J Inno Sci Eng, 2024, 8(2):160-178
https://doi.org/ 10.38088/jise.1455860

Research Article

Deep Learning-Based Damage Assessment in Cherry Leaves

Hazel Bozcu [1], Burakhan Cubukcu[1*]

[1] *Department of Computer Engineering, Bilecik Seyh Edebali University, 11230, Bilecik, Türkiye*

Abstract

This study aims to utilize deep learning methods for detecting diseases in cherry leaves to enhance agricultural productivity. While the detection of leaf diseases is currently performed by expert personnel, and the process can be quite time-consuming. Therefore, the primary objective of this study is to use deep learning-based disease detection applications to increase cherry production and enable early disease diagnosis. Additionally, the study investigates the impact of datasets on performance using two different datasets - one existing (PlantVillage Dataset) and one created for the study (Kozlu Dataset). Furthermore, the study examines the impact of hybrid architectures, combining convolutional neural networks (CNNs) and recurrent neural networks (RNNs), in addition to transfer learning methods and classical CNNs. On the PlantVillage dataset, AlexNet, VGG-16, MobileNet-V2, Inception-V3, and CNN models were compared. Due to the low performance of AlexNet and the long training time of VGG-16, MobileNet-V2, Inception-V3, CNN, and two different CNN+RNN models were compared on the Kozlu dataset. Based on the average results, the MobileNet-V2 model achieved the highest accuracy, approximately 99%, and the highest F1-score, also around 99%, in both datasets. The methods were observed to perform somewhat better on the PlantVillage dataset compared to the Kozlu dataset. Additionally, hybrid models (CNN+RNN) were found to achieve higher performance than the classical CNN model. These findings indicate promising outcomes for deep learning models in cherry leaf disease detection. In the study, the best results were obtained by the MobileNet-V2 model and the CNN + LSTM model, with an average accuracy of approximately 96%. In future studies, the reliability of this study can be increased by using more diverse datasets, and disease detection performance can be enhanced by using different deep learning methods, leading to reduced disease detection times.

Keywords: Hybrid AI models, Transfer learning, Damage assessment, Recurrent neural networks, Cherry Leaves

Cite this paper as:
Bozcu, H. and Cubukcu B. (2024). *Deep Learning-Based Damage Assessment in Cherry Leaves.* Journal of Innovative Science and Engineering. 8(2):160-178

*Corresponding author: Burakhan Cubukcu
E-mail: burakhcubukcu@gmail.com

Received Date: 20/03/2024
Accepted Date: 15/06/2024

1. Introduction

With the increasing global food demand, improving agricultural production and increasing efficiency have become critical issues. Detecting diseases in plant leaves and fruits is important for enhancing this efficiency. Agricultural workers typically observe leaf and fruit diseases with the naked eye, a method that is time-consuming and requires experienced personnel. Moreover, disease detection becomes even more challenging on large farms, where the process is further complicated and prolonged. To address these challenges, deep learning-based disease detection applications have been introduced in agriculture [1].

Moreover, damage to plant leaves can prevent plants from obtaining nutrients through photosynthesis, leading to their death [2]. It is known that unhealthy cherry tree leaves result in undersized fruits. Additionally, cherry fruits from trees with fallen or diseased leaves tend to wrinkle or have lower quality before ripening. Many diseases in cherry leaves can reduce fruit yield. For example, in the advanced stage of leaf spot disease, the plant's leaves may fall off, leaving only the fruit on the cherry tree. As a result, these fruits are often of poor taste, underdeveloped, and of low quality. Another example is powdery mildew disease, which is described as a powdery appearance seen throughout the green areas of the plant. Moreover, cherry leaf powdery mildew disease is usually observed in plants in high humidity areas. The disease, which starts from the upper parts of the plant's leaf and spreads, is a disease that surrounds the leaf by turning white like flour, leading to leaf and even fruit dropping [3]. Thus, early detection of diseases in leaves is important to prevent the progression of these diseases.

One of the methods that can be used for early detection is deep learning, which has made significant progress in recent years in various areas such as handwriting recognition, autonomous vehicles, earthquake prediction, and classification. Similarly, using artificial intelligence methods, leaf diseases can be detected, and plants can be protected through early diagnosis. A review study on plant disease detection and classification using deep learning methods explains the importance of deep learning models in increasing accuracy in detecting plant diseases [4].

Furthermore, for example, in a study on maize disease classification, a model developed using Convolutional Neural Networks (CNNs) can detect three different diseases in maize leaves. One of the techniques used for recognizing plant leaf diseases with the help of computer vision is disease detection through extracting color features from images [5]. Consequently, convolutional neural networks enable the classification and recognition of images by extracting color features.

Additionally, in the literature, there are disease detections using convolutional neural networks on different plant species. A study found in the literature was conducted to increase the diversity in datasets regarding cherry leaf diseases. Studies were aimed to contribute by training convolutional neural networks both with and without transfer learning methods. According to a study conducted in India on rice leaves, they achieved a success rate of 58% with the VGG-16 model [6]. Moreover, in this study, both the VGG-16 model and MobileNet-V2, Inception-V3, Long Short-Term Memory (LSTM) method, and Bidirectional LSTM method were used, and the success rate was aimed to be increased. Specific data augmentation processes and color spaces were applied to increase the success rate. Various examples of these processes are also available in the literature.

In these examples, Support Vector Machines (SVMs) are one of the widely used machine learning algorithms [7]. Except this model, Extreme Learning Machines (ELM), and K-Nearest Neighbor (KNN) were used as classifiers. Additionally, deep learning (DL) models such as ResNet-50, GoogleNet, ResNet-101, and SqueezeNet were used. In many studies, transfer learning models such as AlexNet, GoogleNet, VGG-16, MobileNet, are observed to be used [4]. The datasets used also have an impact on success.

Proposed study mainly focuses on contributing to the literature by classifying common diseases such as potassium deficiency and powdery mildew, and healthy leaves, using deep learning methods. Additionally, the study investigates the impact of datasets on performance using two different datasets. Another novelty of the study is the investigation of the impact of hybrid models by using together CNN with Recurrent Neural Networks (RNN) on performance.

In the continuous parts of this study, which enabling to contribute to the literature with new examples and comparisons, methodology, results and discussion, and conclusion sections are given respectively.

2. Methods

In this study, disease diagnosis in cherry leaves was performed using different deep learning methods on two datasets. The first dataset used in this study is the open-source PlantVillage dataset, where AlexNet, VGG-16, Inception-V3, MobileNet-V2, and CNN models were applied to detect healthy and powdery mildew-infected cherry leaves. In the second dataset, named Kozlu and created for this study, healthy leaves, leaves with potassium deficiency, and leaves with powdery mildew were detected using Inception-V3, MobileNet-V2, CNN, CNN + LSTM, and CNN + BiLSTM models. All methods were executed on Google Colab using a T4 GPU with a 25 GB RAM capacity. This study developed a code using the Python 3 programming language and built on the Google Compute Engine. The following section provides details about the materials and methods used in the study.

2.1. Datasets

In this study, in addition to using the PlantVillage dataset, a new dataset was created from cherry leaves obtained from cherry orchards in Kozlu village, Eskişehir. To increase the diversity of the created dataset and contribute to the generalization ability of the models, images belonging to the powdery mildew class from PlantVillage [8] were added to our new dataset, which was created based on literature research.

2.1.1. PlantVillage Dataset

PlantVillage is an open-source dataset containing approximately 54,000 images of 14 different types and 38 separate categories of healthy and disease cases [8], [9]. To meet the need for a verified large dataset to build a reliable image classification system, a project called PlantVillage was launched, collecting thousands of plant images, including healthy and diseased ones [10]. Experiments were conducted on this dataset using only two classes: leaves with powdery mildew disease and healthy leaves. In the training class, there are 1866 healthy leaves and 1683 leaves with powdery mildew disease. In the validation class, there are 456 healthy leaves and 421 leaves labeled with powdery mildew. Samples of PlantVillage Dataset shown in Fig 1.

