

Article

Multiclass Classification of Grape Diseases Using Deep Artificial Intelligence

Mohammad Fraiwan ^{1,*} , Esraa Faouri ¹  and Natheer Khasawneh ² ¹ Department of Computer Engineering, Jordan University of Science and Technology, Irbid 22110, Jordan² Department of Software Engineering, Jordan University of Science and Technology, Irbid 22110, Jordan

* Correspondence: mafraiwan@just.edu.jo

Abstract: Protecting agricultural crops is essential for preserving food sources. The health of plants plays a major role in impacting the yield of agricultural output, and their bad health could result in significant economic loss. This is especially important in small-scale and hobby-farming products such as fruits. Grapes are an important and widely cultivated plant, especially in the Mediterranean region, with an over USD 189 billion global market value. They are consumed as fruits and in other manufactured forms (e.g., drinks and sweet food products). However, much like other plants, grapes are prone to a wide range of diseases that require the application of immediate remedies. Misidentifying these diseases can result in poor disease control and great losses (i.e., 5–80% crop loss). Existing computer-based solutions may suffer from low accuracy, may require high overhead, and be poorly deployable and prone to changes in image quality. The work in this paper aims at utilizing a ubiquitous technology to help farmers in combatting plant diseases. Particularly, deep-learning artificial-intelligence image-based applications were used to classify three common grape diseases: black measles, black rot, and isariopsis leaf spot. In addition, a fourth healthy class was included. A dataset of 3639 grape leaf images (1383 black measles, 1180 black rot, 1076 isariopsis leaf spot, and 423 healthy) was used. These images were used to customize and retrain 11 convolutional network models to classify the four classes. Thorough performance evaluation revealed that it is possible to design pilot and commercial applications with accuracy that satisfies field requirements. The models achieved consistently high performance values (>99.1%).

Keywords: grape; artificial intelligence; black measles; black rot; isariopsis leaf spot; disease; deep learning; convolutional neural networks



check for updates

Citation: Fraiwan, M.; Faouri, E.; Khasawneh, N. Multiclass

Classification of Grape Diseases Using Deep Artificial Intelligence. *Agriculture* **2022**, *12*, 1542. <https://doi.org/10.3390/agriculture12101542>

Academic Editor: Jiangbo Li

Received: 4 August 2022

Accepted: 21 September 2022

Published: 24 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Protecting agricultural crops is essential for preserving food sources. The health of plants plays a major role in impacting the yield of the agricultural output, and could result in significant economic loss [1]. Plant diseases can be caused by viruses, bacteria, fungi, or even microscopic insects [2]. Extreme weather conditions and unseasonable temperatures can spur the growth and spread of harmful organisms, which may destroy entire crops or drastically reduce yield. Furthermore, global travel and globalization are increasing the risk of infection with transboundary diseases. The Food and Agriculture Organization (FAO) estimates that plant diseases cost the global economy about USD 220 billion in damage per year [3]. Thus, effective disease detection and control mechanisms are required to counter these factors [4].

Grapes are an important and widespread fruit that is consumed as a food or beverage. The total global production of fresh grapes (i.e., combined table grapes and wine grapes) was 77.13 million metric tons in 2019, with a market value of USD 189.19 billion [5]. However, they may be susceptible to several diseases that may affect their growth, production, and quality. Grape diseases can cause from 5% to 80% crop loss depending on the severity and spread of the disease [6]. Thus, they are a major production risk with profound economic ramifications. The early and correct detection and identification of a disease

is greatly important in reducing the disease progression and economic cost [7]. Grapes are prone to a number of diseases, including black measles, black rot, and isariopsis leaf spot. These diseases cause significant damage and may need plant pathologists to correctly diagnose them [8,9]. The next few paragraphs provide more details about these diseases.

Grape black measles (Esca) is one of the oldest known diseases of plants, caused by the fungi *Phaeomoniella Aleophilum* or *Chlamydo-spore*. The disease affects the grape plant and appears on its leaves in the form of irregular yellow and circular spots. It is contagious and can spread in grape fields, destroying vast portions of the crop. Therefore, it must be discovered early to reduce the rapid spread and the resulting damage [10].

Grape black rot is a severe disease that attacks the grape plant and appears clearly on its leaves. It is caused by the *Guignardia Bidwell* fungus. Black rot is a severe threat to grape production, with losses in the range of 5–80% [11]. The control of black rot requires the use of fungicides, which leads to an elevated financial cost of farming.

Grape isariopsis leaf spot is a rapidly spreading disease that appears on leaves in the form of pale red to brown lesions. It is caused by the *Pseudocercospora vitis* fungus. Special fungicides are used as a remedy to limit the spread of the disease, so it must be detected early [12].

Technological advances have powered many agricultural innovations. More specifically, artificial-intelligence (AI) and deep-learning algorithms can use plant images to drive farming control applications. Deep learning is based on neural networks that comprise a number of layers far greater than the input, output, and hidden layers. Convolutional neural networks (CNNs) are a type of deep-learning network that find features and relationships in images through a sequence of convolutional, pooling, and rectified linear unit (ReLU) layers that terminate in a fully connected layer. This layer aggregates the various features discovered by the former layers. CNNs are effective in discerning features regardless of many changes that can affect the input images [13]. In designing CNNs-based systems, researchers can build the network structure from scratch. Alternatively, existing reliable and well-established models can be reused, and their learned knowledge can be transferred to other applications in transfer learning. Deep transfer learning adapts existing models with partial or full retraining in a manner that fits the new application. It has the advantages of detecting generic features (e.g., colors, borders) with earlier layers, and customizing later layers for specific applications.

