

Research Article

Using Multioutput Learning to Diagnose Plant Disease and Stress Severity

Gianni Fenu  and Francesca Maridina Mallocci 

Department of Mathematics and Computer Science, University of Cagliari, Cagliari 09124, Italy

Correspondence should be addressed to Francesca Maridina Mallocci; francescam.mallocci@unica.it

Received 29 October 2020; Revised 13 January 2021; Accepted 25 January 2021; Published 10 February 2021

Academic Editor: Atif Khan

Copyright © 2021 Gianni Fenu and Francesca Maridina Mallocci. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Early diagnosis of leaf diseases is a fundamental tool in precision agriculture, thanks to its high correlation with food safety and environmental sustainability. It is proven that plant diseases are responsible for serious economic losses every year. The aim of this work is to study an efficient network capable of assisting farmers in recognizing pear leaf symptoms and providing targeted information for rational use of pesticides. The proposed model consists of a multioutput system based on convolutional neural networks. The deep learning approach considers five pretrained CNN architectures, namely, VGG-16, VGG-19, ResNet50, InceptionV3, MobileNetV2, and EfficientNetB0, as feature extractors to classify three diseases and six severity levels. Computational experiments are conducted to evaluate the model on the DiaMOS Plant dataset, a self-collected dataset in the field. The results obtained confirm the robustness of the proposed model in automatically extracting the discriminating features of diseased leaves by adopting the multitasking learning paradigm.

1. Introduction

Plant diseases are one of the most affecting factors in agricultural production, as they represent the principal cause of severe economic losses. The Food and Agriculture Organization of the United Nations (FAO) estimates that up to 40% of food crops are lost due to plant pests and diseases annually [1]. As investigated by the study in [2], plant disease protection has become a research hotspot, because it is a highly correlated problem with food security, environmental sustainability, and climate change. Thus, it is crucial and impactful in production, so much to be a pivotal tool in precision agriculture (PA) [3].

Prediction of plant and crop disease is a complex and interconnected problem to be solved, requiring considerable and different technical skills. Plant health is increasingly under threat. In the last years, the spread of noxious plant diseases is further aggravated and accelerated by global trade [1]. Pests and diseases can disseminate in different ways and symptoms, outside their native place, along new frontiers, where there are no previous skills to contrast them. This

inevitably involves treating the disease with tools that are sometimes ineffective, aggressive, and superfluous as well as impacting on environmental sustainability. Indeed, disease identification typically sees the involvement of a field specialist, who, through a careful analysis of the canopy, is able to make a diagnosis from the onset of the first symptoms. However, even an experienced eye can make mistakes. Furthermore, not all farmers can afford counseling, because it is expensive financially and temporally. Therefore, it is necessary to adopt an ecosystem approach, supported by effective tools capable of providing more precise and timely assistance in the treatment of leaf diseases, such as Decision Support Systems (DSS) [4] or Computer-Aided Diagnosis (CAD) systems, that allow any farmer with access to a smartphone to enjoy expert knowledge in a practical and low-cost way [5].

The literature has explored plant disease diagnosis using various state-of-the-art techniques. Some studies have addressed it with Machine Learning and Artificial Intelligence methods such as Neural Network [6], Support Vector Machine [7], Random Forest, and K -nearest neighbor

[8–10]. While more recent studies have applied deep learning models, in detail Convolutional Neural Networks (CNNs), since they have shown relevant results in the image recognition task. In most works, the research focused on solving the problem by identifying only leaf disease.

Sladojevic et al. [11] studied a convolutional neural network to recognize plant disease. The network was able to recognize 13 different types of plant diseases out of healthy leaves, with the ability to distinguish plant leaves from their surroundings. The images were collected by the authors searching for the name of the disease and the plant on the Internet. Liu et al. [12] designed a novel architecture based on AlexNet and GoogLeNet's inception networks to identify four common apple leaf diseases. Using a dataset of 13,689 synthetic images, the developed model provided a feasible solution for the identification and recognition of apple leaf diseases. Similarly, Yan et al. [13] proposed a method based on an improved VGG-16 network to identify four apple leaf diseases. The model trained on a lab-built dataset of 2446 achieved a high accuracy rate and a fast convergence speed. The work was based on the "2018 AI Challenger Global Challenge" dataset. For other contributions, the reader is referred to [14–16].

On the other hand, a more limited effort has focused on identifying severity stress, considered by Kranz [17] and Bock et al. [18], an important task to manage pests and diseases, to predict yield, and to recommend control treatments, but also for understanding fundamental processes in biology, including coevolution and plant disease epidemiology [18]. This limited contribution is due to the lack of representative data containing such essential information. Wang et al. [19] proposed a deep learning approach to automatically discover the discriminative features for estimating apple black rot disease severity. The images labeled with four degrees of severity were extracted from PlantVillage dataset. The authors compared different state-of-the-art architectures as VGG-16, VGG-19, Inception-v3, and ResNet50, where VGG-16 achieved better performance than the other models. A different approach was performed by Barbedo [20], who manually extracted the symptoms from the entire leaf to identify multiple lesions from the same leaf. For other contributions, the reader is referred to [21, 22].

As can be observed from the reported literature, the research has widely explored the diagnosis of plant diseases, considering the problem in two subproblems. The commonly adopted approach trains two separate networks, one for diagnosing the disease and one for estimating severity.

Recently, an alternative method is emerging that explores the problem as two joint sets using the potential of Multitask Learning (MTL). MTL is a learning paradigm that solves multiple tasks employing a shared architecture. Its application potential in precision agriculture is starting to be an object of study only recently through modern analysis techniques. This observation is inferred and reinforced from the currently limited number of scientific contributions, briefly described as follows. Ghosal et al. [23] developed a deep machine vision framework to identify, classify, and quantify eight stresses, divided into biotic and abiotic stress, affecting soybean leaves.

The designed framework also included an unsupervised method to extract high-resolution feature maps that isolate visual symptoms used to measure stress severity. Liang et al. [24] proposed a multitasking system, called PD²SE-Net, able to recognize plant species, to diagnose diseases, and to estimate the severity of diseases. The experiments based on PlantVillage dataset estimated the stress severity classifying the leaves in one out of three classes: healthy, general, and serious. The results confirmed the robustness of the proposed architecture in classifying all three problems. Similarly, Esgario et al. [25] estimated the disease and severity of coffee leaves using a multitask system based on a convolutional neural network. The results demonstrated the effectiveness of this approach in solving the problem.

