Computers and Electronics in Agriculture 181 (2021) 105951

Contents lists available at ScienceDirect

Computers and Electronics in Agriculture

journal homepage: www.elsevier.com/locate/compag

Original papers

Disease detection in tomato leaves via CNN with lightweight architectures implemented in Raspberry Pi 4

Victor Gonzalez-Huitron [a,*], José A. León-Borges [b], A.E. Rodriguez-Mata [a], Leonel Ernesto Amabilis-Sosa [a], Blenda Ramírez-Pereda [a], Hector Rodriguez [c]

[a] CONACyT-Tecnológico Nacional de México/Instituto Tecnológico de Culiacán, Sinaloa, Mexico
[b] Universidad de Quintana Roo, Av. Chetumal SM 260 MZ 21 y 16 LT 1-01, Fracc. Prado Norte, 77519 Cancún, Quintana Roo, Mexico
[c] Tecnológico Nacional de México/Instituto Tecnológico de Culiacán, Sinaloa, Mexico

ARTICLE INFO

Keywords:
Deep learning
Plant disease classification
Image processing
Precision agriculture

ABSTRACT

Deep learning has made essential contributions to classification and detection tasks applied to precision agriculture; however, it is vitally important to move towards an adoption of these techniques and algorithms through low-cost and low-consumption devices for daily use in crop fields. In this paper, we present the training and evaluation of four recent Convolutional Neural Networks models for the classification of diseases in tomato leaves. A subset of the Plantvillage dataset consisting of 18,160 RGB images has been divided into ten classes for transfer learning. The selected models have depthwise separable convolution architecture for application in low-power devices. Evaluation and analysis quantitatively and qualitatively is performed via quality metrics and saliency maps. Finally, an implementation on the Raspberry Pi 4 microcomputer with a graphical user interface is developed.

1. Introduction

Tomato production worldwide exceeds 177 million tons. It has among the leading producers to Mexico, which retains a prominent place in international agricultural trade, reaching up to 5% of tomato tons traded in the world (Almaraz Sánchez et al., 2019). This production reflects the great efforts made by this sector, where tomato, whose production exceeds 4 million tonnes per year, should be highlighted. However, much of the country's agricultural production is done in the open air (Jirón-Rojas et al., 2016), which requires a more significant use of pesticides (Lahiri and Orr, 2018). In consequence, higher losses and problems of yield and quality in the final product (Elgueta et al., 2020), which is why the early detection of pests and diseases is a task of particular interest to the region.

In precision agriculture, the need for early detection systems is a pressing need due to the economic, environmental and social importance of adequate and efficient use of resources for the generation of horticultural crops (Maes and Steppe, 2019). Currently, tomato cultivation presents significant challenges due to the number of diseases, pests, and anomalies in general that can occur in these crops worldwide (Basak, 2016). In several areas of technology, efforts are made to propose solutions with different approaches, from which the results and environmental conditions of the production environment are continuously improved, such as sowing systems, monitoring, automation, harvesting, selection, and treatments (Tangarife and Díaz, 2017; Zhang et al., 2018; Taqi et al., 2017; Elvanidi et al., 2018).

In recent years, the accelerated development of open-source hardware has driven the development and implementation of low-cost devices for agricultural monitoring, which show the capacity to incorporate image processing and artificial intelligence. In a trend that continues to rise, but which presents among its main challenges the availability, reliability, and mobility (Khanna and Kaur, 2018); for example in Osroosh et al. (2017) is presented the design and implementation of a monitoring system using RGB and thermal images, implemented in Raspberry Pi 3, which manages to work in real conditions and adverse environments. In Hsu et al. (2018) Raspberry Pi is used to design a low-cost monitoring network in agricultural applications for broader adoption. A monitoring environment including several devices and interfaces is presented in Morais et al. (2019), establishing the communication capability in low-cost devices for agricultural environments.

The detection of diseases in agricultural fields is largely done under

* Corresponding author.
E-mail address: victor.gonzalez@conacyt.mx (V. Gonzalez-Huitron).

https://doi.org/10.1016/j.compag.2020.105951
Received 12 June 2020; Received in revised form 12 October 2020; Accepted 13 December 2020
Available online 15 January 2021
0168-1699/

expert visual inspection. However, it is subject to human beings limitations, such as fatigue when faced with long working days or error-prone possibility when performing a task that can become highly repetitive (Petrellis, 2018). The development of tools for the automation of these tasks is a topic of continuous interest due to the complexity and high amount of visual and environmental factors surrounding this task.

The growing use of deep learning models has achieved remarkable improvements, expressly, in classification tasks (Russakovsky et al., 2015). However, it is not limited to this; it has also been used in forecasting, scenario reconstruction, image enhancement, detection and reconstruction in agricultural applications. Xue et al. (2019), Zhang et al. (2019) and Kussul et al. (2017). While in the specific field of horticulture it has been used recently for crop classification, anomalies detection, water stress, classification and selection of products and diseases or pests detection (Maeda-Gutiérrez et al., 2020; da Costa et al., 2020).