The related literature includes several studies in the multiclass classification of grape diseases; see Table 1. Huang et al. [14] considered one healthy class and four grape diseases: black rot, black measles, phylloxera, and leaf blight. Classification was performed using one custom baseline model called Vanilla CNN, composed of 7 layers, and transfer learning using modified VGG16, AlexNet, and MobileNet. Furthermore, a fifth ensemble model was developed by combining the predictions from the four aforementioned models. The authors reported an accuracy range of 77–100%. Thet et al. [15] used global average pooling to fine-tune the VGG16 model to perform six-way classification (one healthy + five diseases). Their dataset included the diseases of anthracnose, nutrient insufficiency, downy mildew, black measles, and isariopsis leaf spot. They reported 98.4% accuracy. Lauguico et al. [16] compared the performance of three pretrained models: AlexNet, GoogLeNet, and ResNet-18. The dataset was composed of healthy leaf images and three disease types (i.e., black rot, black measles, and isariopsis). In their work, the highest accuracy was achieved by the AlexNet model (i.e., 95.65%). Similarly, Ji et al. [17] aimed to classify the same set of diseases as that of Lauguico et al. They developed a CNN model called UnitedModel that works by combining features from the Inceptionv3 and ResNet50 models using global average pooling. They achieved an F1 score of 98.96%. Likewise, Lin et al. [18] designed a custom CNN called GrapeNet that contained a convolutional block attention module. This, they claim, has the effect of emphasizing disease features and suppressing irrelevant information. They compared its performance to that of nine other models: GoogLeNet, Vgg16, ResNet34, DenseNet121, MobileNetv2, MobileNetv3_large, ShuffleNetV2, ShuffleNetV1, and EfficientNetV2_s. GrapeNet achieved the highest accuracy value of 86.29%. Liu et al. [19] proposed a dense inception convolutional neural network that surpassed the

performance of ResNet-34 and GoogLeNet with 97.22% accuracy. Tang et al. [20] modified the design of ShuffleNetV1 using channelwise attention and achieved 99.14% accuracy. Andrushia et al. [21] used capsules to represent spatial disease information in the design of convolutional capsule network and reported an accuracy of 99.12% as a result. Hasan et al. designed a simple CNN that was able to achieve 91.37% accuracy. Goncharov et al. [22] collected a special dataset of grape leaf images representing healthy, black measles, black rot, and Chlorosis classes. They performed classification using four models (i.e., VGG19, Inceptionv3, ResNet50, and Xception) and reported a highest accuracy of 90%.

In spite of the great advances in deep learning, traditional machine-learning and classification techniques are still proposed in the literature. Waghmare et al. [23] used segmentation to extract the relevant parts of leaf images. After that, fractal features were extracted and fed to an SVM classifier, which achieved 96.6% accuracy. Jaisakthi et al. [24], and Ansari et al. [25] employed the same classifier as that of Waghmare et al. However, the former used global thresholding and semisupervised techniques to achieve 93% accuracy, and the latter used image enhancements and Haar wavelet transform to achieve 97% precision. Such nondeep neural-network methods are susceptible to image quality changes [26,27], require preprocessing steps that may introduce more errors or slower processing times, and do not achieve better performance than that of deep-learning methods.

Table 1. The state of the art in classifying grape diseases. Each study included one class for the healthy state.

Study	No. of Classes	Dataset	Approach
Huang et al. [14]	Five	5937 leaf images	Custom CNN, VGG16, AlexNet, MobileNet, and an ensemble.
Thet et al. [15]	Six	6000 leaf images	Fine-tuned VGG16.
Lauguico et al. [16]	Four	4062 leaf images	AlexNet, GoogLeNet, and ResNet-18.
Ji et al. [17]	Four	1619 leaf images	UnitedModel.
Lin et al. [18]	Seven	2850 leaf images	GrapeNet custom CNN.
Liu et al. [19]	Six	7669 leaf images	Dense inception convolutional neural network.
Tang et al. [20]	Four	4062 leaf images	Improved ShuffleNet V1.
Andrushia et al. [21]	Four	11,300 leaf images	Convolutional capsule network.
Goncharov et al. [22]	Four	3200 leaf images	VGG19, Inceptionv3, ResNet50, and Xception.
Waghmare et al. [23]	Three	450 leaf images	Fractal features + SVM.
Jaisakthi et al. [24]	Four	5675 leaf images	Segmentation + colour features + SVM.
Ansari et al. [25]	Two	400 leaf images	Segmentation + Haar wavelet transform + SVM
Hasan et al. [28]	Seven	1000 leaf images	Custom CNN.

The research landscape on the use of AI in agriculture and specifically in disease identification is ripe for more innovation and further confirmative studies. The work in this paper evaluates transfer learning using a wide range of CNN models to classify grape diseases. Instead of designing a CNN model from scratch, the work employs efficient well-known models that had undergone extensive evaluation to gain their place in the literature. Furthermore, the approach facilitates deployment and implementation by not requiring explicit feature extraction or elaborate image preprocessing. This work contributes the following:

- Using leaf images as input, CNN models are implemented to classify grape diseases. Three such diseases were considered in this study: black measles, black rot, and isariopsis leaf spot. In addition, a fourth healthy class was included.
- Using transfer learning, 11 CNN models were implemented to classify the input into one of the four classes.
- The performance of the 11 models was measured and compared from various angles of classification capabilities using a wide range of metrics. Moreover, the training and valuations times were recorded. The results show that wrapping such models in mobile and smartphone devices can aid farmers in quickly and correctly identifying diseases.

In the next section, the input and dataset are described in detail, more information is provided about the CNN models, the hyperparameters and computing environment are specified, and the measures of performance are defined. This is followed by the results and discussion in Section 3, and the conclusion is presented in Section 4.

