

Article

# Robust Multi-Sensor Consensus Plant Disease Detection Using the Choquet Integral <sup>†</sup>

Cedric Marco-Detchart <sup>1</sup>, Carlos Carrascosa <sup>1</sup>, Vicente Julian <sup>1,2,\*</sup> and Jaime Rincon <sup>1</sup>

<sup>1</sup> Valencian Research Institute for Artificial Intelligence, Universitat Politècnica de València, Camí de Vera s/n, 46022 Valencia, Spain

<sup>2</sup> Valencian Graduate School and Research Network of Artificial Intelligence, Universitat Politècnica de València, Camí de Vera s/n, 46022 Valencia, Spain

\* Correspondence: vjulian@upv.es

<sup>†</sup> This paper is an extended version of our paper “Plant Disease Detection: An Edge-AI Proposal”. In Proceedings of the Conference Practical Applications of Agents and Multi-Agent Systems in 2022, L’Aquila, Italy, 13–15 July 2022.

**Abstract:** Over the last few years, several studies have appeared that employ Artificial Intelligence (AI) techniques to improve sustainable development in the agricultural sector. Specifically, these intelligent techniques provide mechanisms and procedures to facilitate decision-making in the agri-food industry. One of the application areas has been the automatic detection of plant diseases. These techniques, mainly based on deep learning models, allow for analysing and classifying plants to determine possible diseases facilitating early detection and thus preventing the propagation of the disease. In this way, this paper proposes an Edge-AI device that incorporates the necessary hardware and software components for automatically detecting plant diseases from a set of images of a plant leaf. In this way, the main goal of this work is to design an autonomous device that allows the detection of possible diseases that can detect potential diseases in plants. This will be achieved by capturing multiple images of the leaves and implementing data fusion techniques to enhance the classification process and improve its robustness. Several tests have been carried out to determine that the use of this device significantly increases the robustness of the classification responses to possible plant diseases.



**Citation:** Marco-Detchart, C.; Carrascosa, C.; Julian, V.; Rincon, J. Robust Multi-Sensor Consensus Plant Disease Detection Using the Choquet Integral. *Sensors* **2023**, *23*, 2382. <https://doi.org/10.3390/s23052382>

Academic Editors: Javier Prieto and Ramón J. Durán Barroso

Received: 6 February 2023

Revised: 19 February 2023

Accepted: 20 February 2023

Published: 21 February 2023



**Copyright:** © 2023 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/>).

**Keywords:** smart agriculture; machine learning; EDGE-AI; sensors

## 1. Introduction

The agri-food sector has always played a fundamental role in our society. It is essential because it supplies us with enough food to satisfy the nutritional needs of a constantly growing population. It is strategic because its activity has a significant impact in terms of economic production, employment generation, environmental management, and the maintenance of a living and balanced territory.

The availability of data on a farm is essential so that it can be analysed to generate information and knowledge to support producers in their decision-making and management of their operations. This includes using agricultural inputs more precisely, the ability to foresee the appearance of diseases, pests, or meteorological phenomena and thus adopt adaptive measures to reduce risks, the automation of tasks, and, therefore, a more efficient administration.

In this sense, techniques based on Artificial Intelligence (AI) have emerged as a valuable tool by providing mechanisms and procedures to facilitate the decision-making of specific tasks in the agri-food sector. Over the last few years, different approaches have tried to provide AI techniques for sustainable development in the farming sector, especially machine learning techniques. A deeper analysis and further reviews are presented in [1–3].

Another of the factors that most influence crop yields is the possible incidence of pests and diseases. To reduce their impact, farmers make regular use of phytosanitary products. In this way, the development of potentially dangerous populations can be controlled, thus ensuring production. This group also includes herbicides, which prevent competition for nutrients, water and the establishment of the main crops with other unwanted plants.

In many cases, given the uncertainty of when the pest will appear and how aggressive it will be, farmers often carry out preventive treatments. Over the last few years, the cost of these treatments has shown a clear upward trend; for example, in the EU, sales of active substances qualified as pesticides used in plant protection products exceed 350,000 t per year. Reducing the use of chemical pesticides by half by 2030 and the use of the most hazardous pesticides by 50% is one of the main objectives of the European Green Deal ([https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal\\_en](https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal_en) accessed on 5 February 2023).

Early detection of plant diseases through manual and visual inspections by experts or farmers has its limitations, including the dependency on a limited group of individuals and the potential for error due to the variety of plants and multiple diseases that can affect them. The automation of disease detection through the use of Artificial Intelligence techniques, specifically deep learning, offers numerous benefits [4]. Early treatment, as a result of early detection, reduces the need for chemical products and results in cost savings, preventing production losses, and contributing to environmental sustainability by avoiding the use of harmful phytosanitary products in the long term. The manual approach is also time-consuming and prone to human error, whereas AI automation offers a more efficient and reliable solution.

According to this, the use of Artificial Intelligence, particularly deep learning techniques, in plant disease detection has gained widespread popularity in recent years [5,6]. These approaches analyse and categorise plants to identify potential problems. Satellite and hyperspectral imaging is commonly utilised in agricultural analysis and plant disease detection. Satellite images provide a comprehensive view of the land and crop performance, whereas hyperspectral images offer a view beyond the visible spectrum, allowing for the use of tools such as the NDVI index to measure greenness and detect crop issues [7]. The main drawback of these approaches is the high cost of equipment (cameras and satellites) and processing of large images.

Another possible approach is to use closer images, such as leaves or sections of plants, to be analysed and classified to determine possible diseases [8,9]. Most of the analysed proposals in this line offer cloud services to perform detection. One of the problems we may encounter is the lack of connectivity in some rural regions and the need to transfer large amounts of data to perform the classification process in these services offered in the cloud. In this sense, the use of devices based on edge computing that detect diseases without the need for connections to cloud services and avoids continuous transfers of images over the network may be of greater interest.