In particular, the detection of diseases, pests, or anomalies can be characterized as a set of patterns in images of different origins, where the source of the images can be from various sensors, such as infrared, thermal, visual spectrum, multispectral, hyperspectral or chlorophyll fluorescence (Mahlein, 2016). This range of possibilities increases the complexity of decision making for the model design for disease detection. A notable difference between these sensors being the bandwidth and number of channels that will be taken into account for the generation of a data matrix. In current researches, most of the early detection systems are developed with multispectral images or within the visual spectrum, where most of the work focused on deep learning uses RGB images due to the amount of data already existing through previous researches, for replication and improvement of proposals, while the information from multispectral sensors, although they provide more information, the lack of public datasets and the cost of equipment are a limitation to their higher adoption, which is significant in developing countries.

In related researches, most of the early detection systems are developed with multispectral or within the visual spectrum (RGB) images (Verma et al., 2018; Barbedo, 2018; Jameel et al., 2020), where most of the work is focused on deep learning using the RGB space. This is a consequence of the amount of data already existing through previous researches. Although the information from multispectral sensors provide more information, the lack of public datasets and the cost of equipment are a limitation to their higher adoption, which is significant in developing countries (Marconi et al., 2019).

Thus, in recent works, deep learning has been used over machine learning methods due to better accuracy and hardware availability specifically to classify diseases in datasets of horticultural origin (Gutierrez et al., 2019). Deep learning and plant diseases researches such as (Rangarajan et al., 2018), also show transfer learning with state-of-the-art models AlexNet and VGG16, with higher precision results than those obtained with machine learning classifiers. Also, in Brahimi et al. (2018), classification models for ten tomato diseases are presented employing transfer learning from classic models such as AlexNet and GoogleNet; besides, saliency maps for the identification of regions on the leaves with some particular pattern are presented. As far as hardware implementations are concerned. Other efforts have been directed to use more modern architectures, as in Kumar and Vani (2019), where ResNet and Xception architectures are used with better results compared to previous years' architectures. In Durmus et al. (2017), a classification model based on the AlexNet architecture was developed for ten classes of tomato images, with an implementation on the NVIDIA Jetson TX1 hardware, and training on the same device. The efforts presented in Khan and Narvekar (2020) where a mixed dataset is enhanced, aims for a future mobile applications of the trained models, and finally, in Picon et al. (2019) a dataset for five crops and 17 classes was obtained from images in real conditions, presenting three classification models based on the ResNet50 architecture and developing an application for mobile devices that facilitates the application of the research.

Given the background, the present work shows the results obtained when carrying out training through transfer learning in architectures that have recently been exploited for agricultural purposes, and their implementation in low-cost hardware, such as the Raspberry Pi 4 microcomputer. Thus one of the objectives of this work is the implementation of CNN models in a low-cost and low power consumption device capable of processing the information in real conditions to obtain results that facilitate early detection of anomalies in tomato crops.

The rest of the paper is organized as follows. In Section 2, datasets, data augmentations performed and methodology for Convolutional Neural Networks (CNN) training is presented. In Section 3, the quality criteria, experimental results and the hardware implementation is detailed. Finally, in Section 4, the conclusions are shown.

2. Materials and methods

First, the dataset used and their modifications via data augmentation are presented, where local transformations are performed to the dataset then, a brief description of the CNN architectures employed are shown, along the hyperparameters proposed for the training stage, following, the results obtained are evaluated with objective and subjective criteria to establish advantages and disadvantage for the trained models, and finally, a GUI design is presented to help the usage from the hardware implementation. The block diagram of the proposed framework is presented in Fig. 1.

2.1. Dataset and data augmentation

For the training of the models, the PlantVillage dataset (Hughes and Salathé, 2015; Mohanty et al., 2016) was selected, consisting of 54,305 images from 14 different crops and 38 different classes consisting of healthy and unhealthy leaf images, in color, grayscale, and segmented format. From this dataset, only 18,215 images of tomato leaves divided into ten classes were used, only the color images were selected, the resolution of the images is 256×256 pixels, the number of images per class varies from 373 to 5,358, with an average of 1,821. In the Table 1 the dataset for the tomato leaves are presented in more detail. In Fig. 2 some samples with data augmentation are shown.

To avoid overfitting on the training models, and generalize their response, transformations were made to increase the data, being these horizontal flip, and rotations in four different angles, obtaining six times more images from the original tomato class from PlantVillage dataset. The augmented dataset was split 70% for training and 30% for testing purposes.