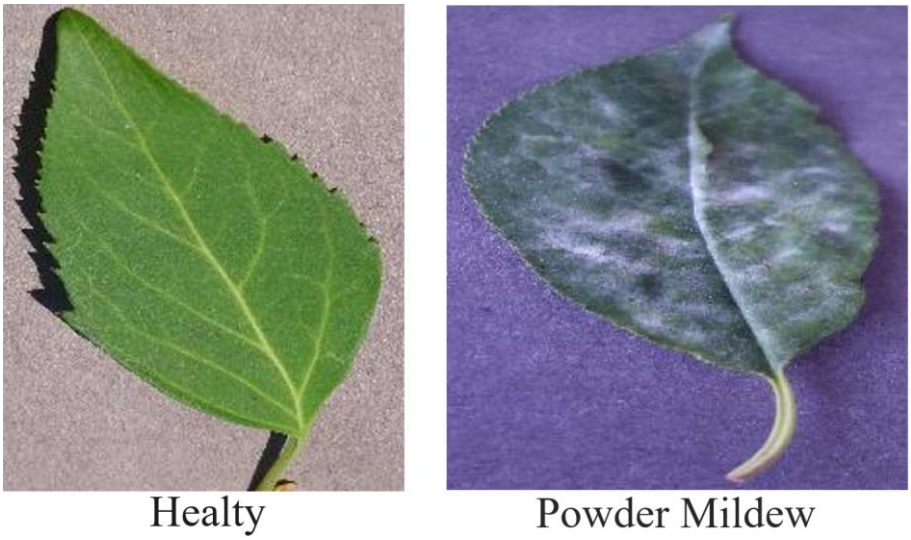

Figure 1: PlantVillage Dataset samples

2.1.2. Kozlu Dataset

The Kozlu dataset samples illustrated in Fig. 2. was derived from images collected from cherry orchards in Kozlu Village, Eskişehir and shared publicly on Kaggle [11]. This study photographed thousands of cherry leaves at a one-day interval. An agricultural engineer labelled the obtained images. To enhance the data diversity for the Kozlu dataset, consisting of three classes, powdery mildew disease was included from the PlantVillage dataset. In addition to leaves affected by powdery mildew disease, the dataset includes leaves with potassium deficiency and healthy leaves. The dataset comprises a total of 1438 images, with 478 images of powdery mildew-infected leaves, 480 images of leaves with potassium deficiency, and 480 images of healthy leaves. Data augmentation was applied to the training dataset in the studies, resulting in an augmented number of examples of 7898. A test dataset was separated at a 9% ratio.

Figure 2: Kozlu Dataset samples

To maintain consistency in the analysis and training processes, images from the PlantVillage and Kozlu datasets were resized to uniform pixel dimensions. This preprocessing step ensured that all images were adjusted to the same width and height, standardizing them to a consistent resolution. The key characteristics of these two datasets are provided in Table 1.

Table 1: The main features of Kozlu and PlantVillage dataset.

Feature	Kozlu Dataset	PlantVillage Dataset
Number of Samples	1438	4426
Healthy Samples	478	2322
Powdery Mildew Sample	480	2104
Potassium Deficiency Samples	480	0
Image Size	256x256 pixels	256x256 pixels
Bit Depth	32-bit	24-bit

2.2. Data Preparation and Preprocessing

Resizing visual data is a critical preprocessing step that involves resampling the original images to a specific size to enhance processing and analysis efficiency. In this study, images initially sized at 3472x4624 pixels were resized to 256x256 pixels. Expert support was enlisted to accurately label the images into healthy and diseased classes.

Following the labeling, data augmentation was applied. Parameters were defined using the ImageDataGenerator class, with rotation set to 20%, width and height shift to 0.2, shear to 0.2, and zoom to 0.2. These parameters were used to augment images for each class, creating a diversified dataset. Brown spots were emphasized on the augmented dataset using the highlight_damaged_areas function. This function is designed to reveal brown spots by masking a specific color range in the images. The process begins by converting the images to the YUV color space. Then, a defined color range is used to create a mask on the image. This masking process highlights only the areas with pixels in the specified color range, making the brown spots more prominent and allowing for better examination of the relevant areas. The function ensures consistent highlighting by being applied to all images in the training and test datasets.

Data augmentation was applied to each image in the training dataset using the datagen.flow function, which created a specified number of augmented images for each original image. Data augmentation enhances the model's ability to adapt to different conditions and variations, increasing its resilience to overfitting. This process aims to improve the model's performance on real-world data. The datasets created through these processes were used in this study.

2.3. Deep Learning Models

In this study, various deep-learning models were used for the best performance. The used models include AlexNet, VGG-16, Inception-V3, MobileNet-V2, CNN + LSTM, and CNN + BiLSTM. This section includes the features of the models used.

2.3.1. AlexNet

AlexNet is a convolutional neural network architecture that won the 2012 Large Scale Visual Recognition Challenge [12]. This competition aims to enhance accuracy in various visual recognition challenges by competing research teams against each other. AlexNet tested on ImageNet, a large labeled image collection containing approximately 15 million high-resolution labelled photographs. AlexNet's architecture consists of a total of eight weighted layers; five of them are convolutional layers, and three are fully connected layers. Rectified Linear Unit (ReLU) activation is used at the end of

each consecutive layer, and a softmax distribution is generated. Dropout is applied in the first two fully connected layers. Additionally, max-pooling is performed after the first, second convolutional layers, and the fifth convolutional layer [13].

In this study, when using AlexNet, the necessary Python libraries and modules are first imported, and preparation is made for data processing and model creation, and the dataset is divided into training, validation, and test data. Various data augmentation operations (rotation, shifting, brightness adjustment) are applied to increase the diversity of the training data. Various Keras callback functions (Early Stopping, Learning Rate Reduction, Model Checkpoint) are defined to monitor and optimize the training process.

2.3.2. VGG-16

This network architecture, produced similar to AlexNet principles, takes 224x224x3 images in RBG format as input. 3x3 size filters are used in the convolution layer. The VGG-16 architecture, introduced in 2014, consists of 16 layers in total, 13 of which are convolutional and 3 of which are full connection. The softmax classifier is used in the last layer of the architecture. Unlike the VGG-16 architecture, 3 more convolution layers were added to the VGG-19 architecture and it was designed as a total of 19 layers, 16 of which are convolution layers [14]. Fig. 3 presents the layers of the VGG architecture [15].

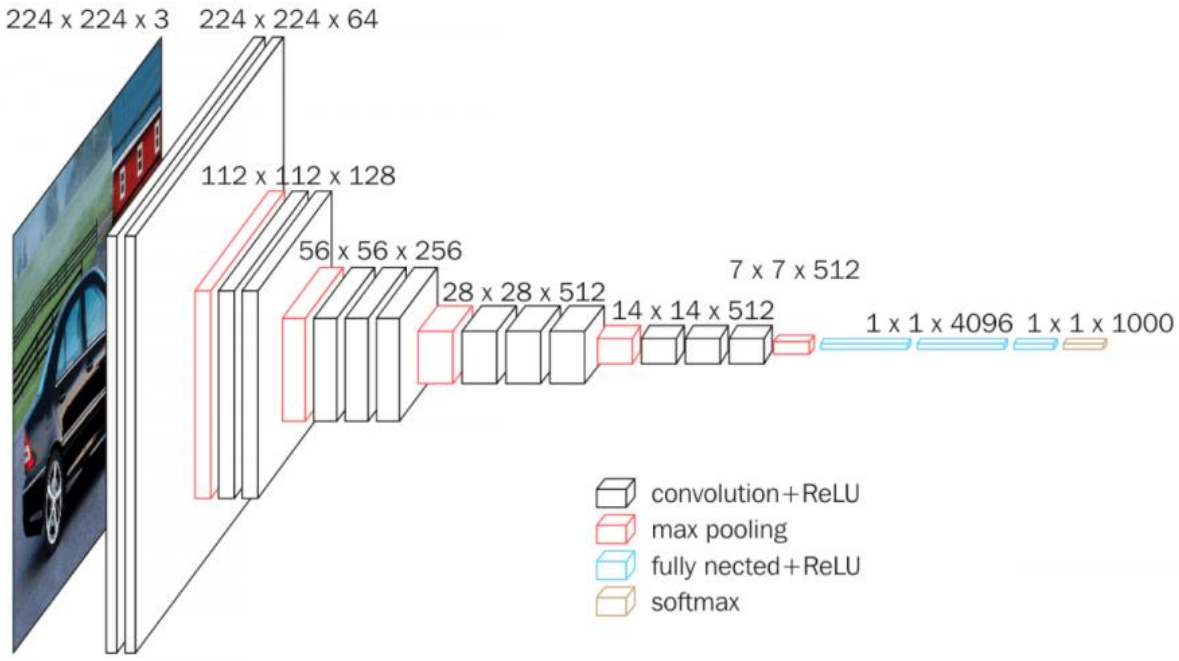

Figure 3: VGG architecture

In this study, the VGG-16 transfer learning model was customized. Initially, the VGG-16 model was imported with pre-trained weights on the ImageNet dataset, excluding the classification layer. Subsequently, to maintain the learned weights unchanged, the trainability of the VGG-16 model was disabled after determining the size of the input data. Finally, new layers, including flattening, fully connected, and classification layers, were then added to the architecture.