Therefore, this paper presents an EDGE device that incorporates the necessary hardware and software components to automatically detect plant diseases from a set of images of the plant leaf. The device can be easily incorporated into an agricultural robot, a drone, or a tractor to facilitate automatic image acquisition in a crop field. Furthermore, the use of a set of images simultaneously, instead of just one, increases the robustness of the classifications, as demonstrated in the tests performed. This paper is an extended version of a previous paper published in the conference proceedings “Practical Applications of Agents and Multi-Agent Systems” [10] where a new device was developed and new designs and evaluations of the proposed solution were carried out.

The rest of the paper is structured as follows. Section 2 analyses previous related works, Section 3 presents the description of the proposed system, Section 4 describes the experiments carried out, and, finally, some conclusions are presented in Section 5.

## 2. Related Work

One of the main approaches to automatically detect plant diseases is through image analysis. This analysis may focus on different features such as geometry or colour. In some specific kinds of images, other indexes are also commonly used. So, for hyperspectral images, the NDVI index, which measures the level of green on images, is used. On the other hand, for visible range images, there are other alternative indexes such as the VARI index or the vNDVI index [11].

Identifying plant diseases automatically poses several challenges, as outlined by the review proposed in [12]. These challenges range from issues during the capture process, such as noise or fog over the camera, to the presence of unwanted information in the images, such as background, soil, or other plants. One way to deal with some of these problems is the pre-processing of images to not only eliminate spurious information, e.g., background segmentation or texture removal (smoothing [13,14]) or even image improvement (e.g., contrast enhancement [15]).

Apart from material-specific issues that may arise during image capture, another critical challenge is the potential existence of multiple diseases in a single plant. After image processing, automatic detection of plant diseases involves a classification task that can be approached using two main methods. The first involves classical Machine Learning (ML) techniques, where a set of features is extracted and chosen from the images, and then classified using techniques such as Support Vector Machines (SVM) [16,17], K-Means algorithm [18], or Random Forest [19], among others. These techniques need a very precise human-made solution (ground truth) and assistance to be performant. Furthermore, they must work well when there is a limited amount of data. Second, and currently used, one of the most popular approaches is the use of Deep Learning (DL) [20] and particularly Convolutional Neural Networks (CNN) [21,22] to train a model to identify the dataset classes. As is well known, even if there are increasingly more available images to work with, the quantity/quality is limited to learning from in a specific task. In those cases, Transfer Learning is used to build a network based on pre-trained information and adapted to the concerned task. These networks are pre-trained on large datasets, e.g., ImageNet [23]. This process takes the first layer of the trained network and removes the last layers adapting them to the specific task and training only these last steps. In this way, the specific task is not trained from scratch, and the computing time is shortened. The most common and efficient networks in the literature are *Alexnet* [24], *ResNet50* [25], *VGG16* [26], and *Inception V3* [27]. *EfficientNet* [28] can also be considered a group of networks, as there are eight types of subnets.

An alternative approach to mostly used network architecture is that of Capsule Networks [29] which fixes one of the significant drawbacks of standard CNN. CNNs do not consider the possible feature hierarchy in an image considering similar images as equal even when they are not. In the work presented by Samin et al. [30] the Capsule Network approach is used without using Transfer Learning, obtaining an accuracy of 93.07%.

More recently, a lightweight CNN approach based on Inception and Residual connection [31]. The proposed approach extracts better features from the input images. This is performed by replacing the standard convolution by a depth-wise separable convolution combined with a point-wise convolution, which results in fewer parameters as well as a speed-up in the processing. The resulting performance with the approach presented is 99.39% accuracy.

After studying the state-of-the-art, it can be seen that there is currently a multitude of proposals, most of them based on deep learning techniques that offer promising results from the existing datasets. However, there are specific gaps that we think should be analysed. On the one hand, some works suggest the need for image pre-processing before classification; in our opinion, this aspect should be studied in greater detail as it may allow for an improvement in the classification process. On the other hand, most of the works are evaluated against a so-called ideal dataset. Using more realistic datasets to validate existing models would allow for analysis of their possible robustness. Nevertheless, Ref. [32] is an

interesting approach, but as they are working with infrared images, they would need to use infrared cameras when applied to the real world, which is expensive to deploy.

Apart from dealing with the above-mentioned gaps, our objective is to build a robust model capable of being deployed on an edge platform. In fact, our system, to be presented in the next section, focuses not only on the development of the software but also on the hardware infrastructure to give support to it.

### 3. System Description

In this section, the operation of the plant disease classification system using an EDGE device is explained in detail. The different software and hardware tools employed are also described. The proposed approach is shown in Figure 1.

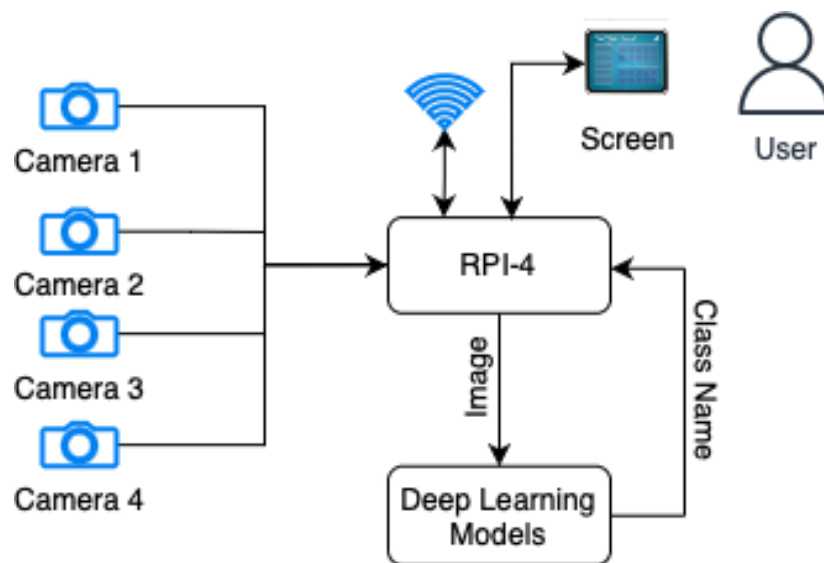
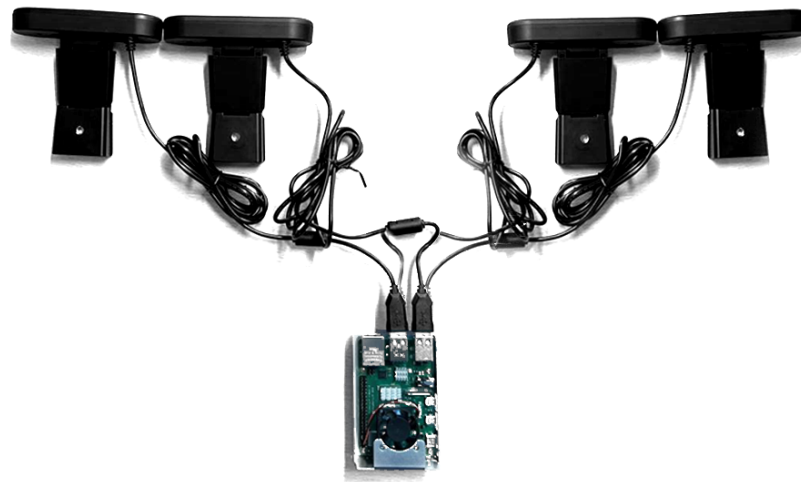


