AgriEngineering

Article

Leveraging Deep Learning for Real-Time Coffee Leaf Disease Identification

Opeyemi Adelaja and Bernardi Pranggono *

School of Computing and Information Science, Anglia Ruskin University, Cambridge CB1 1PT, UK
* Correspondence: bernardi.pranggono@aru.ac.uk

Abstract: Agriculture is vital for providing food and economic benefits, but crop diseases pose significant challenges, including coffee cultivation. Traditional methods for disease identification are labor-intensive and lack real-time capabilities. This study aims to address existing methods' limitations and provide a more efficient, reliable, and cost-effective solution for coffee leaf disease identification. It presents a novel approach to the real-time identification of coffee leaf diseases using deep learning. We implemented several transfer learning (TL) models, including ResNet101, Xception, CoffNet, and VGG16, to evaluate the feasibility and reliability of our solution. The experiment results show that the proposed models achieved high accuracy rates of 97.30%, 97.60%, 97.88%, and 99.89%, respectively. CoffNet, our proposed model, showed a notable processing speed of 125.93 frames per second (fps), making it suitable for real-time applications. Using a diverse dataset of mixed images from multiple devices, our approach reduces the workload of farmers and simplifies the disease detection process. The findings lay the groundwork for the development of practical and efficient systems that can assist coffee growers in disease management, promoting sustainable farming practices, and food security.

Keywords: convolutional neural network; deep learning; disease identification; transfer learning

check for updates

Academic Editor: Zhengjun Qiu

Received: 13 November 2024
Revised: 24 December 2024
Accepted: 3 January 2025
Published: 8 January 2025

Citation: Adelaja, O.; Pranggono, B. Leveraging Deep Learning for Real-Time Coffee Leaf Disease Identification. *AgriEngineering* **2025**, *7*, 13. https://doi.org/10.3390/agriengineering7010013

1. Introduction

Agriculture plays a vital role in global food production and economic stability, with coffee cultivation being one of the most significant contributors to agricultural exports. Coffee *(Coffea arabica)* is grown in more than 70 countries, providing livelihoods for millions of people [1]. However, coffee production faces numerous challenges, including the threat posed by plant diseases. One of the most destructive diseases in coffee cultivation is coffee leaf rust, caused by the fungal pathogen Hemileia vastatrix, which leads to premature leaf drop, reduced photosynthesis, and decreased yields [2]. If not detected early and effectively managed, coffee leaf rust can cause widespread crop loss, significantly affecting small-scale farmers and the global coffee supply chain.

Traditional methods for identifying and diagnosing coffee leaf diseases are typically labor-intensive, relying on manual inspection of plants by skilled workers [3]. These approaches are not only time-consuming but also impractical for large-scale farming operations, where quick and accurate identification is essential to prevent disease outbreaks. Furthermore, traditional disease identification methods often lack the ability to provide real-time results, leading to delayed responses when applying the necessary treatments [4]. As the agricultural sector faces increasing demands for efficiency and sustainability, the need for automated, real-time disease identification solutions has become more pressing [5].

Recent advances in technology, particularly in the fields of machine learning and deep learning, have opened new avenues for automating plant disease identification [6]. convolutional neural networks (CNNs) have shown potential in analyzing plant images and identifying disease-related symptoms with high accuracy [7,8]. Leveraging these technologies can revolutionize disease management by enabling early detection, reducing the dependency on manual labor, and providing real-time insights to farmers.

This study aims to address the limitations of traditional methods by proposing a real-time coffee leaf disease identification system based on deep learning techniques. Specifically, we explore the use of transfer learning models, including ResNet101, Xception, CoffNet, and VGG16, to classify coffee leaf diseases. Transfer learning allows us to leverage pre-trained models, significantly reducing the required training data while maintaining high accuracy [9]. Significant progress in deep learning, particularly CNNs, has improved the diagnosis of plant diseases. The early identification and treatment of diseases in coffee plants, especially in leaves, are crucial in preventing potential agricultural disasters. Traditionally, the precision of disease diagnosis has depended on the expertise of plant pathologists and environmental conditions. However, advanced technology offers improved methods for disease identification and precise diagnosis, leading to a more clearly defined problem statement and a more focused and specialized research direction.

The goal is to provide a clear, reproducible, and unique setup that allows experts in the field of coffee disease to easily quantify the effects of different techniques. Implementing cost-effective algorithms that provide several opportunities for data-driven innovations in the coffee industry. All of these tasks will require time and ongoing testing and evaluation of the methods as they become available. Relying on a single technique immediately is not feasible. The evaluation will not only influence the progress in applying deep learning techniques to coffee leaf analysis but will also serve as a resource for new researchers and experts in the coffee leaf sector to appreciate the impact and reliability of the technique when used as a standard [10]. This will also provide a solution or cure for the disease identified in the coffee leaf after identification.

The real-time disease identification system can be deployed on mobile devices, tablets, or drones equipped with cameras. Farmers could use mobile applications that process images of coffee leaves to provide instant feedback on disease presence and severity. Integration with smart farming platforms that collect additional data (e.g., weather, soil conditions, etc.) could provide comprehensive crop health insights. To use the proposed system, farmers would need basic training in using mobile apps or web interfaces and capturing quality images of leaves under consistent lighting conditions. Tutorials on interpreting the system's feedback and understanding disease severity metrics would also be essential.

In summary, the key contributions of this article are as follows:

- A survey on coffee leaf disease identification is presented, highlighting the key techniques to identify diseases.
- We introduce CoffNet, a novel deep learning model designed specifically for the identification of coffee leaf disease. CoffNet outperforms existing models in terms of processing speed, achieving 125.93 frames per second (fps), making it ideal for real-time applications.
- We evaluate the performance of several state-of-the-art CNN architectures, providing a comprehensive comparison of their accuracy and suitability for real-time deployment in agricultural settings.
- Using a diverse dataset of images captured from various devices, our approach aims to deliver a practical, scalable, and cost-effective solution for coffee growers.

The remainder of this paper is organized as follows: Section 2 presents the related work on the identification of plant diseases. Section 3 outlines the methodology employed in the study, such as dataset and data preprocessing. The results of our experiment are discussed and analyzed in Section 4. Finally, in Section 5, the conclusions are drawn.

2. Related Work

2.1. Disease Identification Techniques

Accurate disease diagnosis is a crucial component in successfully managing and preventing loss of production in coffee production. Conventional approaches to detecting coffee leaf diseases mostly include visual examination by skilled professionals or laboratory procedures, which may be challenging, subjective, and susceptible to mistakes [11]. Advanced technology has been used to investigate creative ways to improve the accuracy and efficiency of identifying diseases in coffee leaves.

2.1.1. Image-Based Analysis

Computer vision and machine learning algorithms are used in image-based analysis approaches to examine digital photographs of crops to identify and categorize diseases [12].

2.1.2. Spectroscopy and Hyperspectral Imaging

Spectroscopy and hyperspectral imaging methods are used to examine the interaction between light and plant tissues for the identification and identification of diseases. These approaches could detect and record spectral patterns across a broad spectrum of wavelengths, including visible, near-infrared, and short-wave infrared areas [13]. Through the analysis of distinct spectral patterns linked to various diseases, these methods have the capability to provide early identification and diagnosis, even before the manifestation of apparent symptoms.

2.2. Molecular Techniques

Molecular methods, such as polymerase chain reaction (PCR) and enzyme-linked immunosorbent assay (ELISA), are used to detect and characterize certain pathogens or disease-causing chemicals in coffee leaves [14]. These procedures depend on the identification of certain genetic or protein markers linked to infections, which ensures very precise and dependable outcomes. However, these strategies can incur higher costs and require more time compared to other approaches.

Remote Sensing and Aerial Imaging

Remote sensing and aerial imaging methods, such as satellite imagery or drone-based imaging, can be used to monitor and identify coffee leaf diseases on a wide scale in extensive plantations or fields. These tools provide useful information on the geographical distribution and dissemination of diseases, allowing for focused treatments and effective allocation of resources [15].

2.3. Coffee Leaf Disease Identification

Maintaining crop health is crucial for maintaining agricultural production and ensuring food security. Diseases affecting plant leaves pose a significant threat to crop cultivation. The early identification and treatment of these diseases are vital to prevent yield losses and preserve crop health.

In [16], Mengistu et al. categorized coffee leaf diseases into three primary types: coffee wilt disease (CWD), coffee berry disease (CBD), and coffee leaf rust (CLR). Initially, the study applied the Gray Level Co-Occurrence Matrix (GLCM) alongside color features to extract relevant features; subsequently, it employed four machine learning algorithms:

Naïve Bayes, k-Nearest Neighbors (KNNs), Artificial Neural Network (ANN), and a hybrid of a Self-Organizing Map (SOM) and Radial Basis Function (RBF) for classifying diseases affecting coffee plant leaves. Their study shows that the combination of SOM and RBF outperformed other algorithms with an accuracy of 90.07%.

The use of a genetic algorithm to identify rust in coffee leaves was introduced by Marcos et al. [17]. Their research specifically utilized a genetic algorithm to determine the optimal convolution kernel mask to enhance the texture and color characteristics of fungal infections.

Manso et al. [18] proposed an application that detects coffee leaf diseases in smartphone images. The study analyzed various background types for images using the YCbCr (luminance and chrominance) and HSV (hue, saturation, and value) color spaces during the segmentation process and compared these with k-means clustering in the YCbCr color space. The iterative threshold algorithm employed, known as the Otsu algorithm, calculates the extent of damage caused by the diseases of coffee plants. For classification during foliar damage segmentation, an ANN trained with backpropagation and extreme learning machine (ELM) was used. Their study revealed the best accuracy of 99.095%.

The YOLOv3-MobileNetv2 model was used to detect diseases in robusta coffee leaves in the study by Javierto et al. [19]. They created a prototype capable of capturing input images and classifying the diseases into four categories: Cercospora, Miner, Phoma, and Rust.

Esgario et al. [20] used 1747 images of Arabica leaves to train multiple deep convolutional neural networks (including VGG16 and ResNet50) to classify the severity level and biotic stress. The VGG16 DCNN, trained to identify various biotic diseases, attained an accuracy of 95.47%, while the ResNet50 model validated the condition of each leaf with an accuracy rate of 95.63%.

A method that integrates remote sensing (RS), wireless sensor networks (WSNs), and deep learning (DL) was introduced by Velasquez et al. in [21]. This study achieved an F1 score of 77.5%.