2.3.3. Inception-V3

Guan et al. employed the Inception-V3 network architecture in their experiments, a multi-layered model featuring three types of Inception modules [16]. These modules, known as Inception A, Inception B, and Inception C, excel at distinguishing data features and have shown promising results in plant classification studies [17]. The TensorFlow library, developed by Google, provides high flexibility for utilizing transfer learning methods, as noted by Kumar et al. opening avenues for deep learning-based research in image recognition and classification tasks [18]. The Inception-V3 model, illustrated in Fig. 4, boasts a versatile architecture comprising convolutional, maximum pooling, concatenation, dropout, fully connected, and average pooling layers [19]. This architecture has been observed to enhance feature extraction effectiveness and aid in interpreting image features [20].

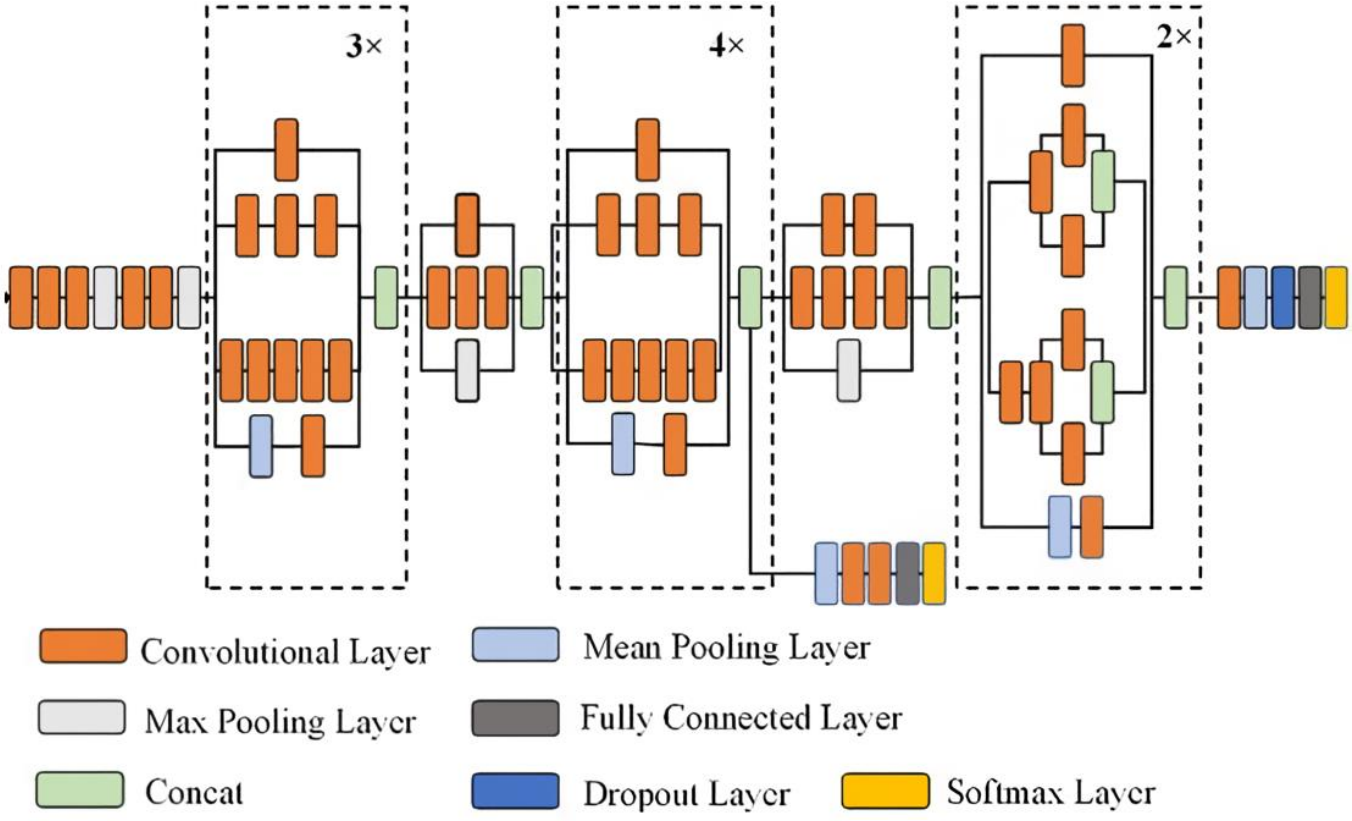

Figure 4: Inception-V3 architecture

In this study, the Inception-V3 architecture was enhanced with an average pooling layer and additional fully connected layers. To activate transfer learning, the layers of the Inception-V3 model were frozen, preventing them from learning. This ensured that the model retained its previously learned information. The Adam optimization method was employed, and categorical cross-entropy was chosen as the loss function. Following training, the model was utilized to predict the classes of the provided images.

2.3.4. MobileNet-V2

The initial version of the MobileNet architecture, introduced by Howard et al. performs a depthwise convolution in the first layer, followed by the addition of a 1x1 pointwise convolution layer in the first version of MobileNet [21]. The pointwise convolution layer combines a single output filter for each image channel. All operations are completed within a single block, and the MobileNet-V1 architecture comprises a total of 13 such blocks. In contrast, MobileNet-V2

incorporates a single block that includes an expansion layer (1x1), a depthwise convolution layer (3x3), and a projection layer (1x1) instead of two convolution layers. The MobileNet-V2 structure consists of a total of 17 blocks. In a study by Barman et al., the MobileNet-V2 architecture was utilized to differentiate citrus leaf diseases from healthy citrus leaves [22]. It was observed that MobileNet-V2 operates much faster than traditional CNN architectures [21].

In this study, the MobileNet-V2 model is extended with two fully connected layers containing 128 and 64 neurons, respectively, followed by a global average pooling layer. After each fully connected layer, a batch normalization layer and a dropout layer are added. The Adam optimizer is utilized, and "categorical_crossentropy" is selected as the loss function.

2.3.5. Classical CNN

In the designed CNN architecture, the input layer consists of a convolutional layer that produces 32 feature maps using 3x3 filters with ReLU activation. Following this, a MaxPooling2D layer is added to reduce image dimensions and prevent overfitting, with a dropout rate of 20%. Similarly, the second and third Convolutional Layers are added, using MaxPooling2D and Dropout layers to further reduce feature maps and prevent overfitting. After these Convolutional Layers, a Flatten layer is used to flatten the output, followed by fully connected layers containing 64 and 128 neurons, respectively, with ReLU activation functions. For the final classification layer, softmax activation is applied. During training, the model uses a learning rate of 0.001 with the Adam optimization algorithm, and the loss function selected is categorical cross-entropy.

2.3.6. CNN+LSTM

In the CNN+LSTM model illustrated in Fig. 5, similar to the CNN model, data is initially processed by a convolutional layer with 32 filters of size 3x3 and an input size of [224, 224, 3]. The CNN layers are responsible for extracting features from the images. Subsequently, a max-pooling layer reduces the dimensions, and a Dropout layer is added to mitigate overfitting. These layers are repeated three times in the same sequence, with the convolutional layers using 32, 64, and 128 filters respectively, and all convolutional layers employing the ReLU activation function. All max-pooling layers are set to 2x2, and the dropout rates are adjusted to 0.2, 0.2, and 0.4 respectively. After appropriate data resizing, the processed data is fed into a 128-cell LSTM layer. Finally, a fully connected layer with a softmax activation function is utilized at the end of the model to complete the classification process.