Figure 1. Description of the system.

The main components integrated into this prototype are the machine vision module, which is composed of four webcams (see Figure 2). We have the data processing module, which receives the four images from the cameras and reconstructs them to create an image composed of four images. The next element integrates the classification models to determine the plant's disease. The system utilizes a WiFi communication system to send the classified data to the cloud, as well as a visualization system through an LCD screen. Further details regarding the hardware components and classification models are outlined in the subsequent sections. At a high level, the EDGE system employs cameras to capture four images from distinct angles, thus acquiring additional information and mitigating potential blind spots. The classification models are designed to identify whether the plant exhibits any of the 38 diseases present in the training database.

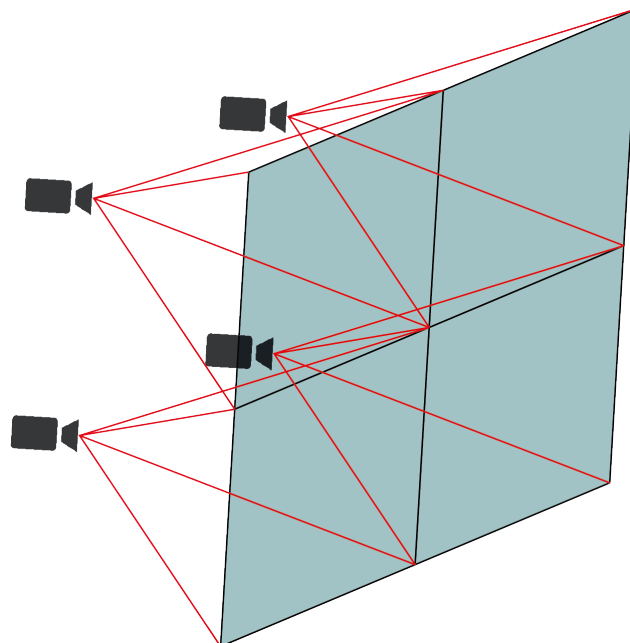


**Figure 2.** Hardware configuration.

### 3.1. Hardware Description

This section describes the hardware used for plant disease recognition using a Raspberry 4. The Raspberry Pi 4 development system is a Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) 64-bit 1.5GHz SoC. We used an RPI-4 with 8GB SDRAM, IEEE 802.11ac wireless protocol, and Bluetooth 5.0, BLE for our experiments. It integrates four USB ports: two USB 3.0 and two USB 2.0.

With this hardware configuration, it is possible to run trained TensorFlow lite models. To capture the images, we have used four 3-megapixel Logitech cameras with a resolution of 720p and a field of view (FOV) of 60 degrees. These cameras are spaced so that the FOVs of each bed overlap at a minimum (Figure 3).



**Figure 3.** Array of cameras pointing at different views of the scene.

Each of these images is resized to a size of  $224 \times 224$ . The models then analyse each of these images to determine what type of disease the plant has. The disease classification process will be described below.

The advantages of the proposed system are its low construction and maintenance costs. At the same time, it is a portable system of low weight, ideal for being integrated into

either an unmanned aerial vehicle (UAV) or an unmanned ground vehicle (UGV). These can autonomously roam the field and report via GPS where the plants with the diseases are located. This would save time and money for the farmer; because they are completely autonomous, these systems can be programmed to start at any time of the day.

### 3.2. Software Description

In the following section, we will describe the different software tools used. The system proposed in this work uses deep learning techniques using a MobileNet v2 network for plant disease classification. This system is embedded in a Raspberry Pi 4, which integrates the trained ML model.

The MobileNet v2 [33] architecture, one of the most widely used neural networks in mobile applications, is based on an inverted residual structure in which the input and output of the residual block are thin bottleneck layers. Unlike other models with these features, MobileNet v2 employs lightweight convolutions in depth to filter features in the intermediate expansion layer. Additionally, to preserve its representational power, the network has eliminated nonlinearities in the narrow layers. Research presented by Kristiani et al. [34] showed from the experiments that Mobilenet outperforms Inception in terms of speed, accuracy and file size. The speed in Inception V3 is 9 frames per second, while that value in Mobilenet is 24 frames per second.

The structure of the compiled model is depicted in Table 1, and it remains consistent across all the configurations (Table 2) utilised in this study. Once the training process is complete, the model is stored as a \*.tflite file and embedded in the Raspberry Pi 4. In the event of a classification improvement or the addition of a new disease class, this model can be effortlessly updated.