2. Materials and Methods

2.1. Dataset

The dataset is composed of 3639 grape leaf images representing three common diseases (1383 black measles, 1180 black rot, and 1076 isariopsis leaf spot), in addition to 423 healthy leaf images [29]. The images contained a picture of one leaf only. All files were in .jpg extension and had a resolution of 256 by 256 pixels. All images were taken using a unified background and lighting. Samples of the dataset are shown in Figure 1.

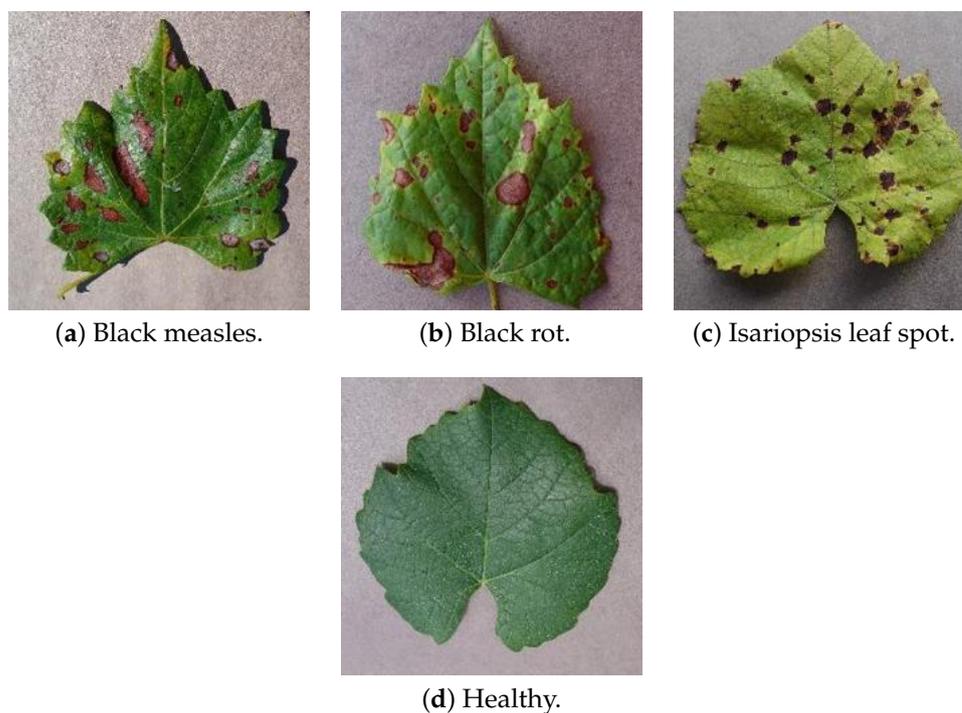


Figure 1. Examples from the dataset representing the four classes.

2.2. CNN Models

The approach used in this paper relies on the concept of transfer learning, which employs existing models including their network structure and parameter values. These models offer a trustworthy, robust, efficient, and well-established CNN design [30]. However, they are typically trained to recognize a wide range of objects (e.g., the ImageNet dataset includes 1000 types of generic objects [31]). Earlier network layers learn high-level features (e.g., object boundaries, color, etc), and later layers recognize object-specific features and combine the contributions of previous layers. Consequently, these pretrained models require repurposing to fit into the requirements of the specific application (e.g., recognizing certain diseases in an X-ray). This process includes resizing the input to the specific model requirements, replacing the final layers to match the required application output, and freezing the initial layers for faster retraining. Moreover, further layers, enhancements (e.g., pooling strategies), and fine-tuning can be conducted. Such an approach to solving image-based problems was extensively reported in the literature with great success in a wide-range of applications [32–35].

In this work, 11 CNN models were used and compared in their ability to classify grape leaf images into the aforementioned four classes. These models were: DarkNet-53 [36], DenseNet-201 [37], GoogLeNet [38], Inceptionv3 [39], MobileNetv2, ResNet-18,

ResNet-50, ResNet-101 [40], ShuffleNet [41], SqueezeNet [42], and Xception [43]. As types of CNNs, these models perform common operations (e.g., convolution). However, they differ greatly in their depth (i.e., layers between the input and output), width (i.e., the size of the input images, and dimensions of the filters and intermediate layers), the aggregation of features (e.g., the squeeze and suppress operations in SqueezeNet), internal connectivity (e.g., multipath design), and efficiency in calculating and updating the model parameters. All of the models were pretrained using the ImageNet database, which consists of one million images [31].

The models chosen here are representative of a wide range of design philosophies. The differences between some of these models can be as small as the optimization of some network parameters. Other models, however, are a result of several innovations in the architecture of the network in the form of novel processing units, modified connections, new designs of processing blocks, number of layers, and input size. For example, the inception concept was introduced by the relatively small CNN called GoogLeNet. This network uses filters of various sizes to discover spatial relationships among the pixels of the input image. Other designs targeted a reduction in computational complexity by introducing new type of processing operation called shuffling, which gave the network its name, ShuffleNet. The models are also different in their depth (e.g., number of layers), with some models (e.g., Inceptionv3 and ResNet) relying upon a significant increase in the number of layers to efficiently discover more complex relationships in the input. Moreover, residual learning and shortcut connections were introduced in the ResNet architecture. Instead of using filters of square (i.e., symmetric) dimensions, the Inceptionv3 model employed smaller asymmetric filters to reduce the cost of the computational overhead. Other design philosophies (e.g., DenseNet-201) adopted an approach that increases the number of connections in the network in an attempt to solve the vanishing gradient problem. Xception is a model that uses depthwise separable convolutions that replace the inception modules. The SqueezeNet model introduced new types of operations (i.e., squeeze and excite) that emphasize or de-emphasize features on the basis of their importance. The MobileNetv2 model was developed with an eye for smartphone and mobile environments (i.e., platforms with limited computational and memory capabilities). They achieved this by diminishing the number of parameters to be updated during training.