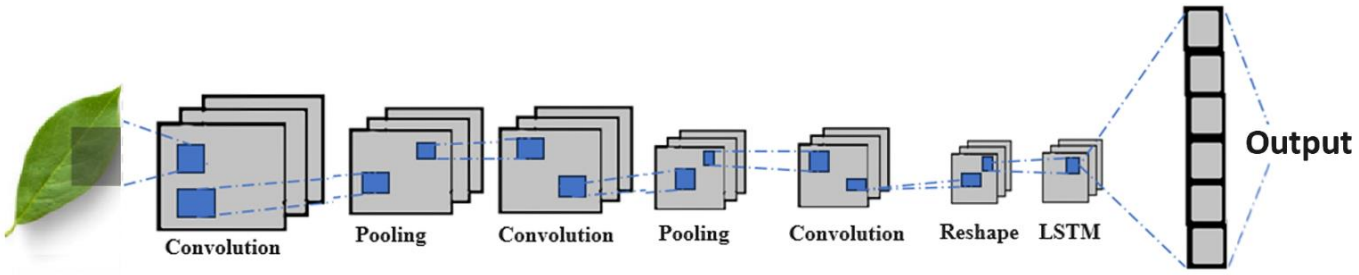

Figure 5: CNN + LSTM model architecture

2.3.7. CNN+BiLSTM

In this approach, as depicted in Fig. 6, the layers and parameters used in the CNN+LSTM model remain the same, except for the substitution of the LSTM layer with a BiLSTM layer. The BiLSTM layer in this method is configured with 128 cells.

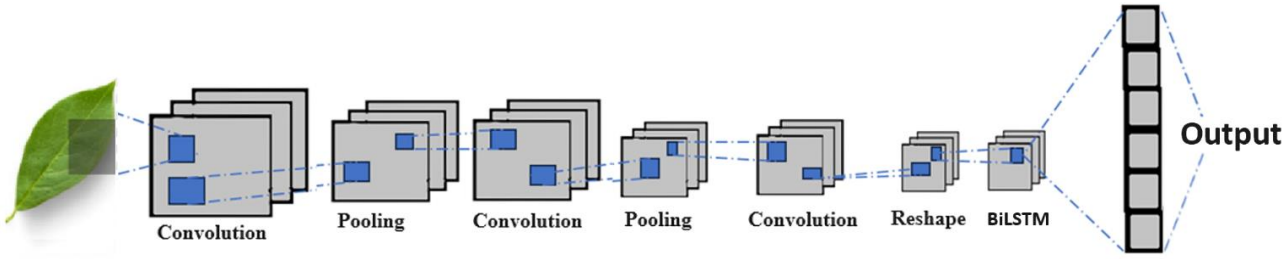

Figure 6: CNN +BiLSTM model architecture

Some characteristics of the deep learning methods used in this study are presented in Table 2 [23], [24], [25], [26].

Table 2: Some characteristics of the deep learning methods

Architecture	Year Introduced	Depth (Layers)	Parameters (approx.)	Top-1 ImageNet Accuracy	Key Features/Innovations
AlexNet	2012	8	60 M	62.5%	Deep convolutional neural network model consists of ReLU activations and densely connected layers.
VGG-16	2014	16	138 M	71.5%	A very deep architecture with small filter sizes and the use of max pooling.
Inception-V3	2015	100+	23.8 M	77.2%	Nested convolutional layers, parallel convolutions, reduced dimensionality
MobileNet-V2	2018	70+	3.47 M	74.7%	Lightweight architecture, linear activations, profiled filters
CNN + LSTM	-	3 convolutional and pooling layers, 1 LSTM layer	-	-	Integration of an LSTM layer on top of a convolutional neural network base
CNN + BiLSTM	-	3 convolutional and pooling layers, 1 BiLSTM layer	-	-	Integration of a bidirectional LSTM layer on top of a convolutional neural network base

2.4. Comparison Metrics

To compare the deep learning methods used, accuracy, precision, recall, and F1-score metrics which are widely used in literature were employed [27], [28], [29]. These metrics were calculated by subtracting the true positive, false negative, false positive, and true negative values obtained from the confusion matrix. An example of the appearance of these values in a confusion matrix is shown in Fig. 7 [30].

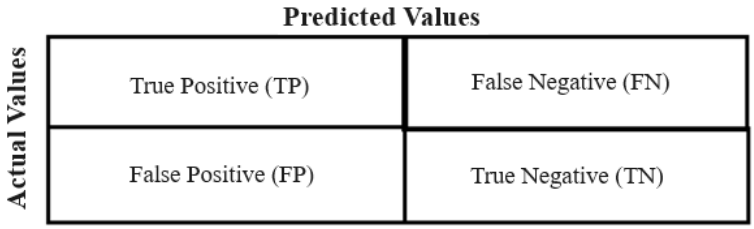

Figure 7: Confusion matrix

True Positive (TP): When the classifier correctly identifies a healthy leaf sample, it is referred to as true positive. For example, when the classifier correctly detects that there are no signs of disease on the leaf, it represents a true positive. The leaf actually has powdery mildew disease.

True Negative (TN): When the classifier correctly identifies a healthy leaf sample, it is referred to as true negative. For example, when the classifier correctly detects that there are no signs of disease on the leaf, it represents a true negative. The leaf does not actually have powdery mildew disease.

False Positive (FP): When a healthy leaf is incorrectly labeled as diseased by the classifier, it is referred to as false positive. For example, when a leaf is incorrectly labeled as diseased despite having no signs of disease, it creates a false positive scenario. An example is when a healthy leaf is predicted to have powdery mildew disease.

False Negative (FN): When a diseased leaf is incorrectly predicted as healthy by the classifier, it is referred to as false negative. For example, when a leaf has clear signs of disease but is incorrectly labeled as healthy by the classifier, it creates a false negative scenario [31]. An example is when a leaf is predicted not to have powdery mildew disease when it actually does.

Accuracy, precision, recall, and F1-score are calculated as shown in the equations 1-4 [27], [28], [29].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

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

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$\boldsymbol{F1\ Score = 2\ x\ \frac{Precision\ x\ Recall}{Precision + Recall}} \quad (4)$$

3. Results

This section presents the comparison of deep learning methods based on two datasets used in the study. The detection of healthy and powdery mildew-infected cherry leaves in the PlantVillage dataset, the models AlexNet, VGG-16, Inception-V3, MobileNet-V2, and CNN were employed. In the Kozlu dataset, healthy leaves, leaves with potassium deficiency and leaves with powdery mildew were detected using the Inception-V3, MobileNet-V2, CNN, CNN + LSTM, and CNN + BiLSTM models. The CNN + LSTM and CNN + BiLSTM methods could not be applied to the PlantVillage dataset due to insufficient RAM resources in the Google Colab environment. All deep learning methods were executed

with 30 epochs and 3 repetitions on both datasets. In the rest of this section, the results obtained for 2 different datasets are presented in separate sections.

3.1. PlantVillage Dataset Results

In the PlantVillage dataset, the goal was to determine whether the leaves were healthy or affected by powdery mildew by applying AlexNet, VGG-16, Inception-V3, MobileNet-V2, and Classical CNN models. The complexity matrices of the results are shown in Fig. 8. In the complexity matrix, 0 values indicate diseased leaves, and 1 values indicate healthy leaves. The red sections represent incorrect predictions by the models, while the green sections represent correct predictions.