**Table 1.** Main characteristics of the employed model.

| Layer (Type)             | Param     | Output Shape# |
|--------------------------|-----------|---------------|
| keras_layer (KerasLayer) | 2,257,984 | (None, 1280)  |
| flatten (Flatten)        | 0         | (None, 1280)  |
| dense (Dense)            | 655,872   | (None, 512)   |
| dropout (Dropout)        | 0         | (None, 512)   |
| dense_1 (Dense)          | 19,494    | (None, 38)    |

The four cameras of the prototype capture four different images of the plant to be analysed. These four images are preprocessed to make them compatible with the trained model. This preprocessing consists of resizing the images from  $1280 \times 720$  to  $224 \times 224$ . Each newly resized image is used as input for the classification model. In this way, the system can analyse the plant from four different points of view. These four images are used by the model one at a time. This result in 38 probabilities that are added to a  $38 \times 4$  matrix (38 classes and four cameras). This matrix is then used as input for the data fusion algorithm to obtain the final probabilities, which is explained in the next section. We used two datasets to train and validate the model: the PlantVillage and the extended PlantVillage. The PlantVillage contains leaf images with a homogeneous background. However, the extended PlantVillage is formed by the original images but with a synthetic background that simulates a field to add noise to the image and make it more realistic. This test aims to check the model's robustness and to see if the model performs well against this more realistic dataset.

## 4. Experimental Setup

In this Section, we analyse the performance of the different configurations used. First, in Section 4.1, we present the data set used for our experiments and the measures used to quantify the results obtained. Second, in Section 4.2, we present the quantitative results of our experiments.



In the first round of experiments, we test two well-known mobile-oriented device networks. Then we select the best performer network and tested it with our consensus classification approach. The experiment employed the Mobilenet V2 and NasnetMobile networks, and several hyperparameters were established to train both networks, while some hyperparameters remained constant across all experiments, others were altered to determine the optimal settings for the best results. To avoid overfitting, a maximum of seven epochs was selected. The learning rate for all models remained identical, and data augmentation and fine-tuning were activated or deactivated, depending on the specific experiment. Table 2 illustrates the hyperparameter configurations.

To be able to treat the four-grid system, we propose using an aggregation procedure to summarise the data captured by the setup. Every set of images captured by the cameras is considered a batch where the corresponding trained network is applied to each one. Then the output of each image is aggregated to produce one result that better represents all of the initial data. To proceed with the data fusion, we need some definitions:

**Definition 1** ([35]). A mapping  $M : [0, 1]^n \rightarrow [0, 1]$  is an aggregation function if it is monotone non-decreasing in each of its components and satisfies the boundary conditions,  $M(\mathbf{0}) = M(0, 0, \dots, 0) = 0$  and  $M(\mathbf{1}) = M(1, 1, \dots, 1) = 1$ .

**Definition 2.** A function  $m : 2^N \rightarrow [0, 1]$  is a fuzzy measure if, for all  $X, Y \subseteq N$ , it satisfies the following properties:

1. Increasingness: if  $X \subseteq Y$ , then  $m(X) \leq m(Y)$ ;
2. Boundary conditions:  $m(\emptyset) = 0$  and  $m(N) = 1$ .

An example of a commonly used fuzzy measure is the power measure, which we use in this work:

$$m_q(X) = \left(\frac{|X|}{n}\right)^q, \text{ with } q > 0, \quad (1)$$

where  $|X|$  is the number of elements to be aggregated,  $n$  the total number of elements and  $q > 0$ . We have selected this measure due to the performance obtained in terms of accuracy in classification problems [36,37].

**Definition 3** ([35]). Let  $m : 2^N \rightarrow [0, 1]$  be a fuzzy measure. The discrete Choquet integral of  $\mathbf{x} = (x_1, \dots, x_n) \in [0, 1]^n$  with respect to  $m$  is defined as the function  $C_m : [0, 1]^n \rightarrow [0, 1]$ , given by

$$C_m(\mathbf{x}) = \sum_{i=1}^n (x_{(i)} - x_{(i-1)}) \cdot m(A_{(i)}),$$

where  $(x_{(1)}, \dots, x_{(n)})$  is an increasing permutation on the input  $\mathbf{x}$ , that is,  $x_{(1)} \leq \dots \leq x_{(n)}$ , with the convention that  $x_{(0)} = 0$ , and  $A_{(i)} = \{(i), \dots, (n)\}$  is the subset of indices of the  $n - i + 1$  largest components of  $\mathbf{x}$ .

The Choquet integral is idempotent and presents an averaging behaviour. Observe that the Choquet integral is defined using a fuzzy measure, permitting to consider the relation between the elements considered to be aggregated (i.e., the components of an input  $\mathbf{x}$ ).

In our experiments, after applying each trained model to the images, the output shows the membership of each image to the 38 classes of the dataset. Then, we aggregate ordering increasingly for each class probability output and apply the Choquet integral to obtain the final membership probability for each class. Finally, we take the maximum response to select the class of the image batch.

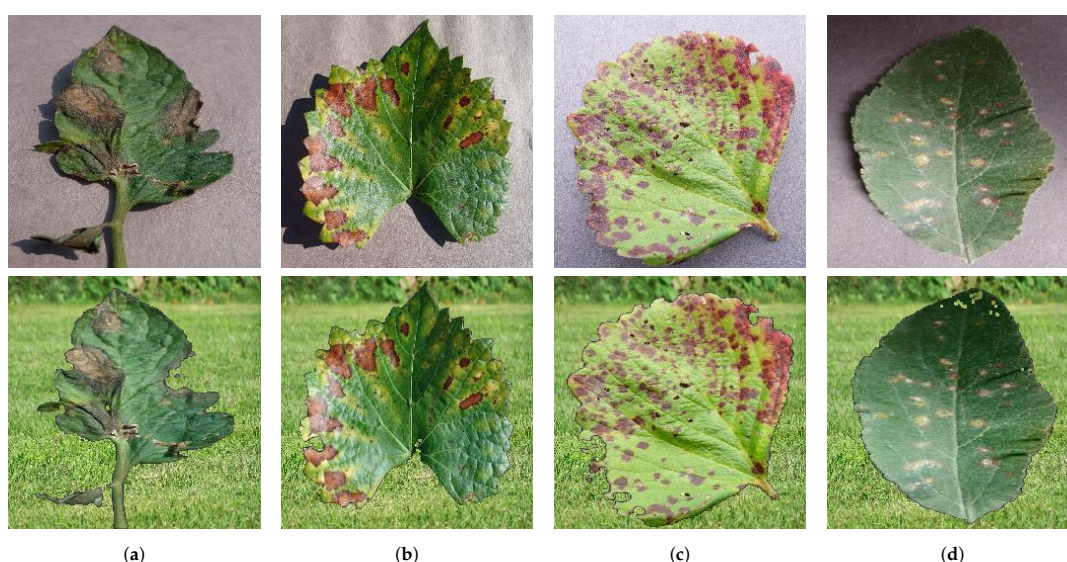
**Table 2.** Hyperparameters used to configure the neural net used in the experiments.

