases Using Image Processing and Machine Learning Algorithms. *2020 ASABE Annual International Virtual Meeting 2000372*, 1. https://doi.org/10.13031/aim.202000372

Ju, R.-Y., Chiang, J.-S., Chen, C.-C., & Lin, Y.-S. (2022). Connection Reduction of DenseNet for Image Recognition. *IEEE*, 1–4.

Khatoon, S., Maruf Hasan, M., Asif, A., Alshmari, M., & Yap, Y.-K. (2021). Image-Based Automatic Diagnostic System for Tomato Plants Using Deep Learning. *Computers, Materials and Continua*, *67*(1), 595–612. https://doi.org/10.32604/cmc.2021.014580

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2017). ImageNet Classification with Deep Convolutional Neural Networks. *Communications of the ACM*, *60*(6), 84–90. https://doi.org/10.1145/3065386

Kumar, A., & Vani, M. (2019). Image Based Tomato Leaf Disease Detection. *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 1–6. https://doi.org/10.1109/icccnt45670.2019.8944692

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

Luaibi, A. R., Salman, T. M., & Miry, A. H. (2021). Detection of Citrus Leaf Diseases Using a Deep Learning Technique. *International Journal of Electrical and Computer Engineering (IJECE)*, *11*(2), 1719–1727. https://doi.org/10.11591/ijece.v11i2.pp1719-1727

Maeda-Gutiérrez, V., Galván-Tejada, C. E., Zanella-Calzada, L. A., Celaya-Padilla, J. M., Galván-Tejada, J. I., Gamboa-Rosales, H., Luna-García, H., Magallanes-Quintanar, R., Guerrero Méndez, C. A., & Olvera-Olvera, C. A. (2020). Comparison of Convolutional Neural Network Architectures for Classification of Tomato Plant Diseases. *Applied Sciences*, *10*(4), 1245. https://doi.org/10.3390/app10041245

Marzougui, F., Elleuch, M., & Kherallah, M. (2020). A Deep CNN Approach for Plant Disease Detection. *2020 21st International Arab Conference on Information Technology (ACIT)*, 1–6. https://doi.org/10.1109/acit50332.2020.9300072

Matin, Md. M. H., Khatun, A., Moazzam, Md. G., & Uddin, M. S. (2020). An Efficient Disease Detection Technique of Rice Leaf Using AlexNet. *Journal of Computer and Communications*, *8*(12), 49–57. https://doi.org/10.4236/jcc.2020.812005

Michelucci, U. (2018). *Applied Deep Learning* (1st Ed.). Apress. https://doi.org/10.1007/978-1-4842-3790-8

Montalbo, F. J. P., & Hernandez, A. A. (2020). An Optimized Classification Model for Coffea Liberica Disease Using Deep Convolutional Neural Networks. *2020 16th IEEE International Colloquium on Signal Processing and Its Applications (CSPA)*, 213–218. https://doi.org/10.1109/cspa48992.2020.9068683

Moussa Dit Kalamou, M., Lebailly, P., & Yamba, B. (2016). Land Issue Around the Lands Recovered in the Region of Tahoua in Niger. *AGROFOR - International Journal*, *1*(1), 195–201. https://doi.org/10.7251/agreng1601195k

Rajeena, P. P., F., S. U., A., Moustafa, M. A., & Ali, M. A. S. (2023). Detecting Plant Disease in Corn Leaf Using EfficientNet Architecture—An Analytical Approach. *Electronics*, *12*(8), 1938. https://doi.org/10.3390/electronics12081938

Rangarajan, A. K., Purushothaman, R., & Ramesh, A. (2018). Tomato Crop Disease Classification Using Pre-Trained Deep Learning Algorithm. *Procedia Computer Science*, *133*, 1040–1047. https://doi.org/10.1016/j.procs.2018.07.070

Rinu, R., & Manjula, S. H. (2021). Plant Disease Detection and Classification Using CNN. *International Journal of Recent Technology and Engineering (IJRTE)*, *10*(3), 152–156. https://doi.org/10.35940/ijrte.C6458.0910321

Suryawati, E., Sustika, R., Yuwana, R. S., Subekti, A., & Pardede, H. F. (2018). Deep Structured Convolutional Neural Network for Tomato Diseases Detection. *2018 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 385–390. https://doi.org/10.1109/icacsis.2018.8618169

Tan, M., & Le, Q. (2019). EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *PMLR Proceedings of Machine Learning Research*, 6105–6114.

Too, E. C., Yujian, L., Njuki, S., & Yingchun, L. (2019). A Comparative Study of Fine-Tuning Deep Learning Models for Plant Disease Identification. *Computers and Electronics in Agriculture*, *161*, 272–279. https://doi.org/10.1016/j.compag.2018.03.032

Wang, A. R., & Shabrina, N. H. (2023). A Deep Learning-Based Mobile App System for Visual Identification of Tomato Plant Disease. *International Journal of Electrical and Computer Engineering (IJECE)*, *13*(6), 6992–7004. https://doi.org/10.11591/ijece.v13i6.pp6992-7004