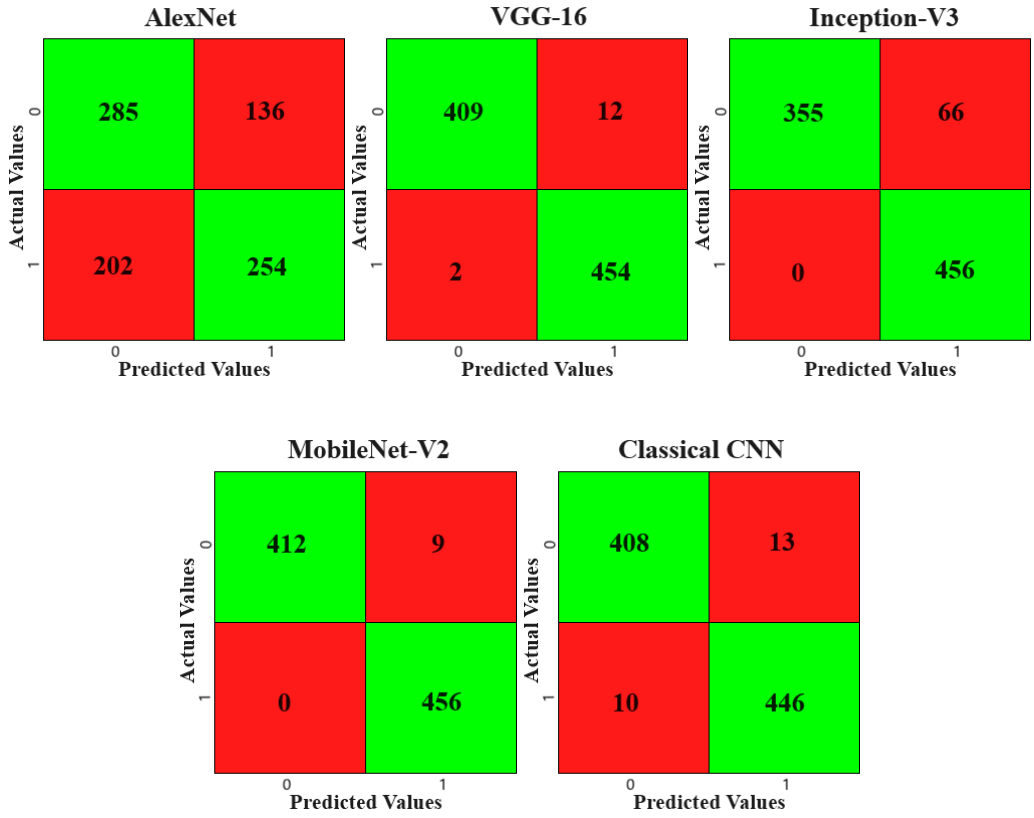

Figure 8: Confusion matrix of PlantVillage dataset results

As depicted in Fig. 8, AlexNet resulted in the worst performance with 338 incorrect predictions, while MobileNet-V2 achieved the best outcome with only 9 incorrect predictions. Both Inception-V3 and MobileNet-V2 models successfully predicted all diseased leaves correctly. Furthermore, the Classical CNN model made more correct predictions than 2 transfer learning models but fewer correct predictions than the other 2 models. The accuracy, F1-scores, precision, and recall results for the models based on 3 repetitions are shown in Table 3 and graphical representation of the mean results for the PlantVillage dataset illustrated in Fig. 9.

Table 3: The results of deep learning models on the PlantVillage dataset.

		Accuracy	F1-scores	Precision	Recall
AlexNet	Worst	0.5986	0.5882	0.6193	0.5986
	Mean	0.6065	0.6008	0.6194	0.6065
	Best	0.6145	0.6135	0.6195	0.6145
VGG-16	Worst	0.9817	0.9817	0.9819	0.9817
	Mean	0.9828	0.9828	0.9830	0.9828
	Best	0.9840	0.9840	0.9842	0.9840
MobileNet-V2	Worst	0.9814	0.9874	0.9876	0.9875
	Mean	0.9897	0.9917	0.9920	0.9918
	Best	0.9932	0.9932	0.9932	0.9932
Inception-V3	Worst	0.9133	0.9124	0.9257	0.9133
	Mean	0.9190	0.9182	0.9290	0.919
	Best	0.9247	0.9241	0.9343	0.9247
Classical CNN	Worst	0.9738	0.9700	0.9700	0.9700
	Mean	0.9738	0.9700	0.9700	0.9700
	Best	0.9738	0.9700	0.9700	0.9700

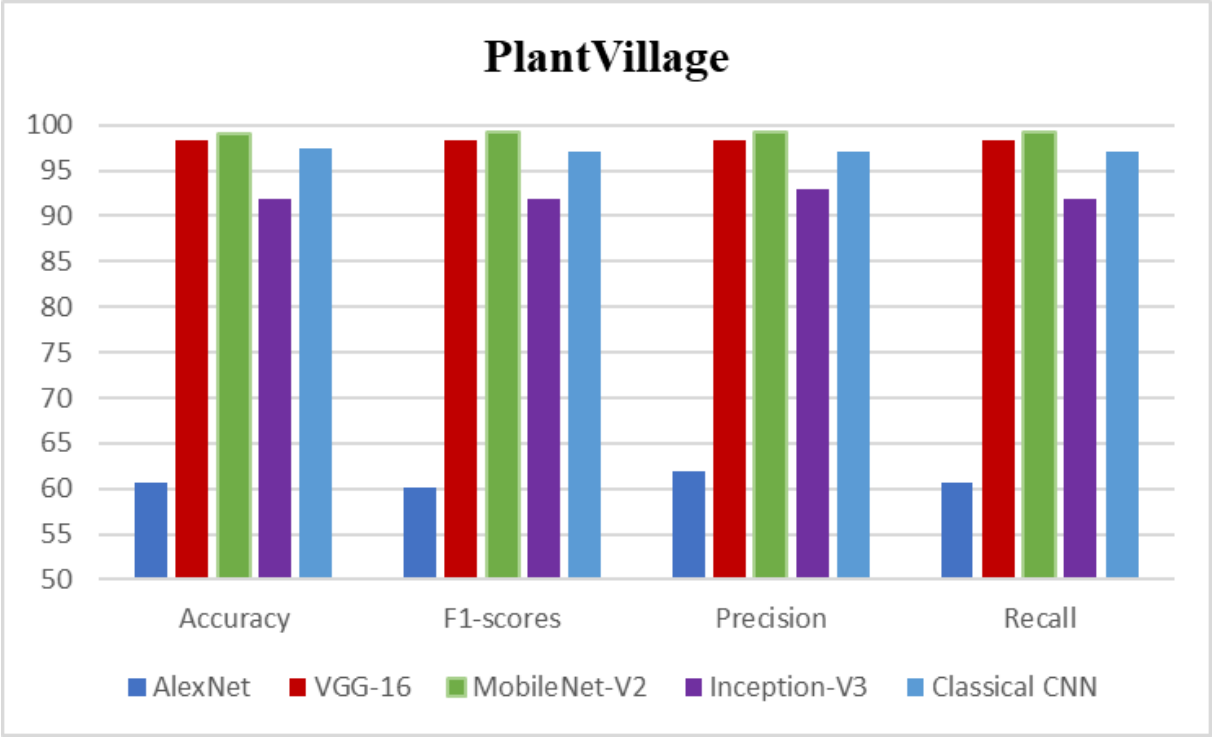

Figure 9: Graphical representation of the mean results for the PlantVillage dataset

Upon reviewing the results of the three repetitions in Table 3 for the PlantVillage dataset, it is observed that the MobileNet-V2 model exhibits the best performance with an average F1-score of 0.9817. In contrast, the AlexNet model performs the poorest with an average F1-score of 0.6008. The Inception-V3 model stands out with lower accuracy and higher loss values compared to MobileNet-V2. The Classical CNN model demonstrates a stable performance with generally high accuracy and low loss values. These findings highlight the superiority of the MobileNet-V2 model and the consistent performance of the Classical CNN model. However, it is evident that the performance of the Inception-V3 model lags behind the other models.

3.2. Kozlu Dataset Results

Based on the results obtained from the PlantVillage dataset, AlexNet was not considered for evaluation on the Kozlu dataset due to its low accuracy rate. Similarly, the VGG-16 model was not compared on the Kozlu dataset due to its lengthy training time. Instead, to examine the impact of CNN+RNN hybrid models on performance, CNN+LSTM and CNN+BiLSTM models were tested on the Kozlu dataset. The complexity matrices for the results of this study are shown in Fig. 10.