| #              | Net Type     | N-Epochs | Learning Rate | Transfer Learning | Data Augmentation | Data Set  |
|----------------|--------------|----------|---------------|-------------------|-------------------|-----------|
| S <sub>1</sub> | Mobilenet V2 | 7        | 0.001         | ✗                 | ✗                 | Raw Image |
| S <sub>2</sub> | Mobilenet V2 | 7        | 0.001         | ✓                 | ✗                 | Raw Image |
| S <sub>3</sub> | Mobilenet V2 | 7        | 0.001         | ✗                 | ✓                 | Raw Image |
| S <sub>4</sub> | Mobilenet V2 | 7        | 0.001         | ✓                 | ✓                 | Raw Image |
| S <sub>5</sub> | NasNetMobile | 7        | 0.001         | ✗                 | ✗                 | Raw Image |
| S <sub>6</sub> | NasNetMobile | 7        | 0.001         | ✓                 | ✗                 | Raw Image |
| S <sub>7</sub> | NasNetMobile | 7        | 0.001         | ✗                 | ✓                 | Raw Image |
| S <sub>8</sub> | NasNetMobile | 7        | 0.001         | ✓                 | ✓                 | Raw Image |

#### 4.1. Dataset and Quantification of the Results

In this work, we use two datasets, one to train and validate our model and a derived one where some modifications are added to the first one to simulate the real-environment process of the proposed approach. This study is in an early stage, so all the processes have been performed in a controlled environment with already captured images. In a future second stage, the objective is to test our proposed system in a real environment.

The primary dataset used in this study is known as PlantVillage (PV) [38], comprising roughly 87,000 RGB images of healthy and diseased crop leaves, which are categorized into 38 distinct classes. The images are captured from individual leaves of each plant and disease against a consistent background. To facilitate training, the dataset was split into three subsets, with an 80/20 ratio: 80% for training, 10% for testing, and the remaining 10% for validation. Raw images (Figure 4) were utilized for model training and validation without undergoing any image pre-processing.



**Figure 4.** Example of images included in the PlantVillage dataset [38] from different plant diseases along with their equivalent in the Synthetic PlantVillage dataset. (a) Tomato—blight; (b) grape—esca; (c) strawberry—scorch; (d) apple—rust.

The second dataset is called SyntheticPlantVillage (SynPV). It is a modified version where the background of the images has been removed and filled with a grass background simulating a real scenario for the leaves. This second dataset is used to put our proposal



to the test analysing its robustness in a real-like scenario. We simulate the camera grid capture taking batches of the same class images and aggregate its classification to obtain a consensus decision about the final disease class.

To interpret the results obtained in the confusion matrix, we use the following Precision/Recall measures:

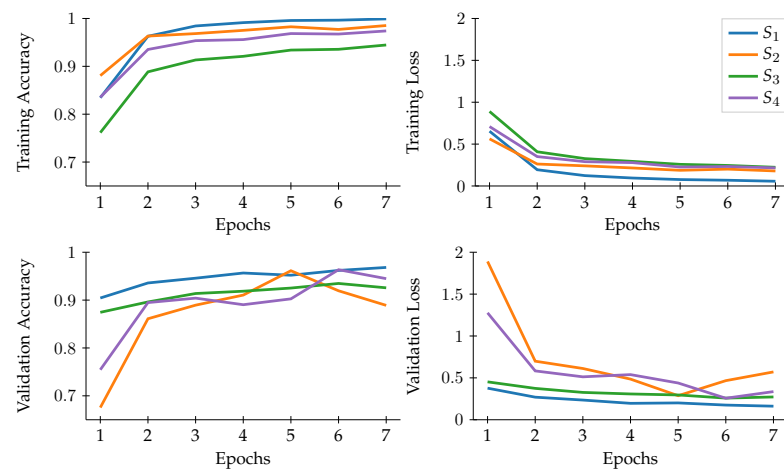
$$Prec = \frac{TP}{TP + FP}, Rec = \frac{TP}{TP + FN}, F_{\beta} = (1 + \beta^2) \frac{Prec \cdot Rec}{\beta^2 \cdot Prec + Rec}.$$

We select the values of  $\beta = 0.5$  and  $\beta = 1$  as it is the most commonly used in the literature.

#### 4.2. Experimental Results

In this section, we present the outcomes obtained with various configurations listed in Table 2. Quantitative results are shown in Table 3. Furthermore, we carried out real-world simulations in a controlled environment of our models using a Raspberry Pi 4 and a four-camera setup, as outlined in the proposed solution.

The behaviour of each configuration used can be seen in Figure 5, which depicts how they perform similarly across various epochs. During the validation phase, on the one hand,  $S_2$  and  $S_4$  exhibit an unpredictable pattern, starting with low accuracy, then suddenly increasing, fluctuating until the end of the training. On the other hand, the performance of  $S_1$  and  $S_3$  is more consistent, reaching stability by the end of the epochs. These trends are further reflected in the quantitative results obtained from the testing phase.



**Figure 5.** Training and validation accuracy and loss obtained with the different configurations from Table 2.

As shown in Table 3, the best results (validated on the training machine) are achieved with configuration  $S_1$ , without the use of transfer learning or data augmentation. The second-best outcome is produced by  $S_3$ , which only employs data augmentation. This observation suggests that the use of MobileNet v2 with transfer learning for the specific task of identifying plant disease does not enhance performance and, in fact, results in a decrease in all measures. The use of transfer learning has a negative impact on the results, as the weights learned from the ImageNet dataset, which has a large number of classes, affect the outcomes in  $S_2$  and  $S_4$ .