Marin et al. [22] proposed a framework employing decision tree models (Logistic Model Tree (LMT), K48, ExtraTree, REPTree, Functional Trees, Random Tree, and Random Forest (RF)) to assess coffee leaf rust severity using vegetation indices derived from multispectral images obtained via a remotely piloted aircraft (RPA). Their study revealed that the LMT approach was the most efficient in identifying CLR disease, achieving an F1 score of 91.5%.

A guided approach to detect coffee diseases using various visualization techniques (Grad-CAM, Grad-CAM++, and Score-CAM) is proposed in [23]. The RoCoLE database is used in this research. The dataset consists of 1560 Robusta coffee leaf images. The study showed that the proposed guided method achieved a classification accuracy of 98%.

Rodriguez et al. [15] introduced a method for detecting coffee leaf rust utilizing two datasets: RoCoLE and a collection of images captured via unmanned aerial vehicles (UAVs). The study used ImageJ for image processing and Python for rust identification. Their approach demonstrated a detection accuracy of 97% on the RoCoLE dataset and over 93.5% on the UAV imagery.

Several hybrid models were proposed by Fasisal et al. in [24] to extract features from input images using a combination of Swin Transformer, MobileNetV3, and variational autoencoder (VAE). Their study used the RoCoLe database and shows that the fusion of the hybrid features of Swin Transformer and MobileNetV3 resulted in the detection of CLR with an accuracy of 84.29%.

The impact of five types of hyperparameters on the performance of CLR classification models was studied by Chavarro et al. [25]. Their study used multiple databases: RoCoLE, BRACOL, D&P, Digipathos, and Locole. The results show that DenseNet201 produces the best results with an accuracy of 94.60%.

RGB aerial imagery and machine learning to detect coffee leaf miner were used by Vilela et al. in [26]. The study used four machine learning algorithms: Random Forest (RF), Logistic Regression (LR), Support Vector Machine (SVM), and Stochastic Gradient Descent (SGD). Their study showed that SVM and SGD models performed better compared to other models.

While molecular techniques such as PCR or ELISA offer high specificity, they require costly reagents, specialized equipment, and trained personnel. In contrast, a deep learning approach utilizes readily available hardware, such as smartphones. This reduces initial and ongoing costs. Smartphones with cameras, commonly available even in remote farming communities, can be leveraged as the primary hardware, reducing the need for specialized equipment. By enabling real-time detection, the approach minimizes delays in diagnosis, allowing farmers to take prompt action. This can significantly reduce disease spread and associated yield losses. Supporting small-scale farmers with accessible technology promotes social equity, helping them compete with larger agricultural operations. A summary of the coffee leaf disease identification studies is shown in Table 1.

Table 1. A summary of coffee leaf disease identification study.

Year & Reference	Methods	Dataset/Number of Images	Performance
2016 [16]	ANN, KNN, Naive Bayes, SOM+RBF	9100	Accuracy: 90.07%
2019 [18]	ANN	690	Accuracy: 99.095%
2020 [20]	VGG16, ResNet50	1747	Accuracy: 95.65%
2020 [21]	RS+WSN+DL	-	F1 score: 77.5%
2021 [23]	DNN+(Grad-CAM, Grad-CAM++, Score-CAM)	RoCoLE/1560	Accuracy: 98%
2021 [22]	LMT, K48, ExtraTree, REPTree, FunctionalTrees, Random Tree, RF	400	F1 score: 91.5%
2023 [24]	Swin Transformer, MobileNetV3, and VAE	RoCoLE/1560	Accuracy: 84.29%
2023 [25]	VGG19, Xception, ResNet50, InceptionV3, DenseNet201	RoCoLE, BRACOL, D&P, Digipathos, and Locole	Accuracy: 94.60%
2023 [15]	Python-based rust detection	RoCoLE/1560	Accuracy: 97%

3. Materials and Methods

This section describes the systematic methodology used to enhance the identification of diseases in horticultural plants, with a specific emphasis on coffee leaf diseases, using convolutional neural networks (CNNs). Our primary objective is to develop a CNN model, named CoffNet, capable of accurately and quickly recognizing and classifying diseases in coffee leaf images.

The framework of the primary model is shown in Figure 1. This diagram highlights that the framework consists of two core elements: image preprocessing and deep learning-based classification.

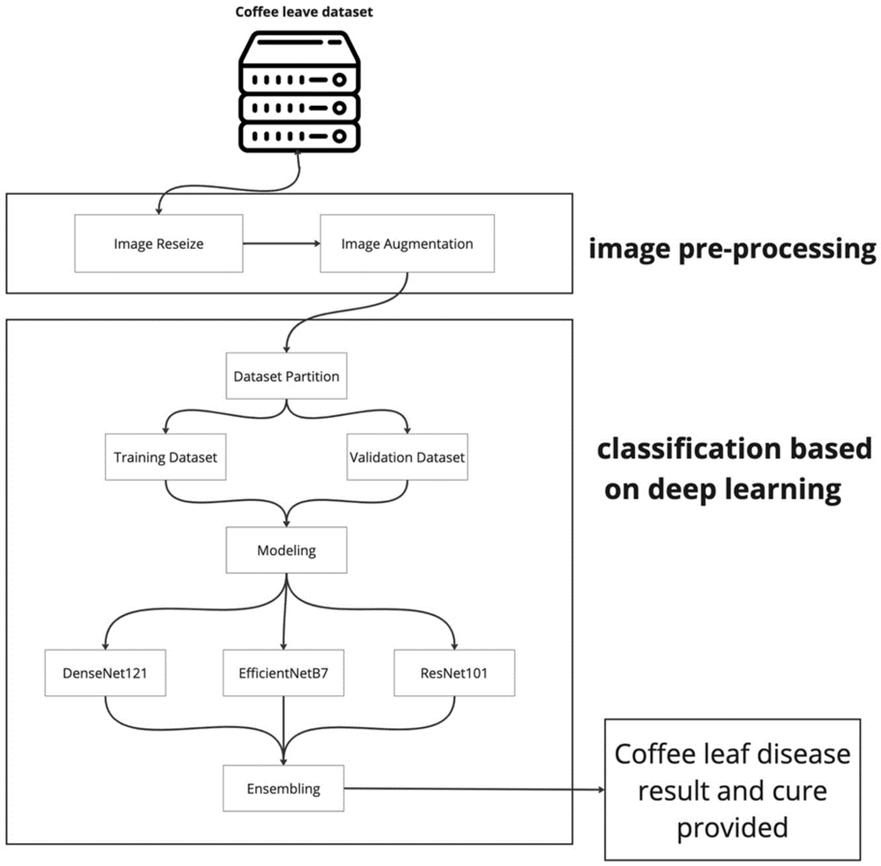


Figure 1. Proposed methodology workflow.

3.1. Dataset

This study made use of a freely available BRACOL dataset that is provided by Krohling et al. [27] and is accessible on Mendeley Data. All images were captured using various smartphones (ASUS Zenfone 2, Xiaomi Redmi 5A, Xiaomi S2, Galaxy S8, and iPhone 6S). The Santa Maria of Marechal Floreano in Brazil's hilly state of Espirito Santo is where the leaves were collected at different times of the year. The images were taken from the abaxial (underside side) of the leaves under controlled conditions and placed against a white background. The photos were acquired without a specific criterion to increase the heterogeneity of the collection. A comprehensive collection of 1747 images depicting Arabica coffee leaves was compiled, featuring both healthy specimens and those affected by various types of biotic stress. An expert was engaged to facilitate the identification of these biotic stresses, ensuring the accurate labeling of the dataset through the analysis of the captured images. The images were labeled based on the main biotic stress affecting each leaf and its severity level. The severity was determined using automatic image processing methods, as described in [18], in conjunction with the symptom and leaf segmentation masks. Labels were assigned according to specific severity ranges: healthy (<0.1%), very low (0.1–5%), low (5–10%), high (10–15%), and very high (>15%) [27].

From the obtained photos, two datasets were generated: a dataset with the original images of the entire leaves and a second one containing only images of symptoms. The distribution of photos in each of the compressed files in the dataset is shown in Table 2 below. The dataset has a total of 19,599 leaves, which are categorized into five classes of disease. The class with the largest number of leaves is coffee Cerscospora, with 4070 instances. Healthy leaves come next with 3925 instances, followed by leaf rust with 3893 instances.

Coffee Phoma has 3891 instances, while coffee miner has the lowest number of instances, with 3820.

To address potential variations in image quality, the following steps were taken:

- Data Preprocessing: images were standardized by cropping and resizing to a uniform resolution (224 × 224 pixels), ensuring consistency across all samples regardless of the device used.
- Data Augmentation: techniques such as rotation, flipping, and zooming were applied to enhance the dataset's diversity and help the model generalize better to varying image qualities.
- Normalization: pixel values were normalized to reduce the impact of lighting and color variations introduced by different camera sensors.
- Robust Model Selection: transfer learning models like CoffNet, Xception, and ResNet101 were chosen for their proven ability to handle diverse image features and adapt to noise or variability.

Table 2. Coffee leaf condition and corresponding image distribution.

Coffee Leaf Condition	Description	Number of Images	Data Distribution (%)
Cerscospora	The appearance of circular to irregular brown spots on the upper surface of coffee leaves	4070	20.8
Healthy	A uniform green color with no spots or visible damage	3925	20.0
Leaf Rust	Characterized by the presence of orange-colored granular pustules on the lower surface of coffee leaves	3893	19.9
Phoma	The presence of asymmetrical brown or reddish-brown lesions on the leaves	3891	19.9
Miner	The existence of serpentine or winding mines inside the leaves	3820	19.5

3.2. Data Preprocessing

3.2.1. Cropping Image

Subsequently, each picture in the dataset was examined to see if it had a uniform square shape. The non-square photos were cropped to extract the central square portion of the image. The dimensionality of the image was verified to ensure that all the photos had the same dimensions. Images were standardized through cropping and resizing to a uniform resolution (224 × 224 pixels). An image editing tool called a crop tool was used to remove unwanted parts of the photos. Figures 2 and 3 depict a picture before and after cropping. Cropping improved the focus on the particular area of interest within the leaf, thereby decreasing the duration of training in image processing.