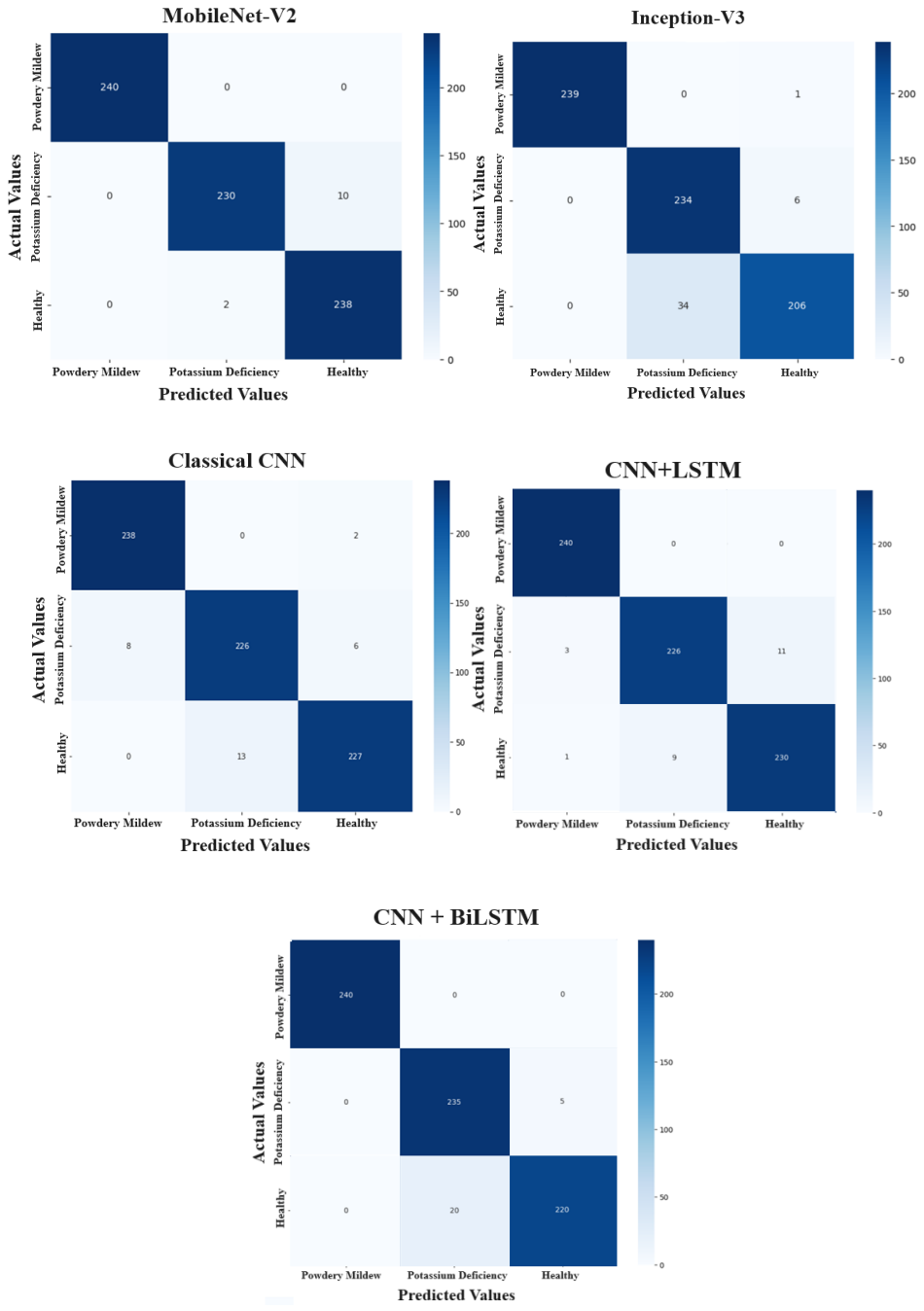

Figure 10: Confusion matrix of Kozlu dataset results

As can be understood from the complexity matrices, all five models have achieved promising results. Overall, the models predict powdery mildew disease with great success, but they also occasionally confuse leaves with potassium deficiency with healthy leaves. The accuracy, F1-scores, precision, and recall results for the models based on 3 repetitions are shown in Table 4 and graphical representation of the mean results for the Kozlu dataset illustrated in Fig.11.

Table 4: The results of deep learning models on the Kozlu dataset.

		Accuracy	F1-scores	Precision	Recall
MobileNet-V2	Worst	0.9833	0.9833	0.9833	0.9833
	Mean	0.9868	0.9893	0.9893	0.9867
	Best	0.9903	0.9933	0.9933	0.9900
Inception-V3	Worst	0.9431	0.9400	0.9433	0.9433
	Mean	0.9431	0.9417	0.9450	0.9433
	Best	0.9431	0.9433	0.9466	0.9433
Classical CNN	Worst	0.9597	0.9566	0.9633	0.9600
	Mean	0.9597	0.9583	0.9633	0.9600
	Best	0.9597	0.9600	0.9633	0.9600
CNN + LSTM	Worst	0.9625	0.9600	0.9633	0.9633
	Mean	0.9646	0.9634	0.9633	0.9650
	Best	0.9667	0.9667	0.9633	0.9667
CNN + BiLSTM	Worst	0.9611	0.9600	0.9633	0.9600
	Mean	0.9632	0.9633	0.9650	0.9633
	Best	0.9653	0.9666	0.9666	0.9667

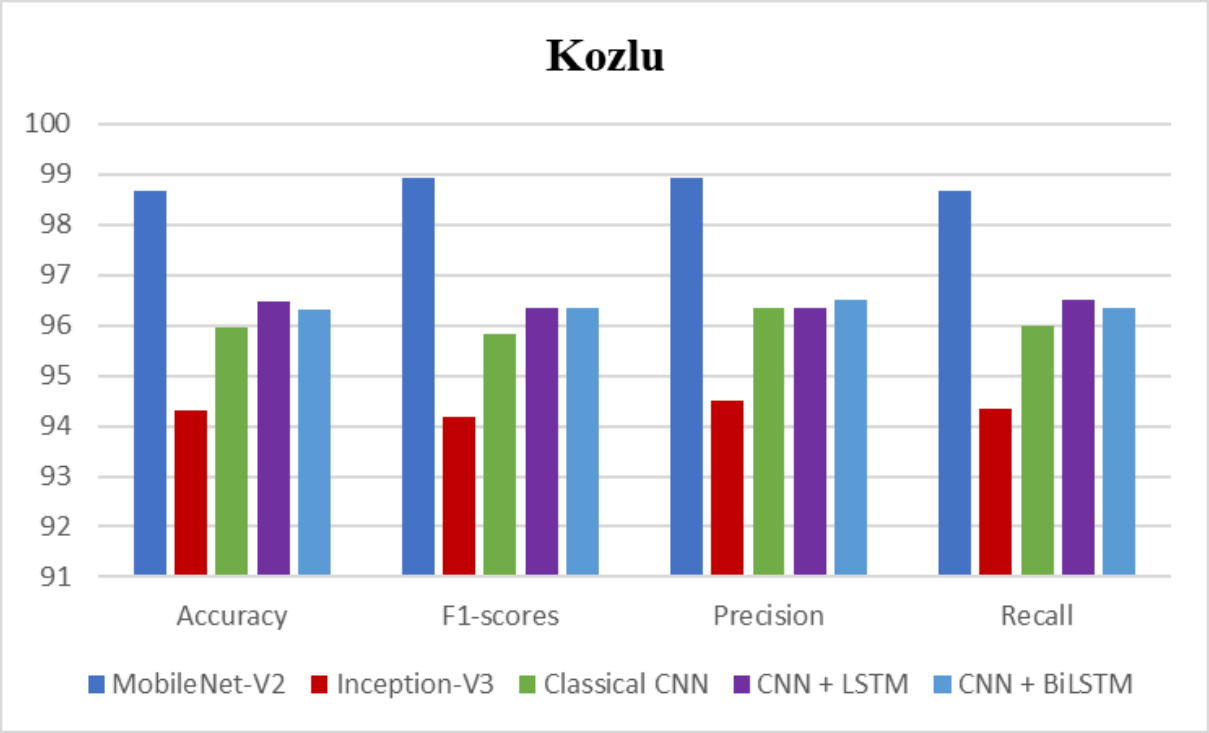

Figure 11: Graphical representation of the mean results for the Kozlu dataset

The results presented in Table 4 include information obtained by evaluating the performance of different deep learning models using various metrics. Among the models examined, MobileNet-V2 stands out as the top-performing model. In the best-case scenario, the MobileNet-V2 model achieved an accuracy of 99.03%, a loss of 0.0410, an F1-score of 99.33%, precision of 99.33%, and recall of 99.00%. The results obtained by other models are also promising. Even the lowest-performing model, Inception-V3, demonstrated successful performance with an accuracy of 94.31%, a loss of 0.1714, an F1-score of 94.00%, precision of 94.33%, and recall of 94.33%.