In this study, we investigate the potential of MTL in the diagnosis of pear leaf disease, based on the assumption that disease diagnosis and severity estimation are two closely related tasks. The main contributions of this work are as follows:

- (1) The firstly large and representative image dataset of healthy and diseased pear leaves is presented into the literature, called DiaMOS Plant dataset.
- (2) An image-based multioutput convolutional neural network to classify biotic stress and identify the related severity affecting pear leaves is studied.
- (3) Comparative studies of different deep learning architectures are conducted.

2. Materials and Methods

The entire procedure to diagnose plant disease and stress severity considered several CNNs deep learning architectures, which are described further in detail. The approach is divided into several steps illustrated in the sections below. Figure 1 illustrates the flowchart below. Based on the data collected, we firstly perform preprocessing and data augmentation to improve the model generalization. Secondly, we train different improved pretrained CNNs to conduct the identification and classification tasks in plant leaves.

2.1. Dataset Collection. An issue that can be inferred in the literature is the lack of representative datasets for the designated problem. Most of the proposed techniques are trained using lab-built datasets, such as PlantVillage [26], in which foliar diseases are portrayed only on the ventral side of the leaf, on a homogeneous background. However, in the real world, it is not possible to have a controlled environment to take photos in perfect conditions, i.e., with the right lighting and angle. Besides, a system should be able to analyze the disease as it occurs directly on the plant. The scarcity of images is certainly due to the fact that the construction of a dataset is an expensive and costly process.

In this work, we collected a field dataset to diagnose and monitor plants' symptoms called DiaMOS Plant. DiaMOS Plant dataset contains pear leaves images affected by three main biotic stresses, mainly occurring on foliage (Figure 2). The images were gathered using different devices including a

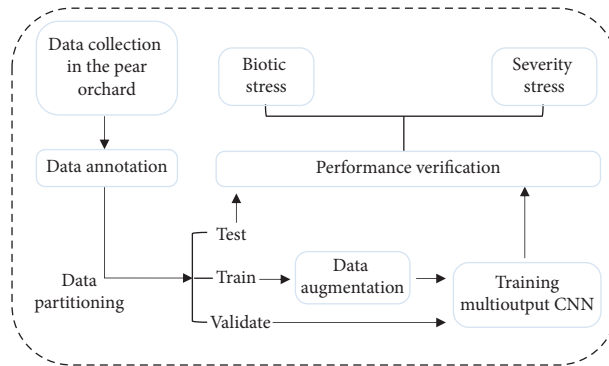


FIGURE 1: System overview of the proposed deep-learning-based approach to diagnose plant diseases.

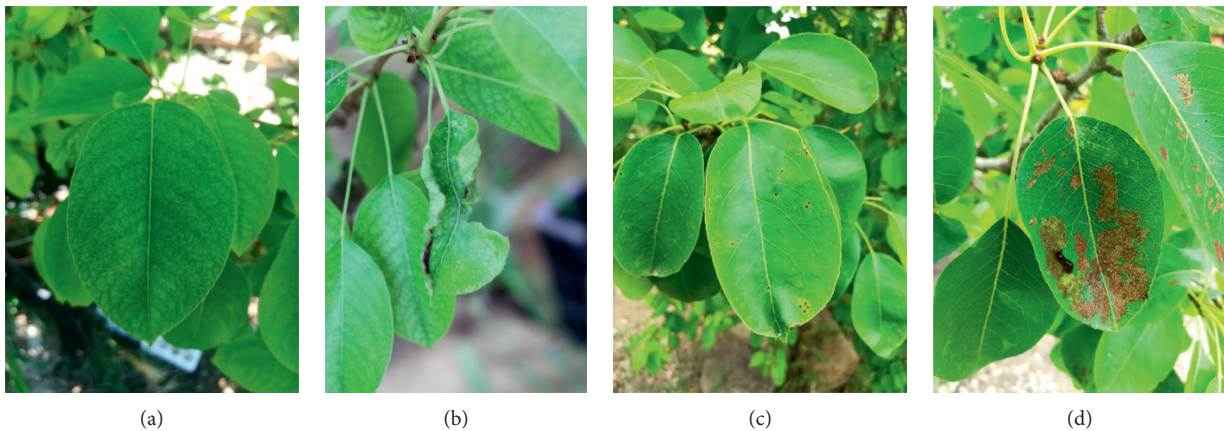


FIGURE 2: Examples of pear leaves affected by different symptoms.

smartphone (Honor 6x) and DSRL camera (Canon EOS 60D). The leaves were captured from the adaxial (upper) leaf side, in a real-world condition without any criteria to make the dataset more heterogeneous. In addition, the images were collected at different times of the year, from February to July, in order to capture the disease evolution from the first symptoms. Thus, it means that the models trained with it can monitor the plant health status and make better decisions to improve precision agriculture management. A total of 3057 images were collected, including healthy leaves and diseased leaves, affected by one or more of the following biotic stresses: leaf spot, leaf curl, and slug damage. The stress severity was calculated identifying five classes expressed as no risk (0%), very low (1–5%), low (6–20%), medium (21–25%), and high (>50%) in a range from 0 to 4. A detailed summary of the dataset is provided in Table 1.

2.2. Data Augmentation. Deep learning models have a high learning capacity that allows them to solve classification and prediction tasks with relevant results, particularly on perceptual problems that receive as input high-dimensional samples as images. However, complex models tend to decrease their generalization capability when trained with a small dataset, an issue known as overfitting. Data augmentation is a technique adopted to mitigate overfitting in computer vision. It takes the approach of generating more

TABLE 1: Dataset details. It presents the number of images for each biotic stress and severity.

Biotic stress	Leaf dataset	Severity	Leaf dataset
Healthy	43	0	43
Spot	884	1	682
Curl	105	2	1139
Slug	2025	3	699
Total	3,057	4	390

training data from existing training samples, by augmenting the samples via a number of random transformations that yield believable-looking images [27]. In the literature, there are several methods to introduce more variability into the dataset. The standard techniques include rotation, shearing, zooming, cropping, flipping, and color variation. In this work, we focused on standard augmentation in order to improve the performance and ability of the model to generalize.