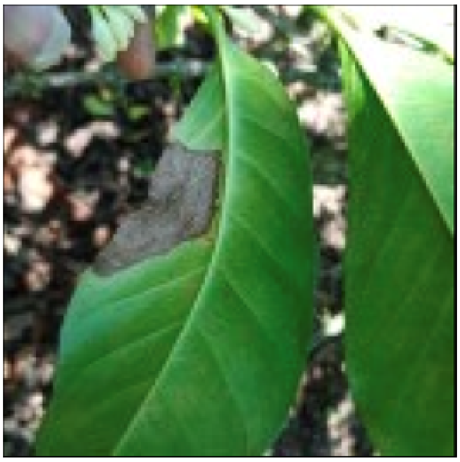


Figure 2. Coffee leaf image before cropping.

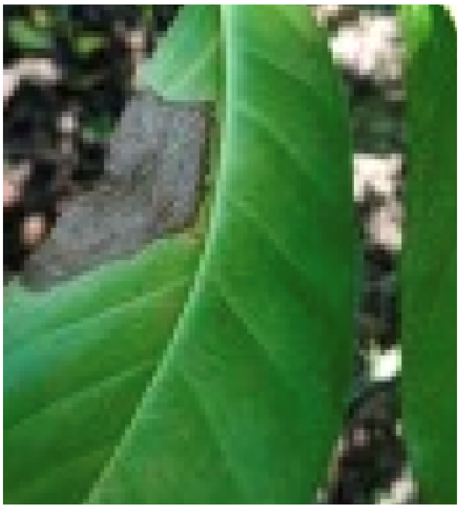


Figure 3. Coffee leaf image after cropping.

3.2.2. Data Augmentation

Data augmentation is a method used to artificially expand the amount and variety of a training dataset by implementing different changes to the current data samples. The initial data for this study were 4329, and the artificially added images were 15,270. This methodology is especially advantageous in computer vision assignments, such as picture categorization, where CNN is often used. Data augmentation is a technique that enhances the model's capacity to generalize, prevents overfitting, and improves its performance on data that it has not been trained on [28]. The image was rotated by 180° both horizontally and vertically to highlight the specific area afflicted by the disease. A picture flipper was used for flipping. Several images underwent horizontal flipping using a horizontal flipper, while others underwent vertical flipping using a vertical flipper.

To ensure that data augmentation transformations like flipping and rotating would not distort important disease features, the following steps were taken:

- Flipping (both horizontal and vertical) was applied only when the disease features (e.g., lesions or rust spots) did not rely on specific orientations. In cases where leaf patterns were symmetric, such transformations were safe and maintained the integrity of the disease features.
- Rotation was limited to 180° to avoid misaligning the disease features. Rotations were performed such that the disease symptoms remained in similar relative positions on the leaf, preserving key visual characteristics needed for accurate classification.

- Data augmentation techniques were carefully selected to avoid altering the shape or size of disease-related features. The focus was on transformations like zooming, which could simulate real-world variability without distorting key visual cues.

Gaussian noise was added during data augmentation to enhance model robustness by simulating real-world variability. Specifically, it was managed as follows:

- Low to moderate noise levels were used, typically with a mean of 0 and a standard deviation of 0.01 to 0.05. These values were chosen to ensure the noise simulated realistic conditions without overwhelming the disease features.
- Noise levels were incrementally tested during preprocessing. Performance metrics (accuracy and loss) were monitored to identify the optimal noise level that improved generalization without reducing classification accuracy.
- The final noise level was selected based on the validation set performance, ensuring the model maintained feature recognition integrity while reducing overfitting to the training data.

3.3. Classification Based on Deep Learning

Deep learning models, namely convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have shown outstanding performance in classifying image and sequence data. Deep learning has a significant benefit in that it can automatically extract and learn features straight from data, eliminating the need for manual feature selection. This allows the model to recognize intricate patterns. Nevertheless, these models need a significant volume of data and resources and often function as "black boxes", providing only a limited understanding of the decision-making process. Despite the difficulties, deep learning is still growing and opening new opportunities in the fields of ML and AI. This progress is fueled by improvements in computer power and innovative algorithms. Figure 4 below shows what a convolutional neural network looks like.

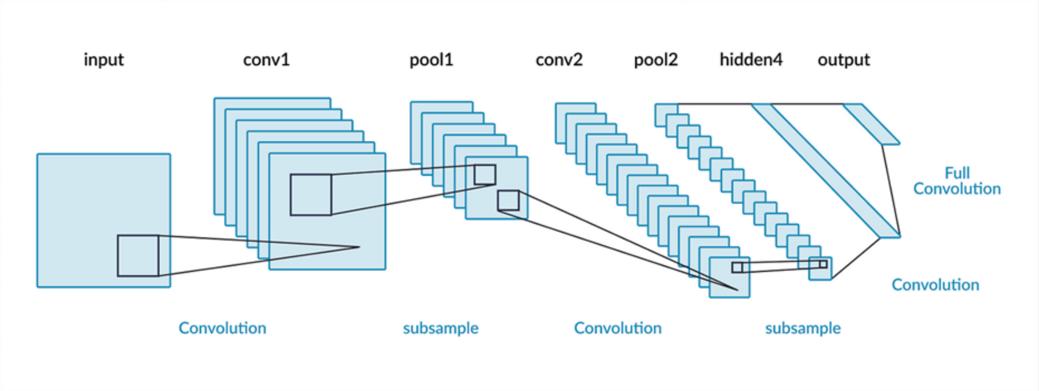


Figure 4. A graphical representation of a convolutional neural network.

3.3.1. Feature Extraction Model

A feature extraction model is a machine learning technique or algorithm that autonomously finds and extracts significant traits or properties from unprocessed data. This technique is essential for the preparation of data for classification or regression tasks. Feature extraction is a process that converts data into a format that is easier for models to grasp and more applicable to their needs. This process reduces the complexity of the data, making learning algorithms more efficient and boosting the performance of the models.

3.3.2. Convolution Map

A convolution map, often referred to as a feature map, is the result of a convolutional layer in the context of CNN. The term refers to the outcome obtained by performing a convolution operation on either the input data or the output of a preceding layer. Convolution maps are created by applying a filter or kernel to the input picture or feature map. This process captures patterns such as edges, textures, or spatial correlations in the data.

3.3.3. Max-Pool Map

"Max-pooling map" refers to a technique used in computer vision and deep learning to down-sampling an input feature map by selecting the maximum value inside a certain region. This helps to reduce the spatial dimensions of the feature A max-pooling map is the outcome of performing a max-pooling operation inside a convolutional neural network. Max-pooling is a method used to decrease the size of feature maps, which in turn reduces the number of parameters and computations in the network. The process involves partitioning the input feature map into a collection of non-overlapping rectangles and extracting the highest value from each rectangle as the output. This method enhances the ability to recognize features in an input picture by making them unaffected by minor shifts and distortions.

3.4. Modeling

3.4.1. Transfer Learning

Transfer learning (TL) is a powerful machine learning technique that allows for the transfer of knowledge from one task to another, which are closely related. It is especially valuable in deep learning, where training large models from scratch can be quite costly in terms of computation and the need for extensive labeled data. The process involves a pre-trained model, which is used as a feature extractor and fine-tuned to suit the new task. The modified model is then trained on the new task's dataset, updating the weights of the final layers to adapt to the new task. Transfer learning can be applied in scenarios like insufficient training data, faster convergence, domain adaptation, and task transfer. However, it is crucial to ensure that the source and target tasks are sufficiently related for effective knowledge transfer.

3.4.2. Xception

Xception is a complex CNN structure that was presented by François Chollet in 2017 [29]. The architecture is an extension of the Inception model, specifically optimized for improved performance on picture classification tasks using the ImageNet dataset. Xception's fundamental novelty is in the use of depthwise separable convolutions, a technique that effectively decreases the number of parameters and computing burden while simultaneously preserving or enhancing the performance when compared to conventional convolutional layers. Xception was chosen for its efficiency in parameter utilization via depthwise separable convolutions, achieving high accuracy with reduced computational complexity.

The Xception architecture has three modules: Entry Flow, Middle Flow, and Exit Flow. The Entry Flow module oversees collecting low-level characteristics, whereas the Middle Flow module obtains high-level characteristics by using a sequence of depthwise separable convolutions. The Exit Flow module utilizes completely linked layers to perform spatial pooling and classification.

We used the Xception model that was pre-trained on the ImageNet dataset for our study. Figure 5 depicts the architecture of our model, which incorporates the pre-trained Xception model.

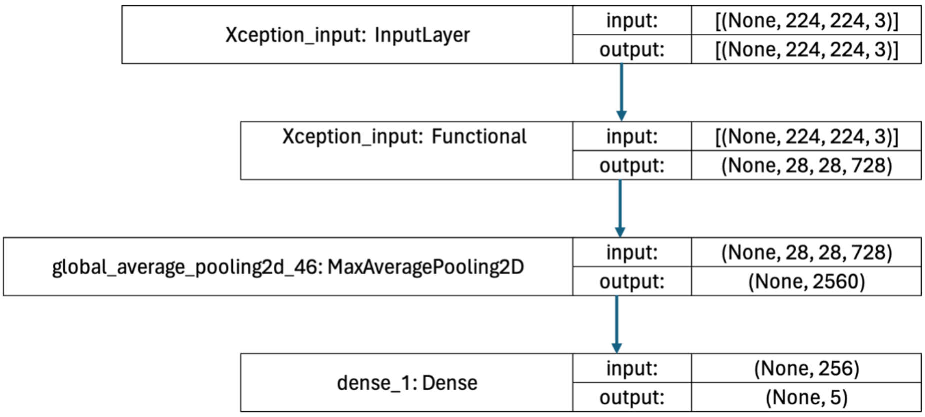

Xception_input: InputLayer
input: [(None, 224, 224, 3)]
output: [(None, 224, 224, 3)]
Xception_input: Functional
input: [(None, 224, 224, 3)]
output: (None, 28, 28, 728)
global_average_pooling2d_46: MaxAveragePooling2D
input: (None, 28, 28, 728)
output: (None, 2560)
dense_1: Dense
input: (None, 256)
output: (None, 5)

Figure 5. Xception structural block.

3.4.3. ResNet101

ResNets are designed with the primary concept of including "skip connections" or "residual connections" that enable the network to circumvent certain levels during the forward pass. This feature allows for efficient transmission of gradients during the back-propagation process, which, in turn, makes it easier to train deeper networks without encountering the issue of degradation. ResNet101 was selected for its ability to train deep networks effectively through residual connections, reducing vanishing gradient issues.