As we can observe, using our new proposal, with a four-grid camera to capture plant images, increases the system's scores. In general, all the configurations benefit from the new setup, and in the case of  $S_1$ , which remains the best performer, the result of  $F_1$  obtains a 98.2% performance. To prove the presented approach's benefits, we tested it over a SynPv dataset, measuring the approach's robustness. This second experiment shows that

comparing the results with SynPV using the initial approach with one camera, the best performer,  $S_1$ , decays to 64.4% but using the new setup increases considerably almost to an 80% performance in terms of  $F_1$ . These results indicate that our new approach increases the robustness of the system, even when artefacts and unwanted information are present in the image.

The model training and validation were performed on a machine equipped with an Intel i5-9500 processor running at 4.4 GHz, 16 GB of RAM, and operating on Ubuntu 20.04.4 LTS. For the actual validation of the model, a Broadcom BCM2835 ARM11 1 GHz with a VideoCore IV GPU and 512 MB of RAM was utilised.

**Table 3.** Resulting test performance of the model trained with the parameters in Table 2 over the original dataset PlantVillage (PV) and the synthetic one (SynPV), simulating grass on the background. Results are shown with the original proposal ( $Prec$ ,  $Rec$ ,  $F_{0.5}$ , and  $F_1$ ) and the new one using the array of cameras ( $Prec'$ ,  $Rec'$ ,  $F'_{0.5}$ , and  $F'_1$ ).

| #     | Dataset | $Prec$       | $Prec'$      | $Rec$        | $Rec'$       | $F_{0.5}$    | $F'_{0.5}$   | $F_1$        | $F'_1$       |
|-------|---------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| $S_1$ | PV      | <b>0.881</b> | <u>0.986</u> | <b>0.904</b> | <u>0.980</u> | <b>0.882</b> | <u>0.985</u> | <b>0.886</b> | <u>0.982</u> |
|       | SynPV   | <b>0.744</b> | <u>0.872</u> | <b>0.665</b> | <u>0.819</u> | <b>0.669</b> | <u>0.811</u> | <b>0.644</b> | <u>0.796</u> |
| $S_2$ | PV      | 0.829        | 0.927        | 0.852        | 0.945        | 0.813        | 0.923        | 0.807        | 0.923        |
|       | SynPV   | 0.616        | 0.676        | 0.465        | 0.553        | 0.456        | 0.546        | 0.423        | 0.510        |
| $S_3$ | PV      | <b>0.881</b> | <u>0.986</u> | 0.897        | 0.978        | 0.881        | 0.983        | 0.883        | 0.980        |
|       | SynPV   | 0.719        | 0.840        | 0.650        | 0.790        | 0.643        | 0.769        | 0.623        | 0.758        |
| $S_4$ | PV      | 0.852        | 0.947        | 0.862        | 0.940        | 0.836        | 0.932        | 0.831        | 0.927        |
|       | SynPV   | 0.658        | 0.487        | 0.375        | 0.383        | 0.419        | 0.368        | 0.358        | 0.337        |

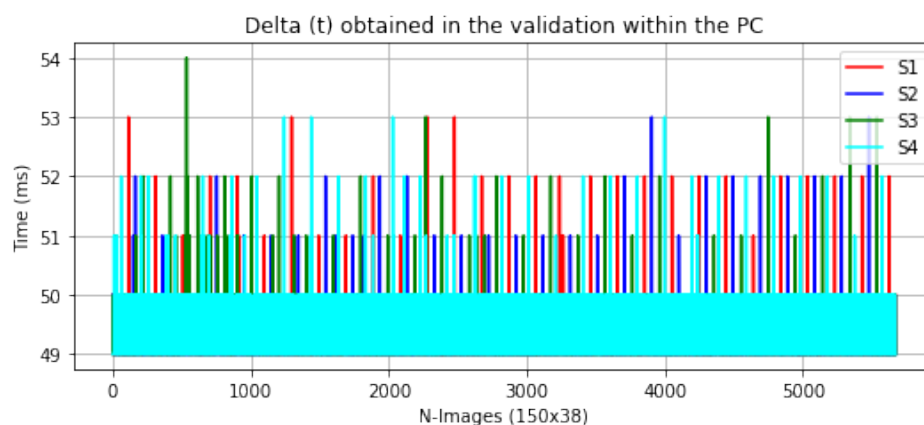
Table 4 presents the top-performing outcomes of the experiments conducted using the NasNetMobile neural network, which was validated with the plant village dataset (PV) and Synthetic PlantVillage (SynPV). Like the MobileNet, the NasNetMobile [39] is a convolutional neural network (CNN) model created to carry out image classification tasks on mobile devices. The NasNetMobile's architecture relies on a search algorithm that utilizes reinforcement learning techniques to identify the optimal network structure for a given task, thereby eliminating the need for manual tuning by developers. Similar to the MobileNet V2, transfer learning with the NasNetMobile network for identifying plant diseases does not enhance its performance; instead, it leads to a decrease in results. The use of transfer learning has an impact on the classification process, which may be attributed to the training of the ImageNet dataset and the types of images utilized to obtain the weights. As a result, the classification values are influenced in the experiments that leverage transfer learning ( $S_2$  and  $S_8$ ).

**Table 4.** Resulting test performance of the model trained with the parameters in Table 2 over the original dataset PlantVillage (PV) and the synthetic one (SynPV), simulating grass on the background. Results are shown with the original proposal using a single camera.