2.3. Performance Evaluation Setup

The hyperparameters were set to be the same values for all models: The number of images to be processed concurrently (i.e., the minimal batch size) was set to 16. This value is controlled by the memory footprint of the specific model and the available memory on the graphical processing unit. Higher values for the batch size result in faster training but more memory consumption, and vice versa. The learning rate was set to 0.0003. The maximal number of epochs was empirically set to 10. Further epochs did not result in an improvement in the loss or training curve (i.e., the curve was flattened). The stochastic gradient descent with momentum (SGDM) algorithm was used for training the network. The SGDM is known for its fast convergence [44]. No further fine-tuning or experimentation was conducted in setting up these parameters and internal algorithms.

Two strategies for the size of the training dataset were used. In the first one, 60% of the data were used for training; in the second one, 80% were used. The data-split step is followed by augmentation operations to optimize the learning process by adding randomness and variation in the input images [45]. This was achieved by random x- and y-axis translations in the range of $[-30, 30]$, and scaling randomly along the same axes in the range of $[0.9, 1.1]$. The size of the dataset remained the same after augmentation (i.e., originals were discarded). The models were implemented and evaluated on the MATLAB R2021a software running with 64 GB RAM, NVIDIA® GeForce RTX™ 3080 GPU with 10 GB dedicated memory.

2.4. Performance Evaluation Metrics

The performance of the grape disease classification models was evaluated and compared using the metrics shown in Equations (1)–(5). The definitions of the various terms are as follows. True positive (TP) is the number of correctly classified images in the disease states. True negative (TN) is the number of correctly classified images in the healthy state. False positive (FP) is the number of wrongly classified images in the disease states. False negative (FN) is the number of wrongly classified images in the healthy state. Since the number of images in different classes was not uniform, the F1 measure was better suited than accuracy for reflecting classification performance [46].

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{P} + \text{N}} \quad (1)$$

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

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

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (4)$$

$$\text{F1} = 2 \times \frac{\text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}} \quad (5)$$

3. Results and Discussion

The performance was evaluated with the goal of comparing the effectiveness of CNN models in identifying the correct disease or health class of the input images. The results represent the mean overall values from ten random runs of the model building, training, and validation codes. Moreover, the minimal, maximal, and standard deviation for the 10 runs were reported for the accuracy metric.

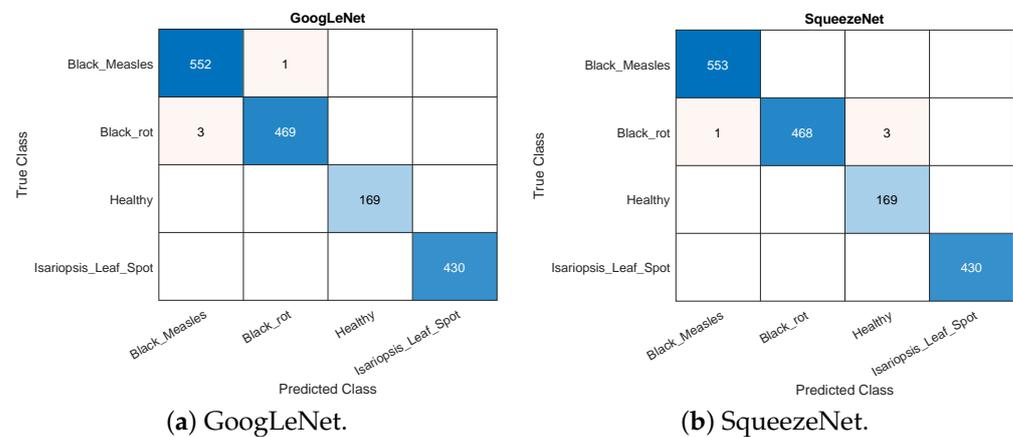
The first of the data-split strategies was 60/40. The results for the performance evaluation metrics are shown in Table 2. It was immediately evident that all models were able to easily discern the different types of diseases. The highest precision (i.e., 99.9%) was achieved by GoogLeNet, which is a relatively small and fast model. Other models achieved a very close performance, with the lowest precision being 99.1%. Further insight into the performance is provided by the confusion matrices in Figure 2. They show that, in most cases, different images were classified perfectly. However, misclassification between black rot and measles was the source of all errors. Table 3 shows the mean, minimum, maximum, and standard deviation of the accuracy taken over 10 runs and using 60% of the data for training. Some models were able to achieve a maximal accuracy of 100% in some of the multiple runs. The SqueezeNet and ResNet-18 models exhibited the most fluctuation with 1% and 0.7% standard deviation (SD), respectively.

Table 2. Results for the performance evaluation metrics using 60% of the data for training. Numbers represent mean overall values over 10 runs.

Model	F1 Score	Precision	Recall	Specificity
SqueezeNet	99.2%	99.1%	99.2%	99.7%
GoogLeNet	99.9%	99.9%	99.9%	99.9%
Inceptionv3	99.9%	99.9%	99.9%	99.9%
DenseNet-201	99.8%	99.8%	99.8%	99.9%
MobileNetv2	99.1%	99.1%	99.2%	99.7%
Resnet101	99.7%	99.7%	99.7%	99.9%
Resnet50	99.8%	99.8%	99.8%	99.9%
Resnet18	99.4%	99.4%	99.4%	99.7%

Table 2. Cont.