ResNet101 is a CNN architecture that is part of the Residual Networks (ResNets) family. ResNet101 is a variation of the ResNet architecture that has 101 layers. It is composed of convolutional layers, batch normalization layers, and ReLU activations. The network uses a bottleneck architecture, where each residual block has three convolutional layers: 1×1, 3×3, and 1×1 convolutions. The first 1×1 convolution decreases the dimensionality, followed by the 3×3 convolution for extracting features, and the last 1×1 convolution restores the dimensionality. Like previous ResNet models, ResNet101 employs global average pooling before reaching the final fully connected layer and uses a SoftMax activation function for classification tasks.

In this research, we used the ResNet101 variant of DenseNet pre-trained on the ImageNet dataset. Figure 6 depicts the structure of our model used in the pre-trained ResNet101 model.

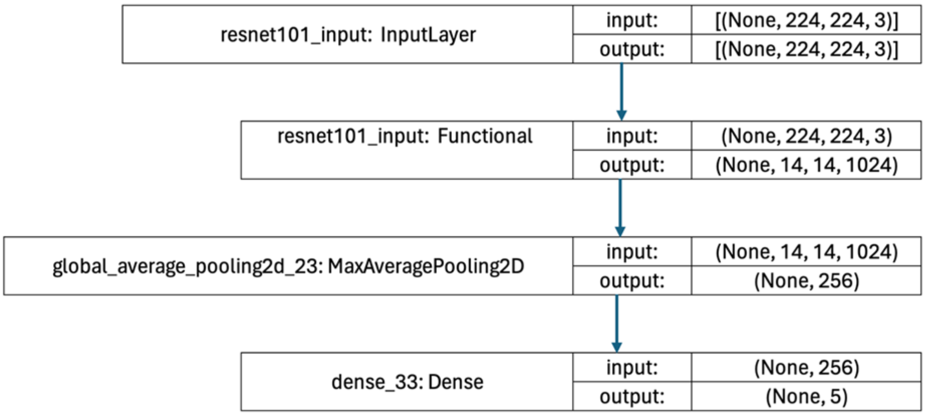

resnet101_input: InputLayer
input: [(None, 224, 224, 3)]
output: [(None, 224, 224, 3)]
resnet101_input: Functional
input: (None, 224, 224, 3)
output: (None, 14, 14, 1024)
global_average_pooling2d_23: MaxAveragePooling2D
input: (None, 14, 14, 1024)
output: (None, 256)
dense_33: Dense
input: (None, 256)
output: (None, 5)

Figure 6. ResNet101 structural block.

3.4.4. VGG16

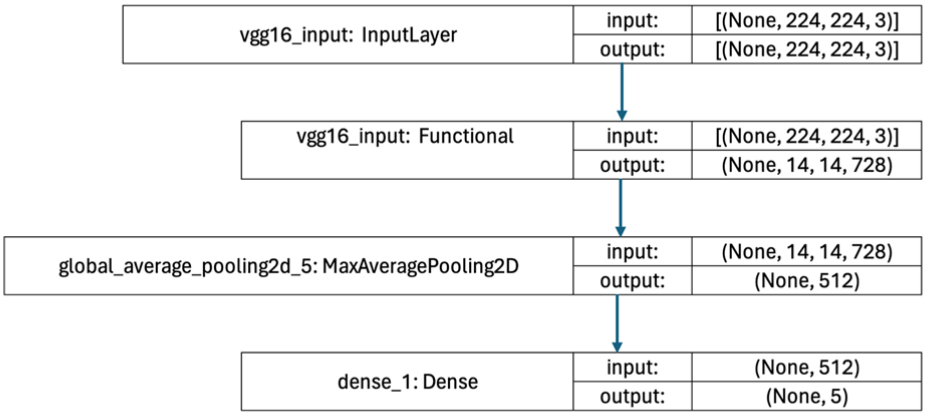
The VGG16 architecture was developed by Andrew Zisserman and Karen Simonyan [30]. VGG16 offers simplicity and a consistent structure that ensures robust feature extraction. The VGG architecture is distinguished by its straightforwardness and consistent structure, including a sequence of convolutional layers with compact (3 × 3) receptive fields, succeeded by max-pooling layers for down-sampling. Padding is used by the network to maintain the spatial resolution after convolution.

One notable aspect of VGG16 is its structure, consisting of 16 layers with adjustable weights, including 13 convolutional layers and 3 fully connected layers. The convolutional layers are organized into five blocks, each containing a series of convolutional layers followed by a max-pooling layer to effectively reduce the spatial dimensions.

Unlike more contemporary designs such as ResNets and DenseNets, VGG16 does not use skip connections or dense connections. Instead, it depends on the network's depth and the narrow receptive fields to acquire more intricate representations from the incoming data.

VGG16 employs rectified linear unit (ReLU) activations to introduce non-linearity and utilizes a SoftMax activation function in the final layer for the purpose of classification tasks. Although VGG16 is a simple model, it showed remarkable performance in the ImageNet challenge when it was first introduced. Since then, it has been extensively used as a benchmark or a tool for extracting features in other computer vision tasks, such as image classification, object recognition, and semantic segmentation.

In our research, we used the VGG16 model that was pre-trained on the ImageNet dataset for our study. Figure 7 depicts the architecture of our model, which incorporates the pre-trained VGG16 model.



vgg16_input: InputLayer | input: [(None, 224, 224, 3)] | output: [(None, 224, 224, 3)]

vgg16_input: Functional | input: [(None, 224, 224, 3)] | output: (None, 14, 14, 728)

global_average_pooling2d_5: MaxAveragePooling2D | input: (None, 14, 14, 728) | output: (None, 512)

dense_1: Dense | input: (None, 512) | output: (None, 5)

Figure 7. VGG16 structural block.

3.4.5. CoffNet

In deep learning, the models are part of the final ensemble. The model structure typically consists of several layers, including the input layer, functional layers (such as convolutional layers or recurrent layers), pooling layers, and dense layers. The pre-trained base model serves as the foundation, and its architecture is modified or extended to suit the target task. The CNN architecture of the CoffNet model is shown in Figure 8.

- Input Layer: The input layer remains unchanged from the pre-trained base model. It is responsible for accepting input data (e.g., images, text, or other data formats) in different dimensions and passing them to the subsequent layers. In this research, the dimension used to feed the model was 224 × 224 × 3.
- Convolutional Layers: These layers are responsible for extracting features and learning representations from the input data. In transfer learning, the lower-level functional layers are often frozen (the weights are not updated) or have a lower learning rate applied during training, as they have already learned useful low-level features from the source task.

- Global Average Pooling Layers: For each feature map, the global average pooling (GAP) layer computes the spatial average of the feature values, essentially reducing the feature map to a single scalar value. The resulting scalars from all feature maps are then concatenated into a vector, representing the final feature representation of the input. This vector is then fed directly into the dense layer.
- Dense Layers (Fully Connected Layers): Dense layers, also known as fully connected layers, are responsible for combining the learned features and making predictions or classifications. In transfer learning, the dense layers from the pre-trained base model are often replaced or fine-tuned to adapt to the target task. New dense layers may be added or modified to match the output size of the target task (e.g., number of classes for classification or dimensions for regression). Typically, the weights of these dense layers are initialized with pre-trained weights and then fine-tuned on the target dataset during training, just as a data scientist would do. This layer reduces the size to the five classification categories and applies SoftMax activation.

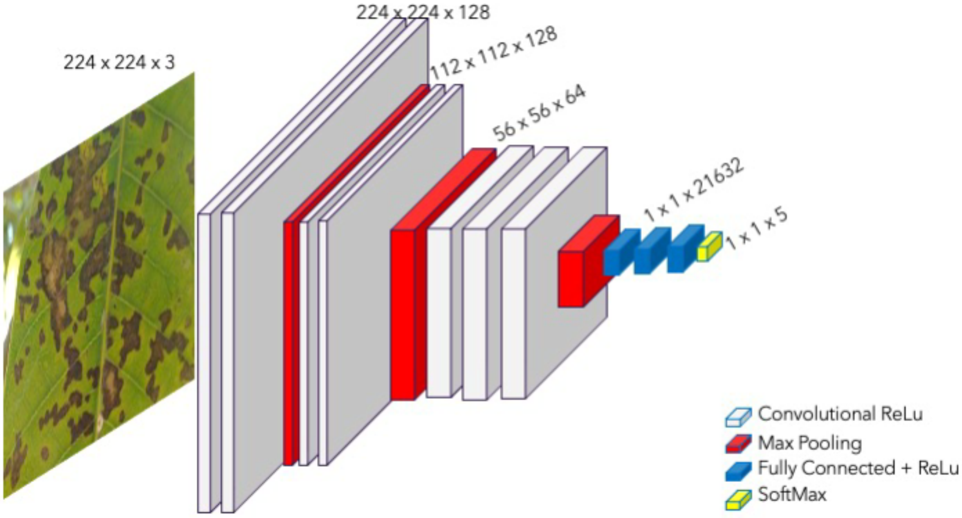


Figure 8. CNN architecture of the CoffNet model.

Using the SoftMax activation function, the output was normalized to create a probability distribution across the five classes. The final output was determined by selecting the class with the highest probability. Optimizing the network's weights was performed using the Adam Optimizer, which was selected for its proven efficiency and adaptive learning capabilities, as demonstrated in prior research [31]. The model was trained for 10 epochs using a batch size of 32.

3.5. Performance Evaluation Metrics

Applying deep learning (DL) methods to identify and categorize plant disease helps overcome the limitations of manually selecting disease spot characteristics. This technique makes the extraction of plant disease features more unbiased and improves research outcomes. In this study, various performance assessment criteria such as accuracy, recall, precision, F1 score, receiver operating characteristic (ROC), and area under the curve (AUC) were used [32–34]. Their equations establish and elucidate the methodology for assessing the efficacy of plant disease detection and classification approaches (see Equations (1)–(5)). In the equations, true positive (TP) means anomalous traffic correctly identified, true negative (TN) means normal traffic correctly identified, false positive (FP) means normal traffic incorrectly identified as anomalous, and false negative (FN) means anomalous traffic incorrectly identified as normal. The true positive rate (TPR) is also known as recall or sensitivity. The TPR is the number of true positive predictions divided by the number of actual positive cases. The false positive rate (FPR) is also known as the probability of false alarm. The FPR is the number of false positive predictions divided by the number of actual negative cases. The efficacy of the classifier is measured by the area under the ROC curve (AUC); an AUC of 1 signifies perfect classification, whereas an AUC of 0.5 suggests no better than random. The frames per second (fps) metric measures the image processing speed of a model, indicating how many images it can process in one second. This measurement is essential for evaluating processing efficiency and plays a pivotal role in enabling real-time disease detection.