2.3. Deep Learning Networks. We adopt six well-known convolutional neural network architectures, including VGG (VGG-16, VGG-19), residual neural network (ResNet50), InceptionV3, MobileNetV2, and EfficientNetB0, since they showed good generalization skills in previous works where the problem was treated as a single task, i.e., disease diagnosis or severity estimation.

2.3.1. VGG. The VGG network follows the archetypal pattern of classic convolutional networks. Proposed by Visual Geometric Group (VGG) at the University of Oxford [28] in the year of 2014, it scored first place on the image localization task and second place on the image classification task in the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC). The novelty of VGGNet was its simplicity in using a deeper layer with smaller filters. The model requires as input a fixed-size 224×224 RGB image. The pre-processing was performed, which consisted in subtracting the mean RGB value from each pixel. An operation was computed over the whole training set. The analysis is performed by a stack of 5 convolutional layers, each of which is followed by a Max Pooling layer in order to reduce the volume size. The final Max Pooling layer is followed by three fully connected (FC) layers. VGG network presents different variants as VGG-16 and VGG-19, which use the same architecture with different number of layers. VGG-14 uses 14 layers, whereas VGG-16 uses 19 layers.

2.3.2. ResNet. He et al. [29] developed the residual neural network (ResNet) to address the problem of vanishing/exploding gradients and accuracy degradation by introducing the concept of residual learning. In general, both problems occur with increasing depth. The first issue, as the number of layers increases, the gradient of propagation in the network may become tenuous or even be lost entirely, rendering the network untrainable. While the second issue, with the network depth increasing, the accuracy gets saturated and then degrades rapidly. To solve these phenomena, the researchers proposed residual connections. A residual connection consists of making the output of an earlier layer available as an input to a later layer, effectively creating a shortcut in a sequential network. Rather than being concatenated to the later activation, the earlier output is summed with the later activation, which assumes that both activations are the same size [27] (Figure 3).

2.3.3. Inception. It is a popular network introduced by Szegedy et al. [30] in 2014. It achieved a milestone in the development of CNN classifiers when previous architectures focused only on improving the performance compromising the computational cost. Differently from VGG, which has achieved remarkable accuracy with a highly computationally expensive architecture, inception implements several expedients to efficiently manage computational resources in terms of cost as well as the number of parameters. The model is relied on a directed acyclic graph, where the input is processed by several parallel convolutional branches whose outputs are then merged back into a single tensor. This structure helps the network separately learn spatial feature and channel-wise features, which is more efficient than learning them jointly [27].

2.3.4. MobileNet. It is a class of convolutional neural networks designed for mobile and embedded vision

applications, which are structured to reduce the computational complexity required by each convolutional layer. Ideated by Howard et al. [31], they are based on a streamlined architecture that uses depthwise separable convolutions to build lightweight deep neural networks. These improvements have achieved introducing two simple global hyperparameters that allow the model builder to choose the right-size model for their application based on the constraints of the problem.

2.3.5. EfficientNet. It is a new family of convolutional neural networks released in 2019 by Tan and Le [32]. Inspired by MobileNet network, the authors examined the scaling of neural networks. They discovered that the best gains come from scaling the width, resolution (like the MobileNets), and depth of the network simultaneously. To this end, the authors used Neural Architecture Search to design a new baseline network and scale it up to obtain a family of models, which achieve much better accuracy and efficiency than previous convolutional neural networks such as MobileNet and ResNet.

2.4. Transfer Learning. Deep learning models typically require a large image dataset to achieve high predictive results. However, a situation common to complex-real problems is the lack of data, especially in precision agriculture sector. Data collection and labeling are onerous tasks that often require considerable technical expertise. In light of these challenges, transfer learning represents a strategy that yields reasonable results despite a relative lack of data. Indeed, it consists of taking features learned on one problem and leveraging them on a new, similar problem [27]. At the state of the art, it has proven to be a well-established and efficient technique in many previous studies for solving several image-based computer vision problems [33, 34]. There are distinct transfer learning methods, which can be applied based on the domain task at hand and the availability of data. The most common are as follows:

- (1) *Feature Extraction.* It uses the representations learned by a previous network to extract relevant features from new samples. To this end, it freezes the first layers of the pretrained network to avoid destroying any of the information previously learned, and it removes the last layers, which are replaced by a new classifier trained from scratch.
- (2) *Fine-Tuning.* It consists of unfreezing a few of the top layers of a frozen model base used for feature extraction and jointly training both the newly added part of the model and these top layers [27].

Because of the small amount of data in this study, we adopted the feature extraction technique in conjunction with the data augmentation in order to optimize and improve the model robustness.

2.5. Multioutput Convolutional Neural Network. Multioutput (multitask) learning is a paradigm based on the simultaneous prediction of multiple outputs given a single

Complexity

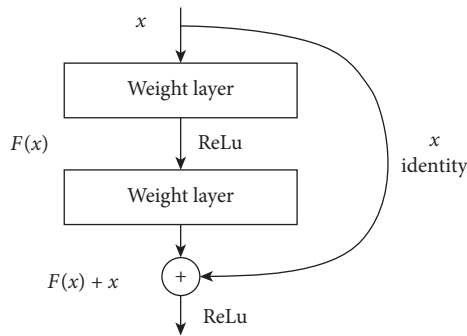


FIGURE 3: Residual learning: a building block. Source: [26].

input, as shown in Figure 4. Recently, its modeling algorithms have increasingly attracted interest from researchers due to its wide application, particularly on problems related to decision-making. Decisions in the real world often involve multiple complex factors and criteria [35].

The intertwining of these factors leads to the branching of studies in different forms according to the nature of the learning problem. Typical cases of multioutput learning include multilabel learning, multidimensional learning, and multitarget regression.

As aforementioned, the goal of our research work is to identify and develop a network architecture able to diagnose biotic stress and its degree of severity in pear leaves. From the technical point of view, it is a multitask learning problem, while in the field point of view, it is a multitasking problem that proves to be crucial in supporting farmers to deal with pathological adversities promptly. At the state of the art, several studies were carried out for the classification of the disease of different crops [13, 19], but not as many for the identification of the crop risk. The approach adopted in the pursuit of the objectives typically sees the use of two separate models.

In our study, we applied several CNN architectures to solve the problem jointly, similarly to what has been done by Esgario et al. [25] for coffee cultivation. This choice is dictated by the fact that the tasks are considered closely related. The problem requires predicting multiple target attributes of the input data. The use of two separate networks would be suboptimal, since the information extracted from the models could be redundant. A joint model would learn richer and more accurate representations of the space of the various diseases and vice versa. To this end, we developed a multioutput convolutional neural network which uses an improved convolutional base of a pretrained model (see Figure 5).