Model	F1 Score	Precision	Recall	Specificity
Xception	99.8%	99.8%	99.8%	99.9%
ShuffleNet	99.7%	99.7%	99.8%	99.9%
DarkNet-53	99.8%	99.8%	99.9%	99.9%

**Figure 2.** Examples of the confusion matrices for GoogLeNet and SqueezeNet models using 60% of the data for training.**Table 3.** The mean, minimum, maximum, and standard deviation of the accuracy taken over 10 runs and using 60% of the data for training.

Model	Mean	Maximum	Minimum	Standard Deviation
SqueezeNet	99.1%	99.8%	97.2%	1.0
GoogLeNet	99.8%	99.9%	99.7%	0.1
Inceptionv3	99.9%	100.0%	99.6%	0.1
DenseNet-201	99.8%	99.9%	99.3%	0.2
MobileNetv2	99.1%	99.7%	96.9%	0.9
Resnet101	99.7%	99.9%	99.0%	0.3
Resnet50	99.7%	100.0%	99.4%	0.2
Resnet18	99.2%	99.8%	97.5%	0.7
Xception	99.7%	99.8%	99.6%	0.1
ShuffleNet	99.7%	99.8%	99.6%	0.1
DarkNet-53	99.8%	100.0%	99.5%	0.2

Given the high performance of the deep-learning models using 60% of the data for training, there was a limited margin for improvement with the current dataset. Nonetheless, a further increase in the training subset to 80% of the dataset was examined. The mean overall F1 score, precision, recall, and specificity for all models and the 80/20 data split are shown in Table 4. All models improved their performance within the possible small margins, with some of them achieving perfect classification scores. Sample confusion matrices, as generated by MATLAB, for the ResNet101 and DarkNet-53 models using 80% of the data for training are shown in Figure 3. This corroborates the numbers in the previous tables. Moreover, Table 5 shows the mean, minimal, maximal, and standard deviation of the accuracy taken over 10 runs and using 80% of the data for training. In comparison to Table 3, more models were able to achieve a perfect accuracy in some of the runs, and the minimal accuracy increased for all models. Furthermore, the previously reported highest standard deviation values over the 10 runs were lower when using 80% of the data for training.

Table 4. Results for the performance evaluation metrics using 80% of the data for training. Numbers represent mean overall values over 10 runs.

Model	F1 Score	Precision	Recall	Specificity
SqueezeNet	99.3%	99.3%	99.4%	99.7%
GoogLeNet	99.9%	99.9%	99.9%	99.9%
Inceptionv3	100.0%	100.0%	100.0%	100.0%
DenseNet-201	99.8%	99.8%	99.8%	99.9%
MobileNetv2	99.6%	99.6%	99.6%	99.8%
Resnet101	99.4%	99.5%	99.4%	99.8%
Resnet50	99.9%	99.9%	99.9%	99.9%
Resnet18	99.7%	99.7%	99.7%	99.9%
Xception	100.0%	100.0%	100.0%	100.0%
ShuffleNet	99.8%	99.8%	99.8%	99.9%
DarkNet-53	99.9%	99.9%	99.9%	99.9%

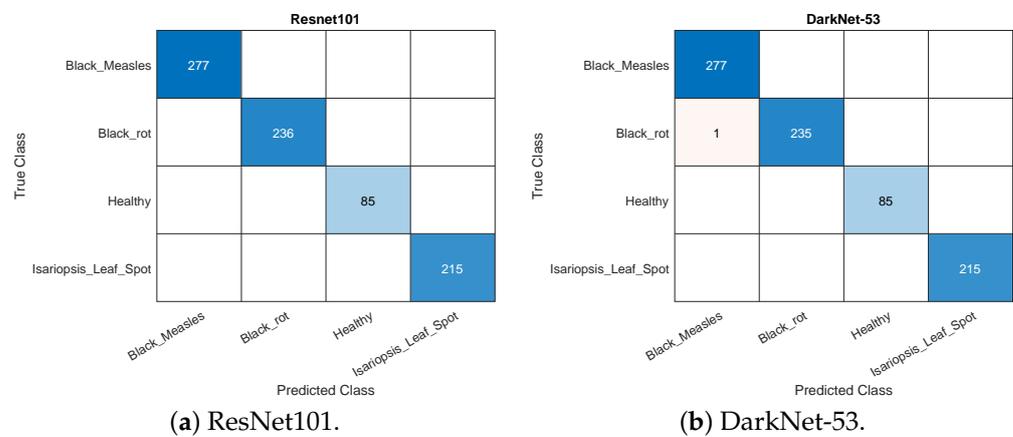


Figure 3. Examples of the confusion matrices for ResNet101 and DarkNet-53 models using 80% of the data for training.

Table 5. The mean, minimum, maximum, and standard deviation of the accuracy taken over 10 runs and using 80% of the data for training.

Model	Mean	Maximum	Minimum	Standard Deviation
SqueezeNet	99.3%	99.9%	98.2%	0.6
GoogLeNet	99.8%	100.0%	99.1%	0.3
Inceptionv3	100.0%	100.0%	99.9%	0.1
DenseNet-201	99.8%	100.0%	99.3%	0.2
MobileNetv2	99.5%	100.0%	98.3%	0.6
Resnet101	99.5%	100.0%	98.6%	0.5
Resnet50	99.8%	100.0%	99.5%	0.1
Resnet18	99.6%	99.9%	99.0%	0.3
Xception	100.0%	100.0%	99.9%	0.1
ShuffleNet	99.8%	99.9%	99.6%	0.1
DarkNet-53	99.9%	100.0%	99.8%	0.1

The time results in the various performance evaluation scenarios are shown in Table 6. There were huge differences in the training or validation times between the various models, even though their classification abilities were very similar. Moreover, there was a huge difference between the slowest and fastest models (i.e., 150.6 vs. 3185.4 s using a 60/40 split, and 169.5 vs. 4038 s using an 80/20 split). In addition, increasing the size of the training set resulted in an increase in a wide range of factors across the various models (e.g., the scalability of SqueezeNet vs. Xception).