All experiments were conducted using Python 3.8 on an Intel i5 CPU with 16 GB RAM and NVIDIA RTX 3080 GPU.

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

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

$$\text{TPR} = \frac{TP}{(TP + FN)} \tag{3}$$

$$\text{FPR} = \frac{FP}{(FP + TN)} \tag{4}$$

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

3.6. Model Performance and Evaluation

The detailed parameters used throughout the experiment are listed below (see Tables 3 and 4). The model was designed to classify images of coffee leaves into five classes based on the type of disease. The five classes are shown in Table 5.

Table 3. CNN architecture used.

Layer	Input Size	Output Size
conv2d	(None, 224, 224, 3)	(None, 222, 222, 128)
activation	(None, 222, 222, 128)	(None, 222, 222, 128)
max_pooling2d	(None, 222, 222, 128)	(None, 111, 111, 128)
dropout	(None, 111, 111, 128)	(None, 111, 111, 128)
conv2d_1	(None, 111, 111, 128)	(None, 109, 109, 64)
activation_1	(None, 109, 109, 64)	(None, 109, 109, 64)
max_pooling2d_1	(None, 109, 109, 64)	(None, 54, 54, 64)
dropout_1	(None, 54, 54, 64)	(None, 54, 54, 64)
conv2d_2	(None, 54, 54, 64)	(None, 52, 52, 32)
activation_2	(None, 52, 52, 32)	(None, 52, 52, 32)
max_pooling2d_2	(None, 52, 52, 32)	(None, 26, 26, 32)
dropout_2	(None, 26, 26, 32)	(None, 26, 26, 32)
flatten	(None, 26, 26, 32)	(None, 21632)
dense	(None, 21632)	(None, 5)

Table 4. CNN training paramaters.

Input size	224 × 224
Split	80:20
Training Image	12,543
Validation Image	3136
Test Image	3920
Optimizer	Adam (Adaptive Moment Estimation)

Table 5. Coffee leaf disease classes.

Class	Disease
0	Cercospora
1	Healthy
2	Leaf Rust
3	Miner
4	Phoma

Each figure accompanying the models compares the predicted disease class with the true disease class for every sample image. For example, if an image caption reads "Predicted: 4, True: 4, Accuracy: 99.42%", it indicates that the model correctly classified the image as belonging to class 4 (Phoma) and assigned it an accuracy of 99.42% (see Figure 9). This accuracy score reflects the model's confidence level for class 4, with the remaining percentage distributed across the other five classes. The model selected the class with the highest accuracy score as its prediction.

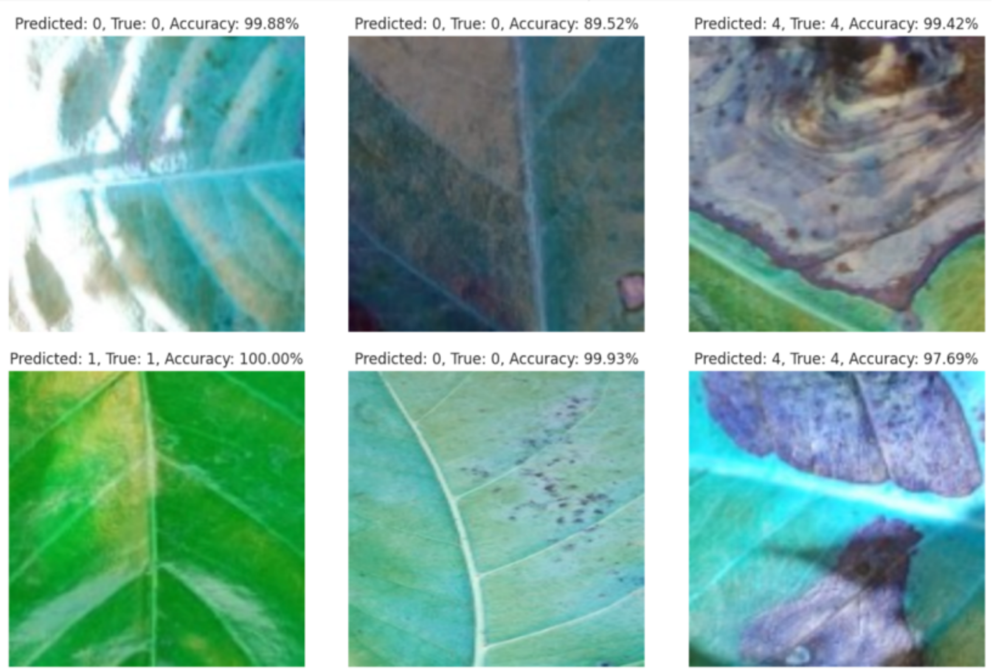

Predicted: 0, True: 0, Accuracy: 99.88%
Predicted: 0, True: 0, Accuracy: 89.52%
Predicted: 4, True: 4, Accuracy: 99.42%
Predicted: 1, True: 1, Accuracy: 100.00%
Predicted: 0, True: 0, Accuracy: 99.93%
Predicted: 4, True: 4, Accuracy: 97.69%

Figure 9. Xception model's results for test data.

To prevent overfitting, an early stopping method was implemented during training. This technique involved monitoring the performance of the validation set and stopping training once validation accuracy began to decline. This not only improved model performance by preserving generalization capability but also conserved computational resources.

4. Experiment Results

This section discusses the findings from various models, highlighting multiple performance metrics and their practical implications. The analysis of the results and observations indicates that models trained on larger resized images outperform those trained on smaller input sizes. Furthermore, minor noise in the images contributes to the high accuracy of the models, and Gaussian noise was specifically employed in this experiment.

4.1. Xception

Image categorization is efficient using Xception, a sophisticated CNN architecture. It has been frequently used with transfer learning, using pre-trained models on ImageNet, which includes millions of photos across thousands of classifications. An Xception model pre-trained on ImageNet was used to start a transfer learning image identification challenge. Even with lesser datasets, the pre-trained Xception model's rich learned feature representations supported quick training and great performance for the new task.

In the model development of the Xception model, the images were augmented before feeding the model with the data using a couple of data augmentation techniques like '*zoom_range, rotation_range, fill_mode, horizontal_flip*', and several other techniques were deployed to reduce overfitting and improve model generalization.

The fine-tuned Xception transfer learning model excelled in picture recognition. The model learned important patterns and characteristics from the training dataset with an accuracy of the test dataset of 97.60%. Figure 9 shows examples of the accuracy results of the Xception model's prediction in the test data. The confusion matrix of the Xception model is shown in Figure 10.

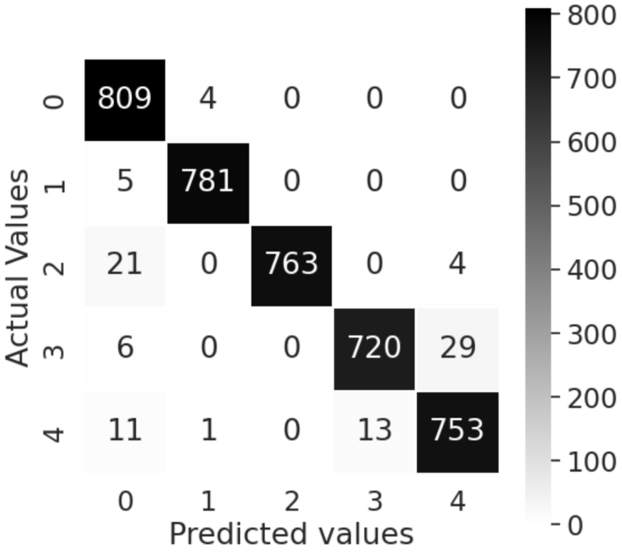


Figure 10. Xception model's confusion matrix for test data.

4.2. ResNet101

The performance of ResNet101 in image classification tasks is typically assessed using metrics such as top-1 accuracy, which measures the percentage of cases where the model's highest prediction matches the actual label, and top-5 accuracy, which measures the percentage of cases where the actual label is among the model's top 5 predictions.

When deploying the ResNet101 model, the top classification layer was set to false, to transfer the pre-trained model's knowledge to the task. This approach leads to faster training, reduces overfitting, and improves generalization.

The accuracy of trained ResNet101 for the test dataset was 97.30%. Figure 11 shows examples of the accuracy results of ResNet101 prediction on the test data. The ResNet101 confusion matrix is shown in Figure 12.

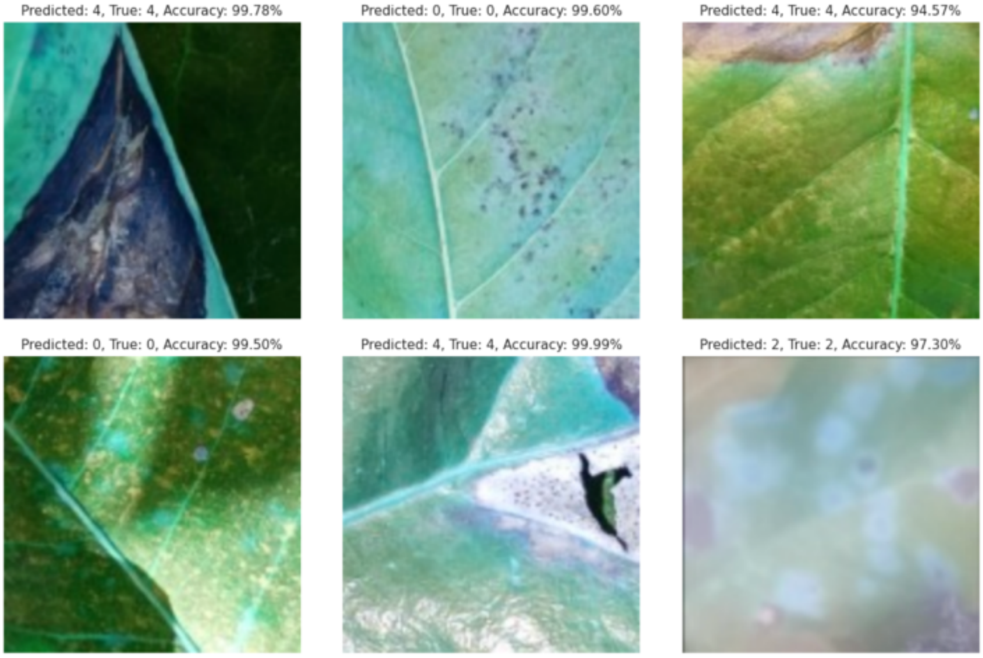
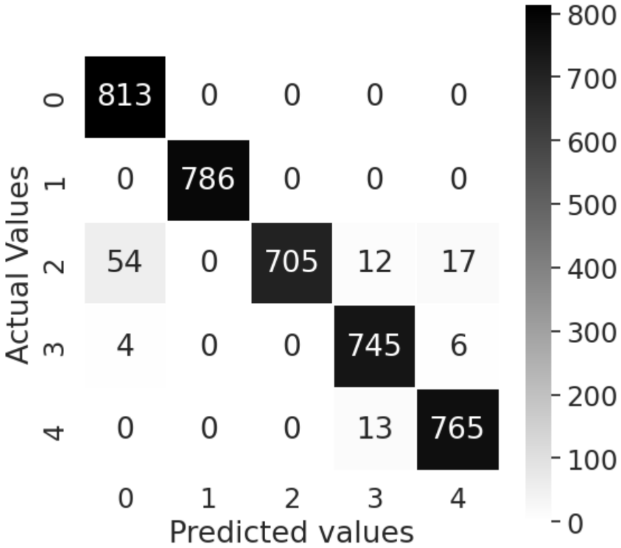