Furthermore, the CNN + LSTM and CNN + BiLSTM models, included in the dataset for evaluating hybrid models, outperformed the Classical CNN model. Considering that they achieved better performance compared to the transfer learning method Inception-V3 and were also trained faster, they are considered useful models.

4. Discussion

Based on the results obtained, the MobileNet-V2 transfer learning model has achieved the highest success in diagnosing diseases from cherry leaves. It has also been observed that adding RNN layers to CNN architectures enhances their performance. Similar observations were made for both datasets used in this study. However, the performance rates in the Kozlu dataset, which consists of real-life images, were lower compared to the PlantVillage dataset, which was produced in a laboratory environment. Since the Kozlu dataset was created specifically for this study, there is no opportunity for comparison with existing literature. Conversely, numerous studies have utilized the PlantVillage dataset for disease detection in agricultural products.

In a study aimed at detecting diseases in tomatoes using the PlantVillage dataset, the VGG-16 model achieved a 77.2% accuracy, MobileNet reached 63.75%, and the InceptionV3 model achieved 63.4% [32]. Another study, which compared its own methods with transfer learning models for tomato disease detection, reported that InceptionV3 achieved 94.58% accuracy, MobileNetV1 reached 82.7%, and MobileNet-V2 reached 92.1% accuracy [33]. In a study called "Attention Embedded Residual CNN for Disease Detection in Tomato Leaves," the proposed ResNet18-based model detected tomato diseases with 99.25% accuracy [34]. Similarly, in a study on detecting diseases in apples, AlexNet achieved 91.19% accuracy, ResNet-20 achieved 92.76%, and VGG16 achieved 96.32% accuracy [35]. In another study on detecting diseases in corn leaves, combining the features of EfficientNetBO and DenseNet121 models resulted in 98.56% accuracy [36]. In a survey of crop leaves, a proposed CNN-based model achieved 98.61% accuracy [37], while the study called "VGG-ICNN: A Lightweight CNN Model for Crop Disease Recognition" reported a 96.21% accuracy rate for detecting corn diseases [38]. The comparison of these studies using the PlantVillage dataset and this study are presented in Table 5.

Table 5: The comparison of some studies using the PlantVillage dataset

Study	Task	Dataset	Method	Accuracy
This study	Cherry leaf disease detection	PlantVillage (Cherry)	AlexNet, VGG-16, MobileNet-V2, Inception-V3, CNN	61.45%, 98.4%, 99.32%, 92.47%, 97.38%
Agarwal et al. [32]	Tomato leaf disease detection	PlantVillage (Tomato)	VGG-16, MobileNet, InceptionV3	77.2%, 63.75%, 63.4%
Thangaraj et al. [33]	Embedded Residual CNN for Disease Detection in Tomato Leaves	PlantVillage (Tomato)	InceptionV3, MobileNetV1, MobileNet-V2	94.58%, 82.7%, 92.1%
Karthik et al. [34]	Tomato leaf disease detection	PlantVillage (Tomato)	ResNet18-based proposed model	99.25%
Reddy and Rekha [35]	Apple leaf disease detection	PlantVillage (Apple)	AlexNet, ResNet-20, VGG16	91.19%, 92.76%, 96.32%
Amin et al. [36]	Corn leaf disease detection	PlantVillage (Corn)	Combining the features of EfficientNetBO and DenseNet121 models	98.56%
Gao et al. [37]	Crop disease detection	PlantVillage	CNN-based proposed model	98.61%
Thakur et al. [38]	Crop disease detection	PlantVillage	VGG-ICNN: A Lightweight CNN Model	96.21%

In most of the reviewed studies, transfer learning methods and proposed approaches achieve over 90% accuracy on various crops. The 99.32% accuracy of MobileNet-V2 in this study is particularly promising compared to the other studies reviewed. It is also observed that similar methods can produce different results across studies. This is expected, as each study's performance depends on various factors such as the number of samples used, parameters, and iterations. Additionally, data preprocessing steps have a direct impact on the studies' performance. For instance, in our study, highlighting the brown spots on the leaves was found to increase accuracy.

Beyond achieving high performance, another primary objective of our study was to create a real-world dataset and compare it with a dataset created in a laboratory environment. First, we created and publicly shared a comprehensive dataset. Given that the performance on our dataset was lower compared to the PlantVillage dataset, we believe real-world datasets are more reliable. This belief is reinforced by another study that compared a real-world dataset with the PlantVillage dataset and made a similar observation [39].

5. Conclusion

This study aims to detect diseases in cherry leaves to enable early diagnosis in agricultural production and increase productivity. The PlantVillage and Kozlu datasets, comprising images from cherry orchards in Eskişehir Kozlu Village, were utilized. In the PlantVillage dataset, AlexNet, VGG-16, Inception-V3, MobileNet-V2, and CNN models were employed, while in the Kozlu dataset, Inception-V3, MobileNet-V2, CNN, CNN + LSTM, and CNN + BiLSTM models were utilized to diagnose leaf diseases and compare the methods' performances.

Due to its lowest accuracy rate of 61%, the AlexNet model was not applied to the Kozlu dataset, while the VGG-16 model, despite achieving high success, was not implemented due to its lengthy training time. The Inception-V3 model demonstrated a high accuracy rate, with the MobileNet-V2 model achieving the highest accuracy rate, surpassing the other models. One of the proposed models in the study, the CNN model, accurately classified both powdery mildew disease and healthy leaves.

In the Kozlu dataset, Inception-V3, MobileNet-V2, CNN, LSTM, and BiLSTM models were examined. The MobileNet-V2 and CNN models achieved successful results with high accuracy rates. Particularly, the MobileNet-V2 model exhibited the highest performance in all classes. Although the CNN + LSTM and CNN + BiLSTM models also achieved good results, they demonstrated slightly lower performance than other models in some classes. It was observed that adding LSTM or BiLSTM layers to the end of the CNN layers enhanced performance.

A limitation of the study was the inability to use CNN + LSTM and CNN + BiLSTM models in the PlantVillage dataset due to their high RAM requirements. Another constraint was the limited number of gardens that could be used to create the Kozlu dataset.

In future studies, the reliability of the models can be enhanced by utilizing larger and more diverse datasets. Additionally, studies can be conducted with different deep learning architectures, and the results can be compared. More detailed analyses can be conducted on the training times and memory usage of the models, which can contribute to the development of more effective and faster models.

References

[1] G. F. S. Al Daban, " Plant Disease Detection Using SVM Classification," Altinbas University, İstanbul, 2019.

[2] Gıda Tarım ve Hayvancılık Bakanlığı, *Kiraz Vişne Hastalık ve Zararlıları ile Mücadele*, vol. 1. Ankara, 2016.

[3] Ş. Kurt, *Bitki Fungal Hastalıkları*, vol. 3. 2020.

[4] M. H. Saleem, J. Potgieter, and K. M. Arif, "Plant Disease Detection and Classification by Deep Learning," *Plants*, vol. 8, no. 11, 2019, doi: 10.3390/plants8110468.

[5] M. Sibiya and M. Sumbwanyambe, "A Computational Procedure for the Recognition and Classification of Maize Leaf Diseases Out of Healthy Leaves Using Convolutional Neural Networks," *AgriEngineering*, vol. 1, no. 1, pp. 119–131, 2019, doi: 10.3390/agriengineering1010009.

[6] P. Tejaswini, P. Singh, M. Ramchandani, Y. K. Rathore, and R. R. Janghel, "Rice Leaf Disease Classification Using Cnn," *IOP Conf Ser Earth Environ Sci*, vol. 1032, no. 1, p. 12017, Jun. 2022, doi: 10.1088/1755-1315/1032/1/012017.