To enhance the robustness of the network to the CNN architectures, we added a global average pooling 2D layer, a batch normalization layer, and a fully connected layer.

A Global Average Pooling (GAP) is an operation that computes the average output of each feature map in the previous layer. We introduced it instead of the fully connected layer to prepare the model for the final classification as well as reduce parameters, as it was proven in [36] for its effectiveness as a regularizer.

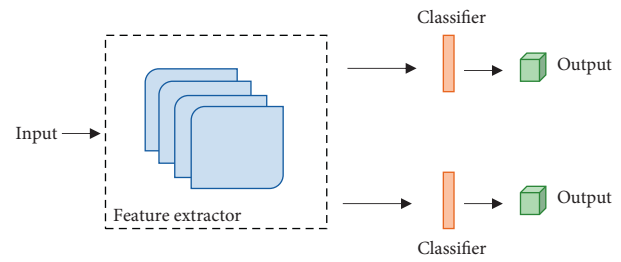


FIGURE 4: A multioutput (or multihead) model. In our study, as feature extractor, we used a pretrained convolutional neural network; as classifier, we adopted a fully connected layer.

The Batch Normalization (BN) layer [37] consists of normalizing the output of a previous activation layer by subtracting the batch mean and dividing by the batch standard deviation. Then, the layer shifts the input by a learnable offset β and scales it by a learnable scale factor γ . Its integration allowed us to enhance the stability of the model leading to faster learning rates.

Fully connected layers (FC) in a convolutional network consist of layers where all the inputs from one layer are connected to every activation unit of the next layer. The aim of the fully connected structure is to take the results of the convolution/pooling layer and use them to classify the data into various classes (labels). The result of convolution/pooling is flattened into a single vector of values, each representing a probability that a certain feature belongs to a label. Through the backpropagation process, it determines the most accurate weights for each neuron in order to formulate the classification decision. In our work, we introduced two fully connected layers to predict biotic stress and severity, respectively. Furthermore, a dropout regularization is added in the fully connected layer, since as demonstrated by Srivastava et al. [38], this strategy prevents complex coadaptations on the training data.

Dropout performs randomly dropping out a number of output features of the layer during training. This operation enhances the generalization because it forces the layer to learn with different neurons the same “concept.” To this end, we added it to reduce overfitting and improve the generalization of the model.

Figure 5 shows graphically the proposed framework. The foliar images are given as input to the modified CNN architecture with the addition of a global average pooling (GAP), a convolutional layer (Conv), a batch normalization layer (BN), and a new fully connected layer (FC).

In the first phase of the process, the convolutional neural network, through the use of increasingly refined and diversified filters according to the architecture in use (VGG-16, VGG-19, ResNet50, InceptionV3, MobileNetV2, or EfficientNetB0), acts as a sieve for image processing. At each step, the layers learn and extract increasingly complex representations as well as abstract visual concepts relevant for the problem at hand. From Figure 6, we can graphically observe the representations learned from the network at each step. In

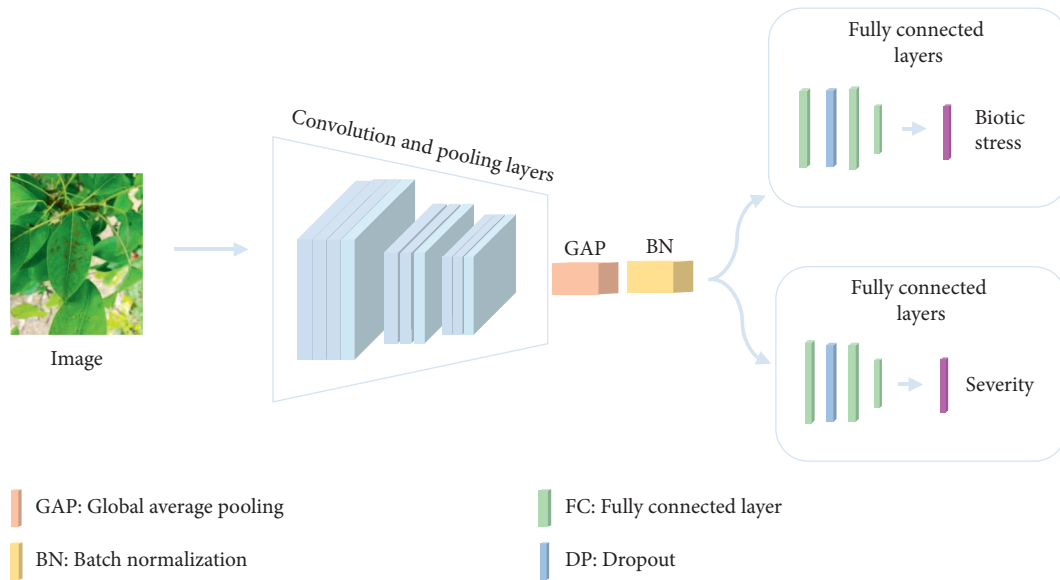


FIGURE 5: A schematic of the proposed image-based multioutput convolutional neural network.

the first layer, the edges are identified, so much so that activations retain almost all the information present in the initial image. Proceeding in the subsequent levels, activations become more abstract and begin to encode higher-level concepts, such as the shape of the leaf. Moreover, the convolutional base shared by both classification problems allows the network to jointly learn data representations useful to both tasks, optimizing the resources in use.

In the second step, the classification process is carried out by two fully connected layers, which uses a ReLU and softmax activation function. The two fully connected layers performed in parallel flatten the results of the features extractor into a single vector of values, each of which represents a probability that a certain feature belongs to a class representing the disease and severity, respectively. Finally, during the training, using a fraction rate equal to 0.5, a certain number of elements contained in the vector are randomly zeroed by the dropout technique in order to provide more accurate results.

3. Results and Discussion

In this section, we present the experimental setup and strategy as well as the obtained results.

3.1. Experimental Setup. The experimental framework written in Python language exploits the Keras deep learning 2.4.3 library based on TensorFlow 2.2.1 environment, executed on a server equipped with a 3.000 GHz Intel® Xeon® Gold. The dataset contains 3057 images categorized into four types of different pear leaf diseases, where only healthy, leaf spot and slug damage are considered for this work. The detailed summary dataset is provided in Table 1.