Predicted: 4, True: 4, Accuracy: 99.78%
Predicted: 0, True: 0, Accuracy: 99.60%
Predicted: 4, True: 4, Accuracy: 94.57%
Predicted: 0, True: 0, Accuracy: 99.50%
Predicted: 4, True: 4, Accuracy: 99.99%
Predicted: 2, True: 2, Accuracy: 97.30%

Figure 11. ResNet101 model's results for test data.



	0	1	2	3	4
0	813	0	0	0	0
1	0	786	0	0	0
2	54	0	705	12	17
3	4	0	0	745	6
4	0	0	0	13	765

Figure 12. ResNet101 model's confusion matrix for test data.

4.3. VGG16

VGG16 is a complex CNN structure. During the deployment of this model, the top layer was set to false, allowing us to take advantage of the feature extraction capabilities of VGG16 while adding custom layers to suit our task. We were able to add our own fully connected layers and use regularization techniques like drop and L2 regularization to the custom top layers, to avoid overfitting.

The accuracy of the trained VGG16 for the testing dataset was 99.89%. Figure 13 shows examples of the accuracy results of VGG16 prediction on the test data. The VGG16 confusion matrix is shown in Figure 14.

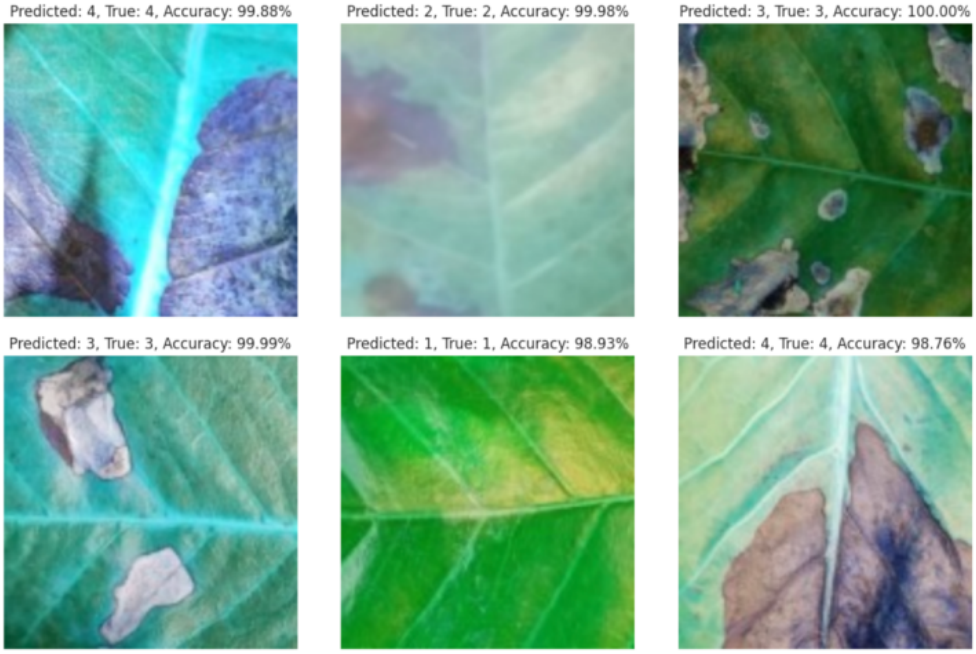

Predicted: 4, True: 4, Accuracy: 99.88%
Predicted: 2, True: 2, Accuracy: 99.98%
Predicted: 3, True: 3, Accuracy: 100.00%
Predicted: 3, True: 3, Accuracy: 99.99%
Predicted: 1, True: 1, Accuracy: 98.93%
Predicted: 4, True: 4, Accuracy: 98.76%

Figure 13. VGG16 model's results for test data.

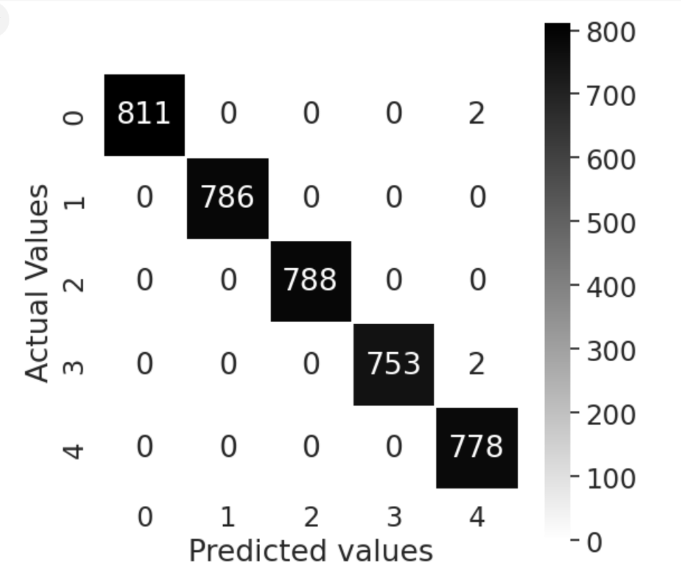


Figure 14. VGG16 model's confusion matrix for test data.

4.4. CoffNet

Typically, CNNs are trained and validated on labeled image datasets. The dataset is typically divided into three subsets: training, validation, and testing. During the training process, the validation set is utilized to assess the model's performance on data that it has not encountered before, as well as to keep an eye out for overfitting. CoffNet has the capacity to extract the characteristics of the picture and perform the image categorization concurrently, providing the advantage of promptly releasing notable photos [35].

Part of the preprocessing phase of the code had data augmentation to help the model train to increase data diversity, simulate real-world variability, and enhance model robustness. This was applied using rotation, horizontal flip, and zoom.

The three main components of CoffNet are the convolution layer, the pooling layer, and the fully connected layer. These layers collaborate to carry out categorization. The trained CoffNet model obtained a high accuracy of 97.88% on the test dataset (see Figure 15).

Figure 15 shows examples of the accuracy results of the CoffNet prediction on the test data. The CoffNet confusion matrix is shown in Figure 16. Figure 17 illustrates the training and validation curve for the proposed CoffNet. The graph shows that the training and validation loss tends to decline and reach convergence. The narrow gap between training and validation accuracy represents the consistent model. The graph shows that the CoffNet loss curve fluctuates within a narrow range after 10 epochs.

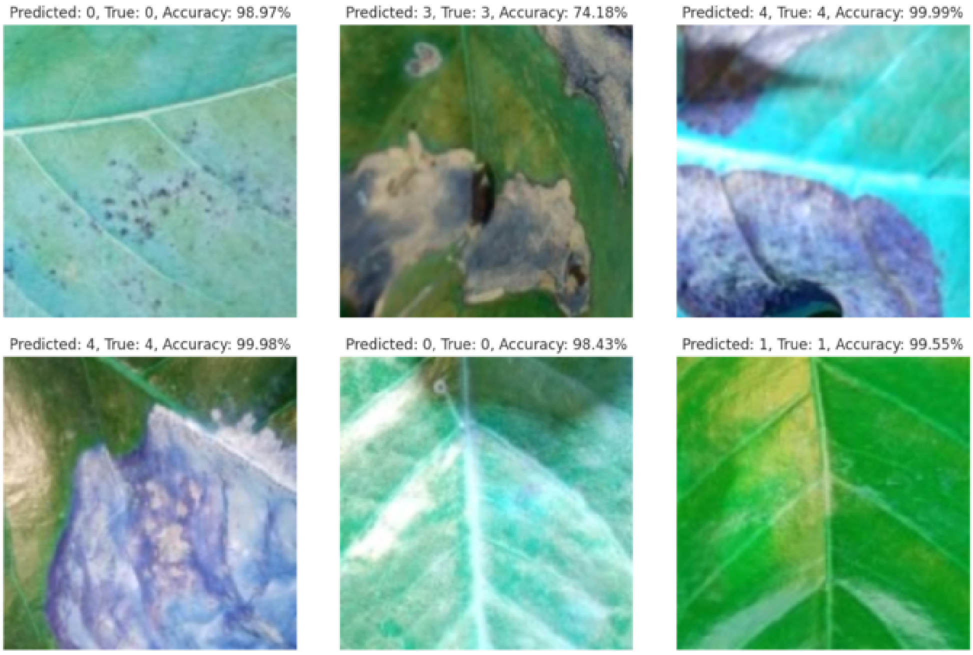

Predicted: 0, True: 0, Accuracy: 98.97%
Predicted: 3, True: 3, Accuracy: 74.18%
Predicted: 4, True: 4, Accuracy: 99.99%
Predicted: 4, True: 4, Accuracy: 99.98%
Predicted: 0, True: 0, Accuracy: 98.43%
Predicted: 1, True: 1, Accuracy: 99.55%

Figure 15. CoffNet model's results for test data.