[7] E. C. Seyrek, "The Use of Machine and Deep Learning on Hyperspectral Image Classification Applications," 2021. Accessed: Mar. 13, 2024. [Online]. Available: http://acikerisim.aku.edu.tr/xmlui/handle/11630/8546

[8] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Front Plant Sci*, vol. 7, 2016, doi: 10.3389/fpls.2016.01419.

[9] "PlantVillage Dataset." Accessed: Jun. 04, 2024. [Online]. Available: https://www.kaggle.com/datasets/mohitsingh1804/plantvillage

[10] F. Mohameth, C. Bingcai, K. A. Sada, F. Mohameth, C. Bingcai, and K. A. Sada, "Plant Disease Detection with Deep Learning and Feature Extraction Using Plant Village," *Journal of Computer and Communications*, vol. 8, no. 6, pp. 10–22, Jun. 2020, doi: 10.4236/JCC.2020.86002.

[11] H. Bozcu, "Kozlu Dataset." Accessed: Jun. 04, 2024. [Online]. Available: https://www.kaggle.com/datasets/hazelk26/kozlu-dataset

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. Burges, L. Bottou, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[13] I. Naeem Oleiwi Al-Mahdi, "CNN googlenet and alexnet architecture deep learning for diabetic retinopathy image processing and classification," İstanbul Gelişim Üniversitesi, İstanbul, 2023.

[14] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv preprint*, 2015.

[15] "VGG16 - Convolutional Network for Classification and Detection." Accessed: May 31, 2024. [Online]. Available: https://neurohive.io/en/popular-networks/vgg16/

[16] Q. Guan *et al.*, "Deep convolutional neural network Inception-v3 model for differential diagnosing of lymph node in cytological images: a pilot study," *Ann Transl Med*, vol. 7, no. 14, pp. 307–307, Jul. 2019, doi: 10.21037/ATM.2019.06.29.

[17] X. Xia, C. Xu, and B. Nan, "Inception-v3 for flower classification," in *2017 2nd International Conference on Image, Vision and Computing (ICIVC)*, 2017, pp. 783–787. doi: 10.1109/ICIVC.2017.7984661.

[18] S. Kumar, R. Ratan, and J. V. Desai, "Cotton Disease Detection Using TensorFlow Machine Learning Technique," *Advances in Multimedia*, vol. 2022, 2022, doi: 10.1155/2022/1812025.

[19] L. Ali, F. Alnajjar, H. Al Jassmi, M. Gochoo, W. Khan, and M. A. Serhani, "Performance Evaluation of Deep CNN-Based Crack Detection and Localization Techniques for Concrete Structures," *Sensors 2021, Vol. 21, Page 1688*, vol. 21, no. 5, p. 1688, Mar. 2021, doi: 10.3390/S21051688.

[20] S. H. Lee, C. S. Chan, S. J. Mayo, and P. Remagnino, "How deep learning extracts and learns leaf features for plant classification," *Pattern Recognit*, vol. 71, pp. 1–13, Nov. 2017, doi: 10.1016/J.PATCOG.2017.05.015.

[21] A. G. Howard *et al.*, "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications," *arXiv e-prints*, p. arXiv:1704.04861, Apr. 2017, doi: 10.48550/arXiv.1704.04861.

[22] U. Barman, R. D. Choudhury, D. Sahu, and G. G. Barman, "Comparison of convolution neural networks for smartphone image based real time classification of citrus leaf disease," *Comput Electron Agric*, vol. 177, p. 105661, Oct. 2020, doi: 10.1016/J.COMPAG.2020.105661.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. Burges, L. Bottou, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2012. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[24] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520. doi: 10.1109/CVPR.2018.00474.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *3rd International Conference on Learning Representations (ICLR 2015)*, Computational and Biological Learning Society, 2015, pp. 1–14.

[26] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Los Alamitos, CA, USA: IEEE Computer Society, Jun. 2016, pp. 2818–2826. doi: 10.1109/CVPR.2016.308.

[27] H. Parlak and B. Çubukçu, "Vgg-19 Based Multiclass Model For Ovarian Cancer Classification From Histopathologic Images," in *Mas 19th International European Conference On Mathematics, Engineering, Natural & Medical Sciences*, 2021, pp. 172–182.

[28] Z. B. G. Aydın and R. Şamlı, "A Comparison of Software Defect Prediction Metrics Using Data Mining Algorithms," *Journal of Innovative Science and Engineering*, vol. 4, no. 1, pp. 11–21, Jun. 2020, doi: 10.38088/JISE.693098.

[29] Ş. Doğru and V. Altuntaş, "Prediction of Cancer in DNA Sequences Using Unsupervised Learning Methods," *Journal of Innovative Science and Engineering*, vol. 7, no. 1, pp. 40–47, Jun. 2023, doi: 10.38088/JISE.1134816.

[30] T. Sulistyowati, P. PURWANTO, F. Alzami, and R. A. Pramunendar, "VGG16 Deep Learning Architecture Using Imbalance Data Methods For The Detection Of Apple Leaf Diseases," *Moneter: Jurnal Keuangan dan Perbankan*, vol. 11, no. 1, pp. 41–53, Jan. 2023, doi: 10.32832/MONETER.V11I1.57.

[31] A. S. Paymode and V. B. Malode, "Transfer Learning for Multi-Crop Leaf Disease Image Classification using Convolutional Neural Network VGG," *Artificial Intelligence in Agriculture*, vol. 6, pp. 23–33, Jan. 2022, doi: 10.1016/J.AIIA.2021.12.002.

[32] M. Agarwal, A. Singh, S. Arjaria, A. Sinha, and S. Gupta, "ToLeD: Tomato Leaf Disease Detection using Convolution Neural Network," *Procedia Comput Sci*, vol. 167, pp. 293–301, Jan. 2020, doi: 10.1016/J.PROCS.2020.03.225.

[33] R. Thangaraj, P. Pandiyan, S. Anandamurugan, and S. Rajendar, "A deep convolution neural network model based on feature concatenation approach for classification of tomato leaf disease," *Multimed Tools Appl*, vol. 83, no. 7, pp. 18803–18827, Feb. 2024, doi: 10.1007/S11042-023-16347-0/TABLES/2.

[34] R. Karthik, M. Hariharan, S. Anand, P. Mathikshara, A. Johnson, and R. Menaka, "Attention embedded residual CNN for disease detection in tomato leaves," *Appl Soft Comput*, vol. 86, p. 105933, Jan. 2020, doi: 10.1016/J.ASOC.2019.105933.

[35] T. Vijaykanth Reddy and K. Sashi Rekha, "Deep Leaf Disease Prediction Framework (DLDPF) with Transfer Learning for Automatic Leaf Disease Detection," *Proceedings - 5th International Conference on Computing Methodologies and Communication, ICCMC 2021*, pp. 1408–1415, Apr. 2021, doi: 10.1109/ICCMC51019.2021.9418245.

[36] H. Amin, A. Darwish, A. E. Hassanien, and M. Soliman, "End-to-End Deep Learning Model for Corn Leaf Disease Classification," *IEEE Access*, vol. 10, pp. 31103–31115, 2022, doi: 10.1109/ACCESS.2022.3159678.

[37] R. Gao, R. Wang, L. Feng, Q. Li, and H. Wu, "Dual-branch, efficient, channel attention-based crop disease identification," *Comput Electron Agric*, vol. 190, p. 106410, Nov. 2021, doi: 10.1016/J.COMPAG.2021.106410.

[38] P. S. Thakur, T. Sheorey, and A. Ojha, "VGG-ICNN: A Lightweight CNN model for crop disease identification," *Multimed Tools Appl*, vol. 82, no. 1, pp. 497–520, Jan. 2023, doi: 10.1007/S11042-022-13144-Z/TABLES/10.

[39] E. Li, L. Wang, Q. Xie, R. Gao, Z. Su, and Y. Li, "A novel deep learning method for maize disease identification based on small sample-size and complex background datasets," *Ecol Inform*, vol. 75, p. 102011, Jul. 2023, doi: 10.1016/J.ECOINF.2023.102011.