3.2. Experimental Analysis. To carry out the study, we divided the dataset into training, validation, and test datasets with a ratio of 7:2:1, respectively. To preserve the percentage of samples for each class, the dataset is split using the ShuffleSplit strategy provided by scikit-learn 0.23.2 library. Before training, we preprocessed the data to meet the requirements of CNN networks. All images are resized to $224 \times 224 \times 3$, which are reshaped into the shape the networks expect and scaled them so that all values are in the $[0,1]$ interval. Subsequently, we transformed them into a float32 array with values between 0 and 1. To improve the robustness of our model, the data augmentation technique is applied in real time during the training phase, performing horizontal and vertical mirroring, rotation, and color variation. The CNN networks receive for each batch slightly different images, whose analysis allows them to adjust the network's weights until the network learns the most relevant features for the given problem. To avoid a long training time, the transfer learning method is applied. The training was performed by adapting CNN networks trained using ImageNet dataset [39], which consists of images from a large variety of objects (1,000 categories). During this phase, the top layers are frozen for preventing their weights from being updated during training. Thus, with this setup, the representations that were previously learned from the convolutional base were not lost. The hyperparameters configurations used are presented in Table 2.

Furthermore, we monitored the model's validation loss to reduce the learning rate when it has stopped improving. This strategy allowed us to get out of local minima during training, a phenomenon known as Plateau [40]. The learning rate is decreased when the validation loss has stopped improving for 4 epochs, dividing it by 10. Finally, the states (set of weights) in which the networks presented the lowest loss

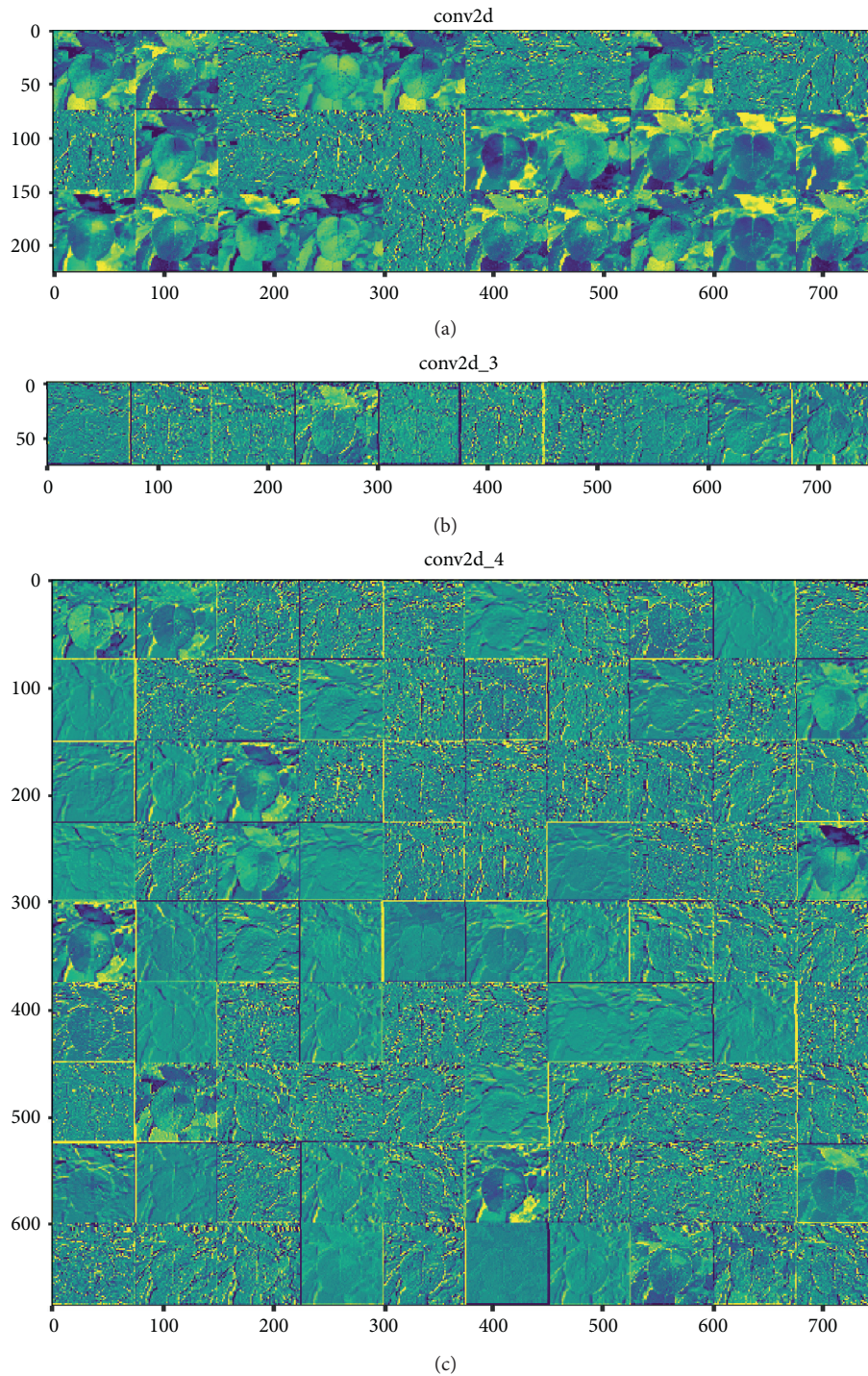


FIGURE 6: The outputs of the first blocks of convolutional layers of the EfficientNet architecture.

value for the validation set are saved. The saved models are evaluated with the test dataset, and the results are computed in terms of accuracy and training time.

3.3. Experimental Results. The obtained results are presented in Table 3, divided by category (biotic stress and severity) and CNN architecture.

TABLE 2: CNN training hyperparameters.