Table 6. Time results in various performance evaluation scenarios. All times are averages of 10 runs and in seconds.

Data Split	60/40	80/20
Model		
SqueezeNet	150.6	169.5
GoogLeNet	276.3	332.46
Inceptionv3	804.7	1005.44
DenseNet-201	2607.4	3226.6
MobileNetv2	1157.1	1473.7
ResNet101	813.8	1008.0
ResNet50	369.6	457.5
ResNet18	162.3	185.9
Xception	3185.4	4038
ShuffleNet	863.2	1075.7
DarkNet-53	642.2	790.5

The research landscape on the use of AI in agriculture and specifically in disease identification is receiving increased focus and efforts. Table 7 shows the state-of-the-art results in classifying grape diseases. Huang et al. [14] achieved an accuracy range that could reach 100%. However, the highest results were produced using an ensemble model that combines features from multiple individual models. Such an approach involves high overhead, as deep-learning models require extensive computational and special requirements. Moreover, although they used the same dataset as the work in this paper, the dataset size was manipulated by augmentation to balance the number of images across all classes. This artificial inflation of a dataset can bias the results due to data leaking, as slightly modified copies of the same image can be easily detected by deep-learning algorithms. In essence, the algorithm was trained and tested on the same images. Thet et al. [15] repeated their training several times and kept the results from the best performing model, which may not reflect the stable performance of the model. Lauguico et al. [16] worked in a different way compared to the literature: by montaging and combining multiple disease images (i.e., 9 images in a 3×3 matrix) into a single image. After that, object recognition algorithms rather than classification methods were used to detect the various diseases. However, such an approach seems to needlessly complicate the problem in order to apply the object detection algorithms. Furthermore, it does not correspond to real-life application scenarios of the algorithm (i.e., why and how would a user combine multiple diseases into a single image?). Ji et al. [17] employed a similar approach as that of Huang et al. [14] by combining multiple CNN models. However, the main critique of their work lies in the small number of images in their dataset (1619 images in total) with two classes tested with much fewer than 100 images. Waghmare et al. [23], and Ansari et al. [25] used even fewer total images, namely, 450 and 400, respectively. Deep-learning algorithms learn better, and achieve a more stable and less variant performance with a large number of images [47]. Goncharov et al. [22] expanded their dataset by dividing single images into at least five new images each, which is the opposite approach to the one employed by Lauguico et al. [16]. However, this may skew the results, as this manual data manipulation artificially aids the deep-learning algorithms by pinpointing and segmenting important features. A similar duplication that definitely leads to data leaking and inflated performance results was employed by Liu et al. [19], and Andrushia et al. [21].

The present study has some limitations. First, more grape diseases (e.g., chlorosis and nutritional deficiency) need to be included in order to develop a truly grape-dedicated comprehensive application. Moreover, further disease types from other plants can be collectively included in the dataset. However, such diversification and variety in the application may not be necessary. Customizing applications to special types of plants may be more accurate and useful (i.e., multiple deeply specialized applications versus one holistic model). Second, the input images were taken using a standard background that does not reflect real-life scenarios and eliminates many sources of classification errors.

Third, in order to add more images into the dataset, and improve the classification accuracy and robustness of the models, mobile applications need to be developed and deployed.

Table 7. State-of-the-art results in classifying grape diseases using leaf images.

Study	Performance
Huang et al. [14]	Accuracy range of 77,100%.
Thet et al. [15]	Accuracy = 98.4%.
Lauguico et al. [16]	Accuracy = 95.65% (AlexNet).
Ji et al. [17]	F1 score = 98.96%.
Lin et al. [18]	Accuracy = 86.29% (GrapeNet).
Liu et al. [19]	Accuracy = 97.22%.
Tang et al. [20]	Accuracy = 99.14%.
Andrushia et al. [21]	Accuracy = 99.12%.
Goncharov et al. [22]	Accuracy = 98.4%.
Waghmare et al. [23]	Accuracy = 96.6%.
Jaisakthi et al. [24]	Accuracy = 93%.
Ansari et al. [25]	Precision = 97%.
Hasan et al. [28]	Accuracy = 91.37%.
This work	>99.1% for all performance metrics.

4. Conclusions

Grapes are highly susceptible to damage inflicted by diseases, which may be caused by insects or fungi. The effect of this damage is further exacerbated by more frequent events caused by global warming, such as unseasonable temperatures and extreme weather conditions. Having these factors coupled with global instability and rising prices, the need for plant disease protection cannot be overemphasized. To this end, this paper evaluated the adaptation of deep-learning convolutional neural networks in classifying grape diseases from images of infected leaves.

Technology has found its way into all aspects of daily life with varying success and impact. In this regard, artificial intelligence is one of the most promising and pervasive areas for innovation. Deep-learning algorithms can be used to solve many research problems. Convolutional neural networks are the best suitable for image-based applications. The results in this paper demonstrate that the transfer-learning approach has the potential to achieve reliable performance. These results demonstrate that it is possible to classify grape diseases in a manner that satisfies field deployment performance requirements. Moreover, the approach in this work performs with little overhead, no preprocessing or image processing, and no explicit feature extraction.