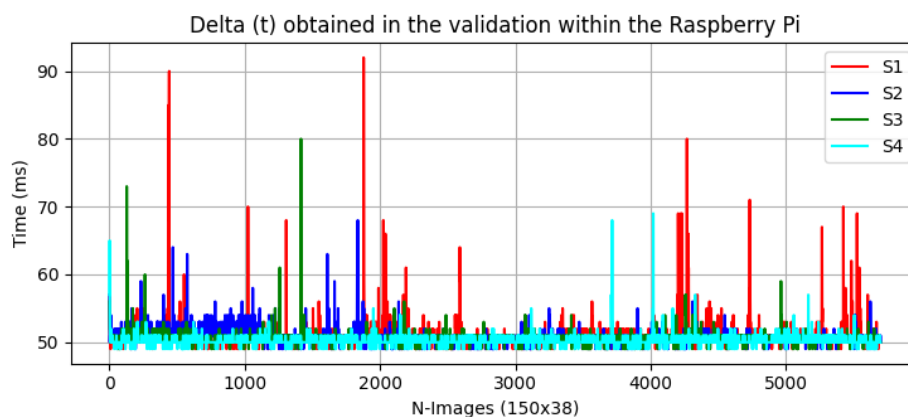
| #     | Dataset | <i>Prec</i> | <i>Rec</i> | $F_{0.5}$ | $F_1$ |
|-------|---------|-------------|------------|-----------|-------|
| $S_5$ | PV      | 0.813       | 0.851      | 0.812     | 0.816 |
|       | SynPV   | 0.630       | 0.553      | 0.558     | 0.531 |
| $S_6$ | PV      | 0.711       | 0.843      | 0.844     | 0.840 |
|       | SynPV   | 0.632       | 0.343      | 0.368     | 0.370 |
| $S_7$ | PV      | 0.787       | 0.810      | 0.780     | 0.779 |
|       | SynPV   | 0.632       | 0.535      | 0.527     | 0.504 |
| $S_8$ | PV      | 0.716       | 0.827      | 0.813     | 0.914 |
|       | SynPV   | 0.672       | 0.490      | 0.492     | 0.499 |

Figures 6 and 7 below show the  $\Delta(t)$  of the validation process between the computer and the Raspberry Pi for each experiment.

As expected, the validation times on the computer were more consistent owing to its processor and memory characteristics. However, the validation times on the Raspberry Pi were more erratic. Experiment  $S_1$  had the highest peak time, with a maximum runtime of roughly 90 ms. This could be because this experiment was trained from scratch, with no data augmentation or fine-tuning, indicating that this model does not employ pre-trained weights like the other models. Additionally, its weight in bytes might be larger than the other models, resulting in higher memory storage and resource utilization. Despite these variances, the models are still appropriate for our application.



**Figure 6.** Delta (t) obtained in the validation within PC.



**Figure 7.** Delta (t) obtained in the validation within the Raspberry Pi.

## 5. Conclusions

This study introduces a low-cost smart device that allows for the detection of possible diseases, increasing the robustness of the classification process by capturing several images of the leaves and integrating techniques based on data fusion. The device has been developed using a Raspberry Pi 4 and a four-camera array and incorporates a deep learning model to classify and display information on an LCD screen and is easily mountable on drones, robots, or agricultural machinery. The performance measures and tests have been performed in a controlled environment with modified images to simulate outdoor spaces and evaluate the system's robustness.

The device offers a more efficient method of visualizing plant disease spots, reducing costs by eliminating the excessive use of fungicides, pesticides, and herbicides.

The results show how the proposed device demonstrates that current EDGE technology permits carrying out plant disease classification and detection systems, considerably lowering the usage cost as no images are transmitted over an Internet connection. Moreover, it allows farmers to perform a pre-analysis of possible diseases that may be present in their plants.

We also show that our system is more robust than a single-camera setup, obtaining better results than the original and synthetic datasets, where noise and unwanted information have been added to the images.

Future work aims to put the proposed system to the test in a natural environment mounted on a robot or drone. Further, another experimental path is to detect changes in plant disease severity over time by adapting models to identify, classify, and assess the extent of disease progression following the disease evolution cycle. Furthermore, determining the effects of multiple infections on plants is also of interest.

**Author Contributions:** Conceptualization, C.M.-D. and J.R.; methodology, C.C.; software, C.M.-D. and J.R.; validation, V.J.; formal analysis, V.J.; investigation, C.M.-D. and J.R.; resources, J.R.; writing—original draft preparation, C.M.-D. and J.R. and C.C. and V.J.; writing—review and editing, V.J.; supervision, V.J. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was partially supported by grant number PID2021-123673OB-C31 funded by MCIN/AEI/ 10.13039/501100011033 and by “ERDF A way of making Europe” and Conselleria d’Innovació, Universitats, Ciència i Societat Digital from Comunitat Valenciana (APOSTD/2021/227) through the European Social Fund (Investing In Your Future).