Parameter	Value
Optimizer	RMSprop
Loss function	Cross-entropy
Learning rate	$2e-5$
Momentum	0.9

Most of the models demonstrated a relevant generalization capacity for the identification of biotic stresses. Both versions of the VGG network achieved an accuracy of 81.12% and 83.91%, respectively. InceptionV3, MobileNetV2, and EfficientNetB0 performed better with an accuracy of 90.68%, 90.01%, 90.18%, respectively. Lowered accurate results were obtained by ResNet50 network with an accuracy of 80.51%. The ResNet50 classification results of biotic stress were consistent only for pear slug damage (see Figure 7). The network was unable to distinguish the different symptoms. These misclassifications may be associated with similarity with other diseases and with the dataset imbalance, pear slug damage represents the majority class. However, generally speaking, from the obtained results, we can infer that classifiers get confused when faced with multiple classes of similar shape. Indeed, infected leaf images at different stages or against different backgrounds may also lead to the high complexity of the patterns that are displayed in the same class, which results in lower performance [41].

Less accurate results were obtained for the severity estimation, despite the ranking has remained unchanged compared to the first task. As for biotic stress, there are no particular differences in the two versions of the VGG network, both of which have reached an accuracy of 64.23% (VGG-16) and 65.93% (VGG-19). Moreover, for this task, the ResNet50 network scored the lowest (52.71%). InceptionV3, MobileNetV2, and EfficientNetB0 performed better with an accuracy of 74.07%, 73.56%, and 78.31%, respectively. EfficientNetB0 proved to be the most robust model in identifying biotic stress (90.18%) and the level of severity (78.31%) on par with InceptionV3 and MobileNetV2.

It is evident that the estimation of severity is a more challenging problem. The decline in performance for the resolution of this task corroborates with the experiments of Esgario et al. [25], which record a lower accuracy for the classification of stress. Looking at the confusion matrix for the EfficientNetB0 and MobileNetV2 networks in Figure 8, it can be seen that the models do not present particular difficulties in separating the low and medium classes. More misclassifications occur for the no risk class because it probably represents the minority class. Considering the very low class, EfficientNetB0 gets better classifications, while MobileNetV2 tends to classify it with the low class. This may be due to the fact that the symptoms at this level are mild and small. Due to the size of the image, models

TABLE 3: Test results obtained with different CNNs architectures.

Category	Architecture	Accuracy
Biotic stress	VGG16	81.12
	VGG19	83.91
	ResNet50	80.51
	InceptionV3	90.68
	MobileNetV2	90.01
	EfficientNetB0	90.18
Severity	VGG16	64.23
	VGG19	65.93
	ResNet50	52.71
	InceptionV3	74.07
	MobileNetV2	73.56
	EfficientNetB0	78.31

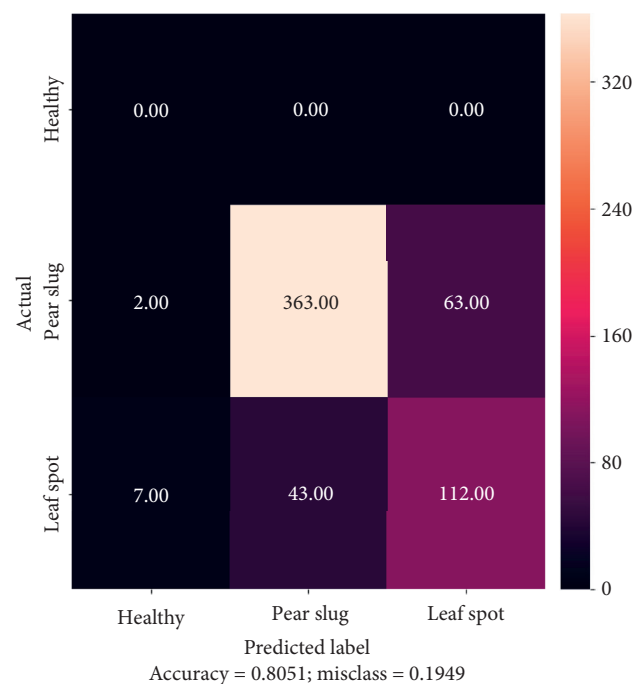


FIGURE 7: Biotic stress confusion matrix of ResNet50.

may have difficulty capturing relevant features. An inverse behavior is seen in the estimation of the high class, where MobileNetV2 makes less serious errors than EfficientNetB0 as this class is more confused with the medium class. Similarly, comparing MobileNetV2 with InceptionV3, the latter achieves a better result in recognizing the medium and high classes. Although the three models have made considerable errors, we note that these errors are located in the main diagonal, so some of them can be considered minor.

The ranking of the models is further confirmed by the computational performance or the time required for