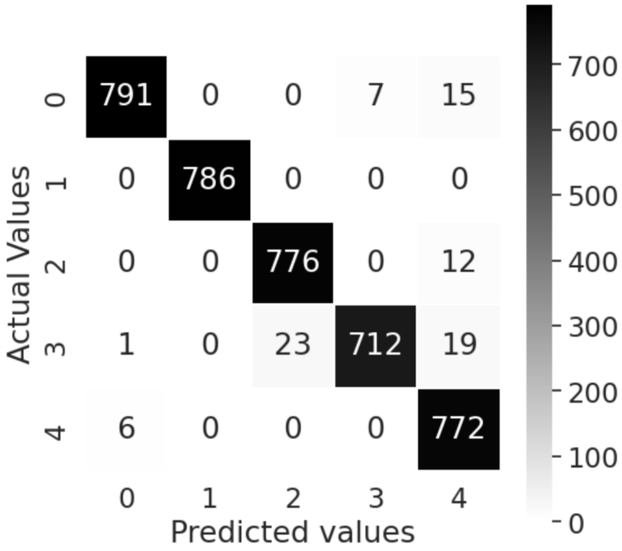


Figure 16. CoffNet model's confusion matrix for test data.

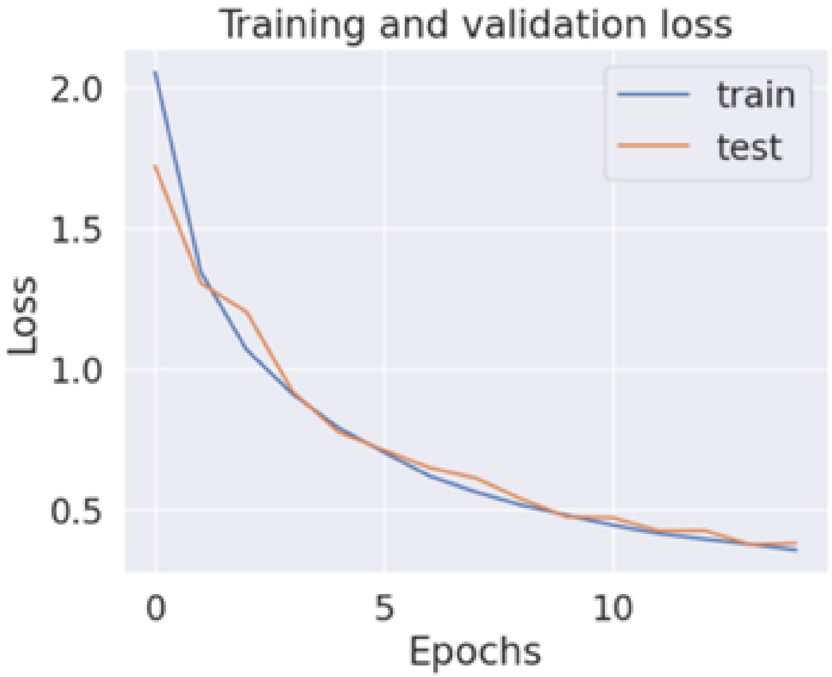


Figure 17. CoffNet training and validation loss analysis.

ROC Analysis

Figure 18 presents the performance of the four models using a receiver operating characteristic (ROC) curve. The ROC curve is a popular evaluation metric in machine learning that visually represents the effectiveness of a binary classifier by plotting the TPR against the FPR across various classification thresholds. The ROC curve serves as a graphical depiction of the trade-off between the FPR and the TPR, facilitating the comparison of the efficacy of diverse predictive models across varying operational thresholds. It illustrates the relationship between the TPR and the FPR at an array of threshold configurations.

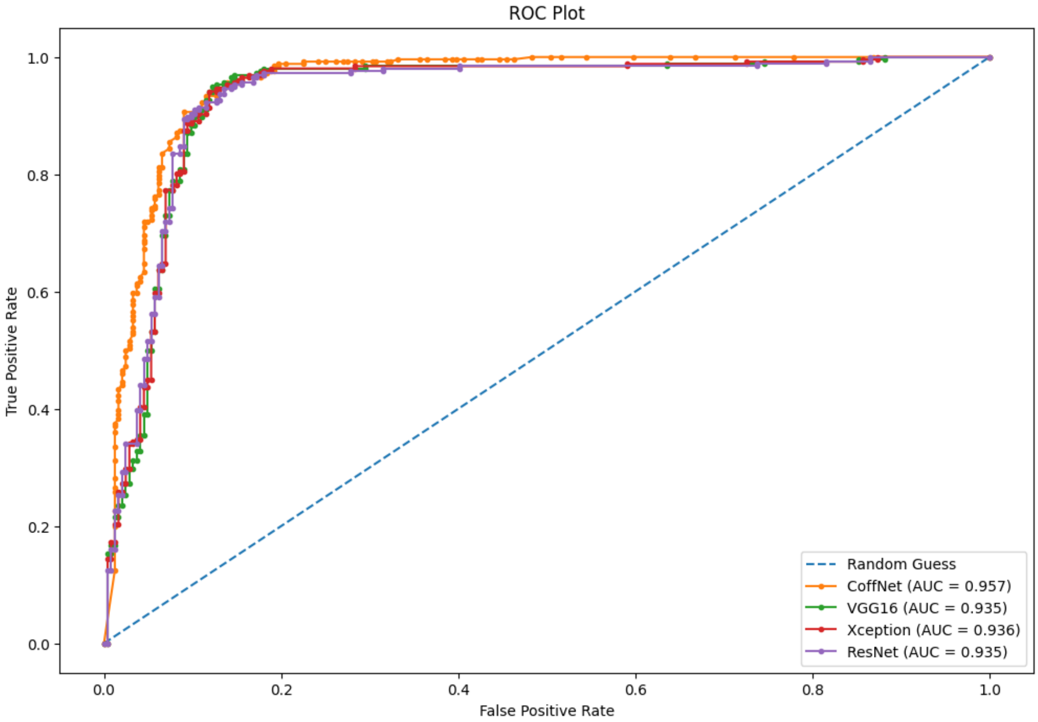


Figure 18. ROC curve analysis for all models.

CoffNet (AUC = 0.957): The ROC curve and AUC value highlight the superior performance of CoffNet, which aligns with the findings of the report. The higher AUC demonstrates that CoffNet excels at distinguishing between diseased and healthy leaves compared to other models. Considering other models (AUC = 0.935–0.936), pre-trained models such as VGG16, Xception, and ResNet also performed well, with AUC values around 0.94. However, the ROC curve confirms that CoffNet has a slight advantage in terms of precision and recall trade-offs, supporting the conclusion that domain-specific models like CoffNet outperform generalized architectures.

The ROC curve further validates the conclusion of the study that CoffNet maintains a better balance between the TPR (sensitivity) and FPR compared to other models. This reflects its improved ability to generalize effectively on the coffee leaf disease dataset.

4.5. Comparison with Other Methods

The timely identification of plant diseases is crucial, as they impose significant social, ecological, and economic burdens. This research utilized three established CNN models, namely Xception, ResNet101, and VGG16, with the newly proposed CNN model (CoffNet) demonstrating that the proposed model outperformed in terms of its rapid training and testing durations.

Table 6 shows the performance comparison of different models across the test dataset. It shows that VGG16 exhibited the highest level of accuracy, making it the most outstanding performance. ResNet101 consistently exhibited a long training time for each epoch across all four models.

Table 6. Performance comparison of all models based on test accuracy and average time per epoch (in seconds).

Model Used	Test Accuracy	Avg. Time per Epoch (s)
Xception	0.98	276 s
CoffNet	0.98	399 s
VGG16	0.99	809 s
ResNet101	0.97	1273 s

Performance metrics of different models are shown in Table 7. In this table, CoffNet demonstrates the best overall performance, achieving the highest precision, recall, and F1 score. These results suggest it is the most effective model among the four models. Xception and VGG16 have comparable performance, each scoring 0.98 across most metrics. They perform well but are slightly below CoffNet. ResNet101 has the lowest overall scores in this comparison, with all values around 0.97. Despite this, it remains a reliable model, although it is not as effective as the others.

Table 7. Performance comparison of all models across various evaluation metrics.

Model	Precision	Recall	F1 Score	Accuracy
Xception	0.98	0.98	0.98	0.98
CoffNet	0.99	0.99	0.99	0.98
VGG16	0.98	0.98	0.98	1.00
ResNet101	0.97	0.97	0.97	0.97

Performance comparison with existing studies in the literature is shown in Table 8. The table shows that our proposed model, CoffNet, has the best accuracy with 98%.

Table 8. Performance comparison of CoffNet with existing models.

Reference	Year	Accuracy (%)
[20]	2020	96
[22]	2021	92
[24]	2023	84
[25]	2023	95
[15]	2023	97
CoffNet	2024	98

In the context of deep learning, frames per second (fps) serve as an indicator of the rate at which a model can analyze data, predominantly image frames. This metric is especially crucial in real-time applications, such as the detection of coffee leaf diseases, where prompt decision-making is essential. A higher fps signifies that the model can process a greater number of images in a shorter duration, rendering it suitable for real-time implementation.

For example, in agriculture, if a drone is flying over a coffee plantation to monitor plant health, the drone's camera captures images at a specific rate (fps). The deep learning model needs to process those images quickly enough to provide real-time feedback, like detecting diseases as the drone moves over the field.

CoffNet with higher fps shows that it is better for real-time applications, and ResNet101 with a low fps model is suitable for applications that prioritize accuracy (see Table 9).

Table 9. Frames per second (fps) comparison of CoffeeNet and other models.

Model	Frames per Second (fps)
CoffNet	125.93
Xception	60.82
VGG16	47.99
ResNet101	37.55

4.6. Discussion

In general, we achieved satisfactory results with our experiments. However, some systematic errors were observed in the confusion matrices, primarily involving misclassifications between diseases with visually similar symptoms, including the following:

- Diseases like Phoma and Cercospora were occasionally confused due to their similar lesion patterns in certain lighting conditions.
- Augmented images with slight distortions or added noise may have reduced the clarity of fine-grained features, contributing to classification errors.
- Classes with fewer samples (e.g., Coffee Miner) showed slightly higher misclassification rates, likely due to less representative training data.

We also identified the following limitations that could impact model performance:

- Environmental Factors: variations in lighting, leaf orientation, and weather conditions in real-world agricultural settings could affect the models' accuracy.
- Class Representation: some disease classes had fewer samples, potentially impacting model generalization for underrepresented conditions.