Author Contributions: Conceptualization, M.F.; methodology, M.F. and E.F.; software, M.F. and E.F.; validation, M.F. and E.F.; formal analysis, M.F.; investigation, M.F., E.F. and N.K.; resources, M.F. and N.K.; data curation, M.F. and E.F.; writing—original-draft preparation, M.F. and E.F.; writing—review and editing, M.F. and N.K.; supervision, M.F. and N.K.; project administration, M.F.; funding acquisition, M.F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial intelligence
FAO	Food and agriculture organization
CNN	Convolutional neural network
TP	True positive
TN	True negative
FN	False negative
FP	False positive
TPR	True positive rate
N	Negatives
P	Positives
SGDM	Stochastic gradient descent with momentum
SD	Standard deviation
SVM	Support vector machine

References

1. Fina, F.; Birch, P.; Young, R.; Obu, J.; Faithpraise, B.; Chatwin, C. Automatic plant pest detection and recognition using k-means clustering algorithm and correspondence filters. *Int. J. Adv. Biotechnol. Res.* **2013**, *4*, 189–199.
2. Li, L.; Zhang, S.; Wang, B. Plant disease detection and classification by deep learning—A review. *IEEE Access* **2021**, *9*, 56683–56698. [[CrossRef](#)]
3. Zhou, C.; Zhang, Z.; Zhou, S.; Xing, J.; Wu, Q.; Song, J. Grape leaf spot identification under limited samples by fine grained-GAN. *IEEE Access* **2021**, *9*, 100480–100489. [[CrossRef](#)]
4. Lee, S.H.; Goëau, H.; Bonnet, P.; Joly, A. New perspectives on plant disease characterization based on deep learning. *Comput. Electron. Agric.* **2020**, *170*, 105220. [[CrossRef](#)]
5. Mordor Intelligence. Grapes Market | 2022–27 | Industry Share, Size, Growth—Mordor Intelligence—mordorintelligence.com. Available online: <https://www.mordorintelligence.com/industry-reports/grapes-market> (accessed on 5 September 2022).
6. Demchak, K. Black Rot of Grapes. Available online: <https://extension.psu.edu/> (accessed on 5 September 2022).
7. Liu, B.; Tan, C.; Li, S.; He, J.; Wang, H. A data augmentation method based on generative adversarial networks for grape leaf disease identification. *IEEE Access* **2020**, *8*, 102188–102198. [[CrossRef](#)]
8. Liu, X.; Wang, H.; Lin, B.; Tao, Y.; Zhuo, K.; Liao, J. Loop-mediated isothermal amplification based on the mitochondrial COI region to detect *Pratylenchus zeae*. *Eur. J. Plant Pathol.* **2017**, *148*, 435–446. [[CrossRef](#)]
9. Alajas, O.J.; Concepcion, R.; Dadios, E.; Sybingco, E.; Mendigoria, C.H.; Aquino, H. Prediction of Grape Leaf Black Rot Damaged Surface Percentage Using Hybrid Linear Discriminant Analysis and Decision Tree. In Proceedings of the 2021 International Conference on Intelligent Technologies (CONIT), Hubli, India, 24–26 June 2021; pp. 1–6.
10. Ji, M.; Wu, Z. Automatic detection and severity analysis of grape black measles disease based on deep learning and fuzzy logic. *Comput. Electron. Agric.* **2022**, *193*, 106718. [[CrossRef](#)]
11. Esteban, M.A.; Villanueva, M.J.; Lissarrague, J. Effect of irrigation on changes in berry composition of Tempranillo during maturation. Sugars, organic acids, and mineral elements. *Am. J. Enol. Vitic.* **1999**, *50*, 418–434.
12. Maia, A.; Oliveira, J.; Schwan-Estrada, K.; Faria, C.; Batista, A.; Costa, W.; Batista, B. The control of isariopsis leaf spot and downy mildew in grapevine cv. Isabel with the essential oil of lemon grass and the activity of defensive enzymes in response to the essential oil. *Crop Prot.* **2014**, *63*, 57–67. [[CrossRef](#)]
13. Sharma, N.; Jain, V.; Mishra, A. An Analysis Of Convolutional Neural Networks For Image Classification. *Procedia Comput. Sci.* **2018**, *132*, 377–384. [[CrossRef](#)]
14. Huang, Z.; Qin, A.; Lu, J.; Menon, A.; Gao, J. Grape Leaf Disease Detection and Classification Using Machine Learning. In Proceedings of the 2020 International Conferences on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData) and IEEE Congress on Cybermatics (Cybermatics), Rhodes Island, Greece, 2–6 November 2020; pp. 870–877.
15. Thet, K.Z.; Htwe, K.K.; Thein, M.M. Grape leaf diseases classification using convolutional neural network. In Proceedings of the 2020 International Conference on Advanced Information Technologies (ICAIT), Yangon, Myanmar, 4–5 November 2020; pp. 147–152.
16. Lauguico, S.; Concepcion, R.; Tobias, R.R.; Bandala, A.; Vicerra, R.R.; Dadios, E. Grape leaf multi-disease detection with confidence value using transfer learning integrated to regions with convolutional neural networks. In Proceedings of the 2020 IEEE Region 10 Conference (TENCON), Osaka, Japan, 16–19 November 2020; pp. 767–772.
17. Ji, M.; Zhang, L.; Wu, Q. Automatic grape leaf diseases identification via UnitedModel based on multiple convolutional neural networks. *Inf. Process. Agric.* **2020**, *7*, 418–426. [[CrossRef](#)]
18. Lin, J.; Chen, X.; Pan, R.; Cao, T.; Cai, J.; Chen, Y.; Peng, X.; Cernava, T.; Zhang, X. GrapeNet: A Lightweight Convolutional Neural Network Model for Identification of Grape Leaf Diseases. *Agriculture* **2022**, *12*, 887. [[CrossRef](#)]