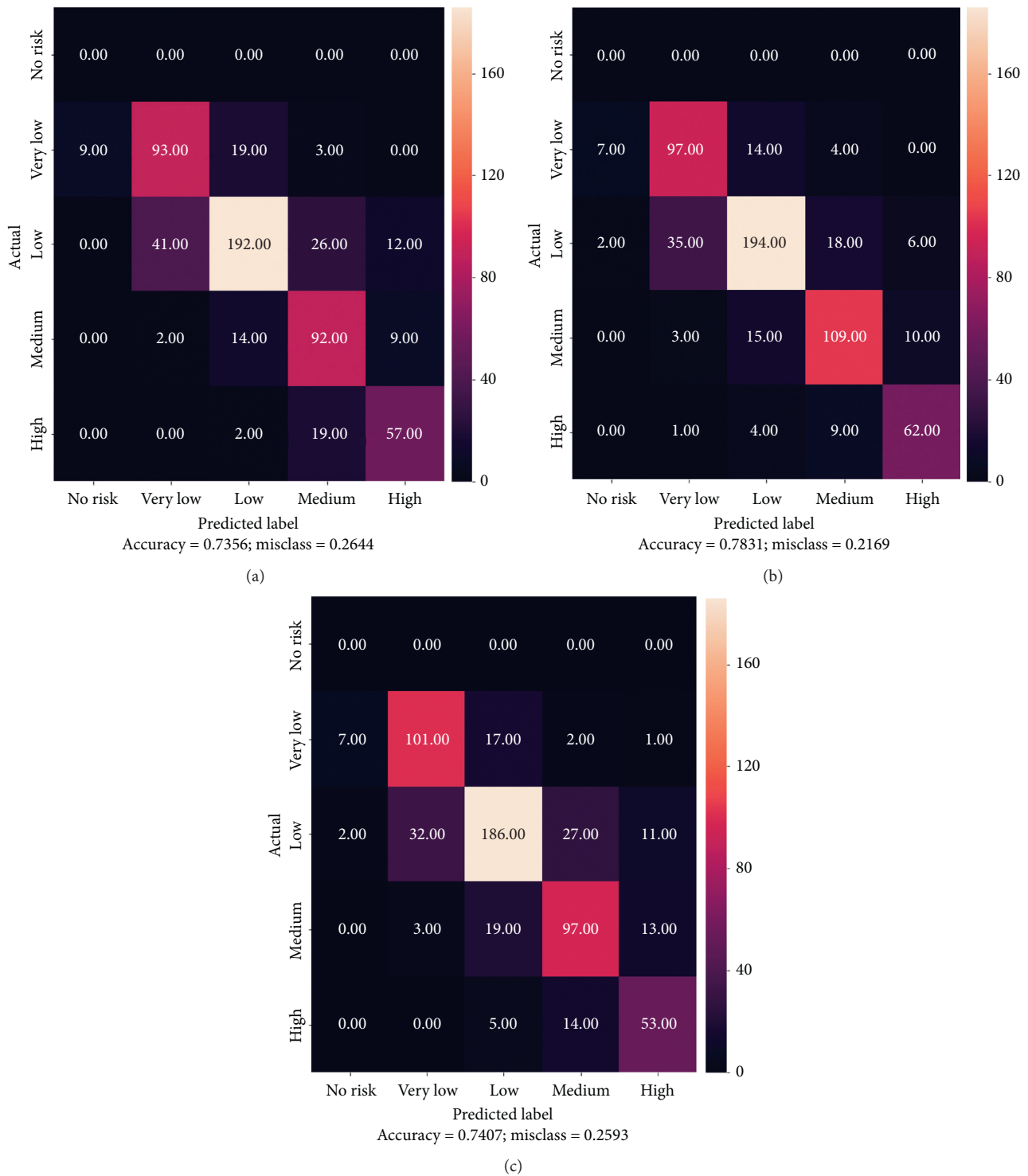


FIGURE 8: Severity confusion matrix. MobileNetV2 (a), EfficientNetB0 (b), and InceptionV3 (c).

training. As expected, given the nature of networks specifically designed to make the most of resources, EfficientNetB0 and MobileNetV2 required shorter time frames, followed by InceptionV3, ResNet50, VGG-19, and

VGG-16. Table 4 details the training time required for each model.

Finally, we can conclude that the EfficientNetB0 performed better since it presented a tradeoff between

TABLE 4: Comparison of the training time.

Architecture	Training time
VGG16	321.30
VGG19	320.78
ResNet50	320.58
InceptionV3	323.53
MobileNetV2	319.56
EfficientNetB0	316.03

computational cost, performance, and robustness in solving the problem with an imbalanced dataset.

4. Conclusions

The present work proposed an image-based multioutput convolutional neural network for biotic stress classification and severity estimation of pear tree diseases. The complete procedure was described, respectively, from gathering the pictures to image preprocessing and augmentation and to training and evaluation of deep networks. The deep learning approach based on multitask learning paradigm has proven its effectiveness in automatically extracting the discriminating features of diseased leaves using a shared architecture. Different CNN architectures were used in the experiments, whereas the network EfficientNetB0 was the one that achieved the best results, followed by InceptionV3 network.

To ensure a satisfactory generalization performance of the proposed model, a dataset of 3057 pear leaf images, called DiaMOS Plant, was collected in real-world conditions without any criteria to make it more representative and heterogeneous. Furthermore, pictures were gathered at different times of the year, from February to July, in order to capture the disease evolution from the first symptoms. A limitation of this work is related to the unbalanced data, which introduces a further level of complexity to the problem under examination. Indeed, misclassifications are found in the elements belonging to the minority class. However, the results obtained are consistent and confirm the robustness of the model in predicting three biotic stresses and five levels of severity in nonoptimal conditions.

As a future line of research, we foresee an extension of the current dataset to balance the classes and to enrich its representativeness with more biotic stresses, in order to train better models. Furthermore, based on the work done, we will integrate the diagnosis model into our Decision Support System called LANDS DSS [4] to recognize biotic stresses and gravity in real time through a mobile application. This device will assist farmers (regardless of experience level) in the rapid recognition of foliar symptoms as well as in the decision-making process for the use of pesticides.

Data Availability

The dataset used to support the findings of this study are available from the corresponding author upon request. The source code is available at <https://github.com/mallici-Francesca/leaf-disease-toolbox>.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

Francesca Maridina Mallocci gratefully acknowledges the Department of Mathematics and Computer Science of the University of Cagliari for the financial support of her Ph.D. scholarship.

References

- [1] Food and Agriculture Organization of the United Nations, *Plant Health and Food Security*, International Plant Protection Convention, Rome, Italy, 2017.
- [2] G. Fenu and F. M. Mallocci, "Forecasting plant and crop disease: an explorative study on current algorithms," *Big Data and Cognitive Computing*, vol. 5, no. 1, p. 2, 2021.
- [3] C. A. Harvey, Z. L. Rakotobe, N. Rao et al., "Extreme vulnerability of smallholder farmers to agricultural risks and climate change in Madagascar," *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 369, no. 1639, Article ID 20130089, 2014.
- [4] G. Fenu and F. M. Mallocci, "Lands DSS: a decision support system for forecasting crop disease in Southern Sardinia," *International Journal of Decision Support System Technology*, vol. 13, no. 1, pp. 21–33, 2021.
- [5] K. Dehnen-Schmutz, G. L. Foster, L. Owen, and S. Persello, "Exploring the role of smartphone technology for citizen science in agriculture," *Agronomy for Sustainable Development*, vol. 36, no. 2, p. 25, 2016.
- [6] G. Fenu and F. M. Mallocci, "Artificial intelligence technique in crop disease forecasting: a case study on potato late blight prediction," in *Smart Innovation Systems and Technologies*, pp. 79–89, Springer, Singapore, 2020.
- [7] G. Fenu and F. M. Mallocci, "An application of machine learning technique in forecasting crop disease," in *Proceedings of the 2019 3rd International Conference on Big Data Research*, Paris, France, November 2019.
- [8] P. B. Padol and A. A. Yadav, "SVM classifier based grape leaf disease detection," in *Proceedings of the 2016 Conference on Advances in Signal Processing (CASP)*, Pune, India, June 2016.
- [9] F. Qin, D. Liu, B. Sun, L. Ruan, Z. Ma, and H. Wang, "Identification of alfalfa leaf diseases using image recognition technology," *PLoS One*, vol. 11, no. 12, Article ID e0168274, 2016.
- [10] M. Islam, A. Dinh, K. Wahid, and P. Bhowmik, "Detection of potato diseases using image segmentation and multiclass support vector machine," in *Proceedings of the 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE)*, Windsor, Canada, April 2017.
- [11] S. Sladojevic, M. Arsenovic, A. Anderla, D. Culibrk, and D. Stefanovic, "Deep neural networks based recognition of plant diseases by leaf image classification," *Computational Intelligence and Neuroscience*, vol. 2016, Article ID 3289801, 11 pages, 2016.
- [12] B. Liu, Y. Zhang, D. He, and Y. Li, "Identification of apple leaf diseases based on deep convolutional neural networks," *Symmetry*, vol. 10, no. 1, p. 11, 2018.
- [13] Q. Yan, B. Yang, W. Wang, B. Wang, P. Chen, and J. Zhang, "Apple leaf diseases recognition based on an improved