The proposed identification framework showed good accuracy and higher fps when compared to other models (see Tables 7 and 9); however, the proposed method was tested using a specific dataset. The performance of our proposed method requires further evaluation with a larger number of subjects in real-world scenarios. There are several suggestions for future direction in this field, such as the following:

- Combine the proposed model with Internet of Things (IoT) devices, such as sensors and drones, for real-time, automated disease monitoring and reporting;
- Incorporate real-world datasets with diverse backgrounds and lighting conditions to enhance CoffNet's robustness in practical agricultural settings;
- Collaborate with coffee growers to create intuitive user interfaces and mobile applications to ensure that the system meets practical requirements in agricultural settings.

5. Conclusions

This study examined and assessed the performance of various cutting-edge CNN models, including Xception, ResNet101, VGG16, and CoffNet, with transfer learning, for the classification of coffee leaf diseases. In particular, this study highlighted the essential functions of the three primary CNN components: the convolution layer, the pooling layer, and the fully connected layer. These layers worked synergistically to extract pertinent features and ensure precise classification, contributing to the overall high accuracy of the trained CoffNet model on both the training and testing datasets. Our proposed model, CoffNet, demonstrated the best overall performance, achieving the highest precision, recall, and F1 score. The obtained results also show that our proposed CoffNet model outperforms some recent machine learning and deep learning techniques. In terms of processing speed, the CoffNet shows a notable processing speed of 125.93 frames per second (fps), making it ideal for real-time applications. Utilizing advanced technologies like drones and smartphones, this study could play a significant role in the early and automated detection of diseases affecting tomato plants. Using a diverse dataset of mixed images from multiple devices, our approach significantly reduces the workload of coffee growers, simplifies the disease detection process, and delivers superior results. In summary, this study advanced the field of diagnosis and treatment of coffee leaf disease by demonstrating the effectiveness of advanced deep learning algorithms and CNN architectures. The findings lay the groundwork for the development of practical and efficient systems that can assist coffee growers in disease management, promoting sustainable farming practices, and food security.

Author Contributions: Conceptualization, O.A. and B.P.; methodology, O.A. and B.P.; software, O.A.; investigation, O.A.; writing—original draft preparation, O.A. and B.P.; writing—review and editing, O.A. and B.P.; visualization, O.A.; supervision, B.P.; project administration, B.P.; funding acquisition, B.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data supporting this study are available at the following repositories: Krohling, Renato A.; Esgario, Guilherme J. M.; Ventura, José A. (2019), "BRACOL—A Brazilian Arabica Coffee Leaf images dataset to identification and quantification of coffee diseases and pests", Mendeley Data, V1, doi: 10.17632/yy2k5y8mxg.1

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

References

1. Pancsira, J. International Coffee Trade: A literature review. *J. Agric. Inform.* **2022**, *13*, 26–35. [CrossRef]
2. McCook, S.; Vandermeer, J. The big rust and the red queen: Long-term perspectives on coffee rust research. *Phytopathology* **2015**, *105*, 1164–1173. [CrossRef] [PubMed]
3. Shoaib, M.; Shah, B.; Ei-Sappagh, S.; Ali, A.; Ullah, A.; Alenezi, F.; Gechev, T.; Hussain, T.; Ali, F. An advanced deep learning models-based plant disease detection: A review of recent research. *Front. Plant Sci.* **2023**, *14*, 1158933.
4. Sharma, A.; Jain, A.; Gupta, P.; Chowdary, V. Machine learning applications for precision agriculture: A comprehensive review. *IEEE Access* **2020**, *9*, 4843–4873. [CrossRef]

5. Novtahaning, D.; Shah, H.A.; Kang, J.M. Deep learning ensemble-based automated and high-performing recognition of coffee leaf disease. *Agriculture* **2022**, *12*, 1909. [CrossRef]
6. Kamilaris, A.; Prenafeta-Boldú, F.X. Deep learning in agriculture: A survey. *Comput. Electron. Agric.* **2018**, *147*, 70–90. [CrossRef]
7. Mohanty, S.P.; Hughes, D.P.; Salathé, M. Using deep learning for image-based plant disease detection. *Front. Plant Sci.* **2016**, *7*, 1419. [CrossRef]
8. Thakur, P.S.; Sheorey, T.; Ojha, A. VGG-ICNN: A Lightweight CNN model for crop disease identification. *Multimed. Tools Appl.* **2023**, *82*, 497–520. [CrossRef]
9. Bansal, P.; Kumar, R.; Kumar, S. Disease detection in apple leaves using deep convolutional neural network. *Agriculture* **2021**, *11*, 617. [CrossRef]
10. Barbedo, J.G.A. Plant disease identification from individual lesions and spots using deep learning. *Biosyst. Eng.* **2019**, *180*, 96–107. [CrossRef]
11. González-Domínguez, E.; Monzó, C.; Vicent, A. New Trends in Disease and Pest Management: Challenges and Opportunities. *Agronomy* **2021**, *11*, 923. [CrossRef]
12. Picon, A.; Alvarez-Gila, A.; Seitz, M.; Ortiz-Barredo, A.; Echazarra, J.; Johannes, A. Deep convolutional neural networks for mobile capture device-based crop disease classification in the wild. *Comput. Electron. Agric.* **2019**, *161*, 280–290. [CrossRef]
13. Brahimi, M.; Arsenovic, M.; Laraba, S.; Sladojevic, S.; Boukhalfa, K.; Moussaoui, A. Deep learning for plant diseases: Detection and saliency map visualisation. In *Human and Machine Learning: Visible, Explainable, Trustworthy and Transparent*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 93–117.
14. Barbedo, J.G. Factors influencing the use of deep learning for plant disease recognition. *Biosyst. Eng.* **2018**, *172*, 84–91. [CrossRef]
15. Rodriguez-Gallo, Y.; Escobar-Benitez, B.; Rodriguez-Lainez, J. Robust coffee rust detection using uav-based aerial rgb imagery. *AgriEngineering* **2023**, *5*, 1415–1431. [CrossRef]
16. Mengistu, A.D.; Alemayehu, D.M.; Mengistu, S.G. Ethiopian Coffee Plant Diseases Recognition Based on Imaging and Machine Learning Techniques. *Int. J. Database Theory Appl.* **2016**, *9*, 79–88. [CrossRef]
17. Marcos, A.P.; Rodovalho, N.L.S.; Backes, A.R. Coffee leaf rust detection using genetic algorithm. In Proceedings of the 2019 XV Workshop de Visao Computacional (WVC), São Bernardo do Campo, Brazil, 9–11 September 2019; pp. 16–20.
18. Manso, G.L.; Knidel, H.; Krohling, R.A.; Ventura, J.A. A smartphone application to detection and classification of coffee leaf miner and coffee leaf rust. *arXiv* **2019**, arXiv:1904.00742.
19. Javierto, D.P.P.; Martin, J.D.Z.; Villaverde, J.F. Robusta Coffee Leaf Detection based on YOLOv3-MobileNetv2 model. In Proceedings of the 2021 IEEE 13th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment, and Management (HNICEM), Manila, Philippines, 28–30 November 2021; pp. 1–6.
20. Esgario, J.G.; Krohling, R.A.; Ventura, J.A. Deep learning for classification and severity estimation of coffee leaf biotic stress. *Comput. Electron. Agric.* **2020**, *169*, 105162. [CrossRef]
21. Velásquez, D.; Sánchez, A.; Sarmiento, S.; Toro, M.; Maiza, M.; Sierra, B. A method for detecting coffee leaf rust through wireless sensor networks, remote sensing, and deep learning: Case study of the Caturra variety in Colombia. *Appl. Sci.* **2020**, *10*, 697. [CrossRef]
22. Marin, D.B.; Santana, L.S.; Barbosa, B.D.S.; Barata, R.A.P.; Osco, L.P.; Ramos, A.P.M.; Guimarães, P.H.S. Detecting coffee leaf rust with UAV-based vegetation indices and decision tree machine learning models. *Comput. Electron. Agric.* **2021**, *190*, 106476. [CrossRef]
23. Yebasse, M.; Shimelis, B.; Warku, H.; Ko, J.; Cheoi, K.J. Coffee disease visualization and classification. *Plants* **2021**, *10*, 1257. [CrossRef]
24. Faisal, M.; Leu, J.S.; Darmawan, J.T. Model selection of hybrid feature fusion for coffee leaf disease classification. *IEEE Access* **2023**, *11*, 62281–62291. [CrossRef]
25. Chavarro, A.F.; Renza, D.; Ballesteros, D.M. Influence of hyperparameters in deep learning models for coffee rust detection. *Appl. Sci.* **2023**, *13*, 4565. [CrossRef]
26. Vilela, E.F.; Silva, C.A.d.; Botti, J.M.C.; Martins, E.F.; Santana, C.C.; Marin, D.B.; Freitas, A.R.d.J.; Jaramillo-Giraldo, C.; Lopes, I.P.d.C.; Corrêdo, L.d.P.; et al. Detection of Coffee Leaf Miner Using RGB Aerial Imagery and Machine Learning. *AgriEngineering* **2024**, *6*, 3174–3186. [CrossRef]
27. Krohling, R.A.; Esgario, J.; Ventura, J.A. BRACOL—A Brazilian Arabica Coffee Leaf images dataset to identification and quantification of coffee diseases and pests. *Mendeley Data* **2019**, *1*.
28. Shorten, C.; Khoshgoftaar, T.M. A survey on image data augmentation for deep learning. *J. Big Data* **2019**, *6*, 60. [CrossRef]
29. Chollet, F. Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1251–1258.
30. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
31. Kingma, D.P. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
32. Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2023.

33. Jouini, O.; Aoueileyine, M.O.E.; Sethom, K.; Yazidi, A. Wheat Leaf Disease Detection: A Lightweight Approach with Shallow CNN Based Feature Refinement. *AgriEngineering* **2024**, *6*, 2001–2022. [CrossRef]
34. Rainio, O.; Teuho, J.; Klén, R. Evaluation metrics and statistical tests for machine learning. *Sci. Rep.* **2024**, *14*, 6086. [CrossRef]
35. Jmour, N.; Zayen, S.; Abdelkrim, A. Convolutional neural networks for image classification. In Proceedings of the 2018 International Conference on Advanced Systems and Electric Technologies (IC_ASET), Hammamet, Tunisia, 22–25 March 2018; pp. 397–402.

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