19. Liu, B.; Ding, Z.; Tian, L.; He, D.; Li, S.; Wang, H. Grape Leaf Disease Identification Using Improved Deep Convolutional Neural Networks. *Front. Plant Sci.* **2020**, *11*. [[CrossRef](#)] [[PubMed](#)]
20. Tang, Z.; Yang, J.; Li, Z.; Qi, F. Grape disease image classification based on lightweight convolution neural networks and channelwise attention. *Comput. Electron. Agric.* **2020**, *178*, 105735. [[CrossRef](#)]
21. Andrushia, A.D.; Neebha, T.M.; Patricia, A.T.; Umadevi, S.; Anand, N.; Varshney, A. Image-based disease classification in grape leaves using convolutional capsule network. *Soft Comput.* **2022**. [[CrossRef](#)]
22. Goncharov, P.; Ososkov, G.; Nechaevskiy, A.; Uzhinskiy, A.; Nestsiarenia, I. Disease detection on the plant leaves by deep learning. In Proceedings of the International Conference on Neuroinformatics, Moscow, Russia, 8–12 October 2018; pp. 151–159.
23. Waghmare, H.; Kokare, R.; Dandawate, Y. Detection and classification of diseases of Grape plant using opposite colour Local Binary Pattern feature and machine learning for automated Decision Support System. In Proceedings of the 2016 3rd International Conference on Signal Processing and Integrated Networks (SPIN), Noida, India, 11–12 February 2016. [[CrossRef](#)]
24. Jaisakthi, S.; Mirunalini, P.; Thenmozhi, D.; Vatsala. Grape Leaf Disease Identification using Machine Learning Techniques. In Proceedings of the 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), Chennai, India, 21–23 February 2019. [[CrossRef](#)]
25. Ansari, A.S.; Jawarneh, M.; Ritonga, M.; Jamwal, P.; Mohammadi, M.S.; Veluri, R.K.; Kumar, V.; Shah, M.A. Improved Support Vector Machine and Image Processing Enabled Methodology for Detection and Classification of Grape Leaf Disease. *J. Food Qual.* **2022**, *2022*, 1–6. [[CrossRef](#)]
26. Zhai, G.; Min, X. Perceptual image quality assessment: A survey. *Sci. China Inf. Sci.* **2020**, *63*. [[CrossRef](#)]
27. Min, X.; Ma, K.; Gu, K.; Zhai, G.; Wang, Z.; Lin, W. Unified Blind Quality Assessment of Compressed Natural, Graphic, and Screen Content Images. *IEEE Trans. Image Process.* **2017**, *26*, 5462–5474. [[CrossRef](#)]
28. Hasan, M.A.; Riana, D.; Swasono, S.; Priyatna, A.; Pudjiarti, E.; Prahartiwi, L.I. Identification of Grape Leaf Diseases Using Convolutional Neural Network. *J. Phys. Conf. Ser.* **2020**, *1641*, 012007. [[CrossRef](#)]
29. Pandian, J.A.; Geetharamani, G. Data for: Identification of Plant Leaf Diseases Using a 9-layer Deep Convolutional Neural Network. *Comput. Electr. Eng.* **2019**, *76*, 323–338. [[CrossRef](#)]
30. Kim, H.E.; Cosa-Linan, A.; Santhanam, N.; Jannesari, M.; Maros, M.E.; Ganslandt, T. Transfer learning for medical image classification: A literature review. *BMC Med. Imaging* **2022**, *22*, 69. [[CrossRef](#)] [[PubMed](#)]
31. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [[CrossRef](#)]
32. Khasawneh, N.; Fraiwan, M.; Fraiwan, L.; Khassawneh, B.; Ibban, A. Detection of COVID-19 from Chest X-ray Images Using Deep Convolutional Neural Networks. *Sensors* **2021**, *21*, 5940. [[CrossRef](#)] [[PubMed](#)]
33. Fraiwan, M.; Audat, Z.; Fraiwan, L.; Manasreh, T. Using deep transfer learning to detect scoliosis and spondylolisthesis from x-ray images. *PLoS ONE* **2022**, *17*, e0267851. [[CrossRef](#)]
34. Jia, J.; Zhai, G.; Zhang, J.; Gao, Z.; Zhu, Z.; Min, X.; Yang, X.; Guo, G. EMBDN: An Efficient Multiclass Barcode Detection Network for Complicated Environments. *IEEE Internet Things J.* **2019**, *6*, 9919–9933. [[CrossRef](#)]
35. Zhang, J.; Min, X.; Jia, J.; Zhu, Z.; Wang, J.; Zhai, G. Fine localization and distortion resistant detection of multi-class barcode in complex environments. *Multimed. Tools Appl.* **2020**, *80*, 16153–16172. [[CrossRef](#)]
36. Redmon, J. Darknet: Open Source Neural Networks in C. 2013–2016. Available online: <https://pjreddie.com/darknet/> (accessed on 5 September 2022).
37. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269. [[CrossRef](#)]
38. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9. [[CrossRef](#)]
39. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A.A. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 4278–4284.
40. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]
41. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6848–6856. [[CrossRef](#)]
42. Iandola, F.N.; Moskewicz, M.W.; Ashraf, K.; Han, S.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <1 MB model size. *arXiv* **2016**, arXiv:1602.07360v4.
43. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–27 July 2017; pp. 1800–1807. [[CrossRef](#)]
44. Qian, N. On the momentum term in gradient descent learning algorithms. *Neural Netw.* **1999**, *12*, 145–151. [[CrossRef](#)]
45. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 60. [[CrossRef](#)]

-
46. Tharwat, A. Classification assessment methods. *Appl. Comput. Inform.* **2020**, *17*, 168–192. [[CrossRef](#)]
 47. Sarker, I.H. Deep Learning: A Comprehensive Overview on Techniques, Taxonomy, Applications and Research Directions. *SN Comput. Sci.* **2021**, *2*, 420. [[CrossRef](#)]