- convolutional neural network,” *Sensors*, vol. 20, no. 12, p. 3535, 2020.
- [14] I. Ahmad, M. Hamid, S. Yousaf, S. T. Shah, and M. O. Ahmad, “Optimizing pretrained convolutional neural networks for tomato leaf disease detection,” *Complexity*, vol. 2020, Article ID 8812019, 6 pages, 2020.
- [15] S. P. Mohanty, D. P. Hughes, and M. Salathé, “Using deep learning for image-based plant disease detection,” *Frontiers in Plant Science*, vol. 7, p. 1419, 2016.
- [16] D. Radovanovic and S. Dukanovic, “Image-based plant disease detection: a comparison of deep learning and classical machine learning algorithms,” in *Proceedings of the 24th International Conference on Information Technology (IT)*, Zabljak, Montenegro, February 2020.
- [17] J. Kranz, “Measuring plant disease,” in *Experimental Techniques in Plant Disease Epidemiology*, pp. 35–50, Springer, Berlin, Germany, 1988.
- [18] C. H. Bock, G. H. Poole, P. E. Parker, and T. R. Gottwald, “Plant disease severity estimated visually, by digital photography and image analysis, and by hyperspectral imaging,” *Critical Reviews in Plant Sciences*, vol. 29, no. 2, pp. 59–107, 2010.
- [19] G. Wang, Y. Sun, and J. Wang, “Automatic image-based plant disease severity estimation using deep learning,” *Computational Intelligence and Neuroscience*, vol. 2017, Article ID 2917536, 8 pages, 2017.
- [20] J. G. Arnal Barbedo, “Plant disease identification from individual lesions and spots using deep learning,” *Biosystems Engineering*, vol. 180, pp. 96–107, 2019.
- [21] P. Wspanialy and M. Moussa, “A detection and severity estimation system for generic diseases of tomato greenhouse plants,” *Computers and Electronics in Agriculture*, vol. 178, p. 105701, 2020.
- [22] M. Prabhakar, R. Purushothaman, and D. P. Awasthi, “Deep learning based assessment of disease severity for early blight in tomato crop,” *Multimedia Tools and Applications*, vol. 79, no. 39–40, pp. 28773–28784, 2020.
- [23] S. Ghosal, D. Blystone, A. K. Singh, B. Ganapathysubramanian, A. Singh, and S. Sarkar, “An explainable deep machine vision framework for plant stress phenotyping,” in *Proceedings of the National Academy of Sciences*, vol. 115, no. 18, pp. 4613–4618, 2018.
- [24] Q. Liang, S. Xiang, Y. Hu, G. Coppola, D. Zhang, and W. Sun, “PD²SE-net: computer-assisted plant disease diagnosis and severity estimation network,” *Computers and Electronics in Agriculture*, vol. 157, pp. 518–529, 2019.
- [25] J. G. Esgario, R. A. Krohling, and J. A. Ventura, “Deep learning for classification and severity estimation of coffee leaf biotic stress,” *Computers and Electronics in Agriculture*, vol. 169, pp. 105–162, 2020.
- [26] D. P. Hughes and M. Salathé, “An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing,” 2015, <http://arxiv.org/abs/1511.08060>.
- [27] F. Chollet, *Deep Learning with Python*, Manning Publications, Shelter Island, NY, USA, 2017.
- [28] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, <http://arxiv.org/abs/1409.0575>.
- [29] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.
- [30] C. Szegedy, W. Liu, Y. Jia et al., “Going deeper with convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, Boston, MA, USA, June 2015.
- [31] A. G. Howard, M. Zhu, B. Chen et al., “Mobilenets: efficient convolutional neural networks for mobile vision applications,” 2017, <http://arxiv.org/abs/1704.04861>.
- [32] M. Tan and Q. V. Le, “Efficientnet: rethinking model scaling for convolutional neural networks,” 2019, <http://arxiv.org/abs/1905.11946>.
- [33] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanekharan, “Using deep transfer learning for image-based plant disease identification,” *Computers and Electronics in Agriculture*, vol. 173, p. 105393, 2020.
- [34] S. Coulibaly, B. Kamsu-Foguem, D. Kamissoko, and D. Traore, “Deep neural networks with transfer learning in millet crop images,” *Computers in Industry*, vol. 108, pp. 115–120, 2019.
- [35] D. Xu, Y. Shi, I. W. Tsang, Y.-S. Ong, C. Gong, and X. Shen, “Survey on multi-output learning,” *IEEE transactions on neural networks and learning systems*, vol. 31, no. 7, pp. 2409–2429, 2019.
- [36] M. Lin, Q. Chen, and S. Yan, “Network in network,” 2013, <http://arxiv.org/abs/1312.4400>.
- [37] S. Ioffe and C. Szegedy, “Batch normalization: accelerating deep network training by reducing internal covariate shift,” in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 448–456, Lille, France, July 2015.
- [38] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [39] J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei, “Imagenet: a large-scale hierarchical image database,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, Miami, FL, USA, June 2009.
- [40] Y. Yoshida and M. Okada, “Data-dependence of plateau phenomenon in learning with neural network—statistical mechanical analysis,” 2019, <http://arxiv.org/abs/2001.03371v1>.
- [41] P. Jiang, Y. Chen, B. Liu, D. He, and C. Liang, “Real-time detection of apple leaf diseases using deep learning approach based on improved convolutional neural networks,” *IEEE Access*, vol. 7, pp. 59069–59080, 2019.