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

## References

1. Jha, K.; Doshi, A.; Patel, P.; Shah, M. A comprehensive review on automation in agriculture using artificial intelligence. *Artif. Intell. Agric.* **2019**, *2*, 1–12. [[CrossRef](#)]
2. Vadlamudi, S. How Artificial Intelligence Improves Agricultural Productivity and Sustainability: A Global Thematic Analysis. *Asia Pac. J. Energy Environ.* **2019**, *6*, 91–100.
3. Benos, L.; Tagarakis, A.C.; Doliias, G.; Berruto, R.; Kateris, D.; Bochtis, D. Machine learning in agriculture: A comprehensive updated review. *Sensors* **2021**, *21*, 3758. [[CrossRef](#)] [[PubMed](#)]
4. Vishnoi, V.K.; Kumar, K.; Kumar, B. Plant disease detection using computational intelligence and image processing. *J. Plant Dis. Prot.* **2021**, *128*, 19–53. [[CrossRef](#)]
5. Singh, V.; Sharma, N.; Singh, S. A review of imaging techniques for plant disease detection. *Artif. Intell. Agric.* **2020**, *4*, 229–242. [[CrossRef](#)]
6. Li, L.; Zhang, S.; Wang, B. Plant disease detection and classification by deep learning—A review. *IEEE Access* **2021**, *9*, 56683–56698. [[CrossRef](#)]
7. Golhani, K.; Balasundram, S.K.; Vadamalai, G.; Pradhan, B. A review of neural networks in plant disease detection using hyperspectral data. *Inf. Process. Agric.* **2018**, *5*, 354–371. [[CrossRef](#)]
8. Orchi, H.; Sadik, M.; Khaldoun, M. On using artificial intelligence and the internet of things for crop disease detection: A contemporary survey. *Agriculture* **2022**, *12*, 9. [[CrossRef](#)]
9. Qazi, S.; Khawaja, B.A.; Farooq, Q.U. IoT-equipped and AI-enabled next generation smart agriculture: A critical review, current challenges and future trends. *IEEE Access* **2022**, *10*, 21219–21235. [[CrossRef](#)]
10. Marco-Detchart, C.; Rincon, J.; Julian, V.; Carrascosa, C. *Plant Disease Detection: An Edge-AI Proposal. Highlights in Practical Applications of Agents, Multi-Agent Systems, and Complex Systems Simulation. The PAAMS Collection: International Workshops of PAAMS 2022, L'Aquila, Italy, 13–15 July 2022*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 108–117.
11. Costa, L.; Nunes, L.; Ampatzidis, Y. A new visible band index (vNDVI) for estimating NDVI values on RGB images utilizing genetic algorithms. *Comput. Electron. Agric.* **2020**, *172*, 105334. [[CrossRef](#)]
12. Barbedo, J.G.A. A review on the main challenges in automatic plant disease identification based on visible range images. *Biosyst. Eng.* **2016**, *144*, 52–60. [[CrossRef](#)]
13. Perona, P.; Malik, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **1990**, *12*, 629–639. [[CrossRef](#)]
14. Marco-Detchart, C.; Lopez-Molina, C.; Fernandez, J.; Bustince, H. A gravitational approach to image smoothing. In *Advances in Fuzzy Logic and Technology 2017*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 468–479.
15. Madrid, N.; Lopez-Molina, C.; Hurtik, P. Non-linear scale-space based on fuzzy contrast enhancement: Theoretical results. *Fuzzy Sets Syst.* **2021**, *421*, 133–157. [[CrossRef](#)]
16. Camargo, A.; Smith, J. Image pattern classification for the identification of disease causing agents in plants. *Comput. Electron. Agric.* **2009**, *66*, 121–125. [[CrossRef](#)]
17. Rumpf, T.; Mahlein, A.K.; Steiner, U.; Oerke, E.C.; Dehne, H.W.; Plümer, L. Early detection and classification of plant diseases with support vector machines based on hyperspectral reflectance. *Comput. Electron. Agric.* **2010**, *74*, 91–99. [[CrossRef](#)]
18. Gueye, Y.; Mbaye, M. KMeans Kernel-learning based AI-IoT framework for plant leaf disease detection. In *International Conference on Service-Oriented Computing*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 549–563.
19. Khan, S.; Narvekar, M. Novel fusion of color balancing and superpixel based approach for detection of tomato plant diseases in natural complex environment. *J. King Saud Univ. -Comput. Inf. Sci.* **2022**, *34*, 3506–3516. [[CrossRef](#)]
20. Schwarz Schuler, J.P.; Romani, S.; Abdel-Nasser, M.; Rashwan, H.; Puig, D. Reliable Deep Learning Plant Leaf Disease Classification Based on Light-Chroma Separated Branches. In *Frontiers in Artificial Intelligence and Applications*; Villaret, M., Alsinet, T., Fernández, C., Valls, A., Eds.; IOS Press: Lleida, Spain, 2021.
21. Gui, P.; Dang, W.; Zhu, F.; Zhao, Q. Towards automatic field plant disease recognition. *Comput. Electron. Agric.* **2021**, *191*, 106523. [[CrossRef](#)]
22. Atila, U.; Uçar, M.; Akyol, K.; Uçar, E. Plant leaf disease classification using EfficientNet deep learning model. *Ecol. Informatics* **2021**, *61*, 101182. [[CrossRef](#)]
23. 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*; IEEE: Piscataway, NJ, USA, 2009; pp. 248–255.
24. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
25. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, 26 June – 1 July 2016*; pp. 770–778.
26. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
27. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, USA, 26 June – 1 July, 2016*; pp. 2818–2826.
28. Tan, M.; Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning, PMLR, Long Beach, USA, 9–15 June 2019*; pp. 6105–6114.



29. Sabour, S.; Frosst, N.; Hinton, G.E. Dynamic routing between capsules. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
30. Samin, O.B.; Omar, M.; Mansoor, M. CapPlant: A capsule network based framework for plant disease classification. *PeerJ Comput. Sci.* **2021**, *7*, e752. [[CrossRef](#)] [[PubMed](#)]
31. Hassan, S.M.; Maji, A.K. Plant disease identification using a novel convolutional neural network. *IEEE Access* **2022**, *10*, 5390–5401. [[CrossRef](#)]
32. Bhakta, I.; Phadikar, S.; Majumder, K.; Mukherjee, H.; Sau, A. A novel plant disease prediction model based on thermal images using modified deep convolutional neural network. *Precision Agriculture* **2022**, *24*, 1–17. [[CrossRef](#)]
33. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
34. Kristiani, E.; Yang, C.T.; Nguyen, K.L.P. Optimization of deep learning inference on edge devices. In Proceedings of the 2020 International Conference on Pervasive Artificial Intelligence (ICPAI), Taipei, Taiwan, 3–5 December 2020; IEEE: Piscataway, NJ, USA, 2020; pp. 264–267.
35. Beliakov, G.; Bustince Sola, H.; Calvo, T. Studies in Fuzziness and Soft Computing. In *A Practical Guide to Averaging Functions*; Springer International Publishing: Berlin/Heidelberg, Germany, 2016; p. 329.
36. Lucca, G.; Sanz, J.A.; Dimuro, G.P.; Bedregal, B.; Mesiar, R.; Kolesarova, A.; Bustince, H. Preaggregation Functions: Construction and an Application. *IEEE Trans. Fuzzy Syst.* **2016**, *24*, 260–272. [[CrossRef](#)]
37. Lucca, G.; Sanz, J.A.; Dimuro, G.P.; Bedregal, B.; Bustince, H.; Mesiar, R. CF-integrals: A new family of pre-aggregation functions with application to fuzzy rule-based classification systems. *Inf. Sci.* **2018**, *435*, 94 – 110. [[CrossRef](#)]
38. Hughes, D.; Salathé, M.; others. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv* **2015**, arXiv:1511.08060.
39. Zoph, B.; Vasudevan, V.; Shlens, J.; Le, Q.V. Learning transferable architectures for scalable Image recognition. In Proceedings of the IEEE Conference on Computer vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8697–8710.

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