2.2. CNN architectures

According to Brahimi et al. (2018), retraining of architectures using the PlantVillage dataset gives better results if performed with initial weights from the ImageNet dataset (Russakovsky et al., 2015) compared to a random weight initialization. Similarly, if all layers are labeled for training instead of only the top ones, the results should be superior. For this work, four architectures were selected to perform transfer learning: MobileNetV2 (3 M parameters, 90.1% Accuracy) (Sandler et al., 2018), NasNetMobile (5 M parameters, 91.9% Accuracy) (Zoph et al., 2018), Xception (22 M parameters, 94.5% Accuracy) (Chollet, 2017), and MobileNetV3 (3 M parameters, 75.2% Accuracy) (Howard et al., 2019) due to their high performance with the ImageNet Top-5 challenge and their relatively low number of parameters compared to deeper architectures, which is mandatory for real-time crop diagnosis in resource constrained mobile devices, where lightweight architectures are suitable and proven for these tasks (KC et al., 2019).

Depthwise separable convolutions have the primary goal of replacing a full convolution operator with a factorized version that obtains similar results with two convolution operators. Where the first layer is called depthwise convolution, it is a set of filters whose kernels are strictly 2D,

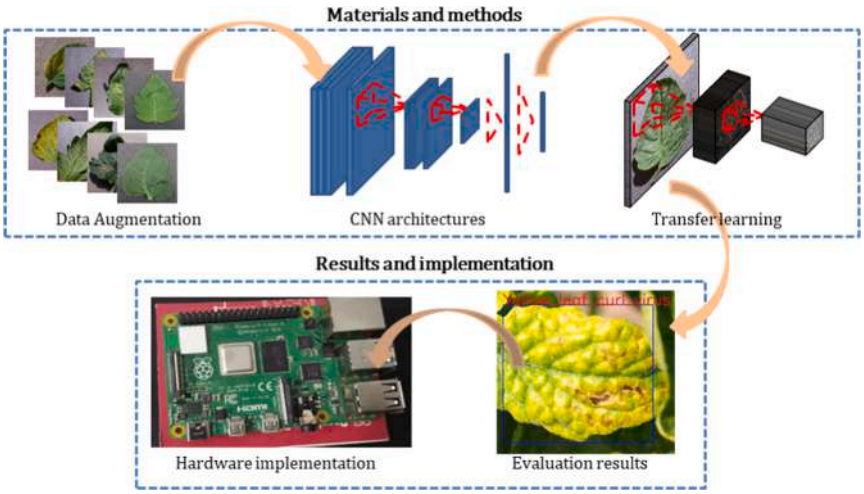

Fig. 1. Diagram with the main stages of the proposal.

Table 1
Image quantity for tomato classes before and after data augmentation.

Class	Number of images	with data augmentation
Bacterial Spot	2,133	12,798
Early Blight	1,010	6,060
Healthy	1,595	9,570
Late Blight	1,916	11,496
Leaf Mold	962	5,772
Mosaic Virus	1,779	10,674
Septoria Leaf Spot	1,677	10,062
Two Spotted Spider Mites	1,412	8,472
Target Spot	373	2,238
Yellow Leaf Curl Virus	5,358	32,148
Total	**18,215**	**109,290**

the second operator, which contains a kernel size of 1x1, reducing the number of channels of the processed volume exclusively. In this way, the number of operations and parameters in the network are drastically reduced, decreasing the processing time by 8 to 9 times compared to standard convolutions with minimal differences in accuracy (Howard et al., 2017).

For context, the main ideas behind these architectures are briefly explained. The NasNetMobile model is the result of the process called NasNet search space, which consists of the definition of a convolutional cell, integrated by a filters bank to be trained, which is adjusted to a given dataset, for example, ImageNet, and from this serial cell connections are made. Two variants for this cell are designed, according to the size of its output, either preserving the dimensions (Normal Cell) or reducing them according to the stride value used in the internal convolutions (Reduction Cell). The number of connected cells for the network design is considered an adjustable parameter for new datasets. MobileNetV2 includes, in addition to depthwise separable convolutions, adds the use of linear bottlenecks, which consists of 1x1 convolutional filters depth-wise processed, reducing these depths with a minimum variation in the final results, consequently decreasing the number of operations and parameters to be adjusted in the network. Xception, (Extreme Inception) is a variant of the inception block (Szegedy et al., 2015), which contains parallel filter banks of 1x1 (bottleneck) followed by others with different kernel size, then a concatenation at the output of these modules is performed. It is proposed to use a single bottleneck layer for a whole set of filters within the module and then apply depthwise separable convolutions, the main difference with other architectures being this change in the order of operations. Finally, MobileNetV3 is presented in two sizes (small and large), according to the target application, designed from Network Architecture Search (NAS) to optimize the architecture of each block and the NetAdapt algorithm, in charge of optimizing the number of filters per layer, consequently a proposed architecture is presented that combines blocks used in MobileNetV1 and MobileNetV2, using hard-sigmoid as activation function.

2.3. Transfer learning

The hyperparameters shown in Table 2 were defined to perform the training. The four models mentioned above, MobileNetV2, NasNetMobile, Xception and MobileNetV3 were trained through transfer learning, setting an input resolution of 224x224 pixels. Weights coming from

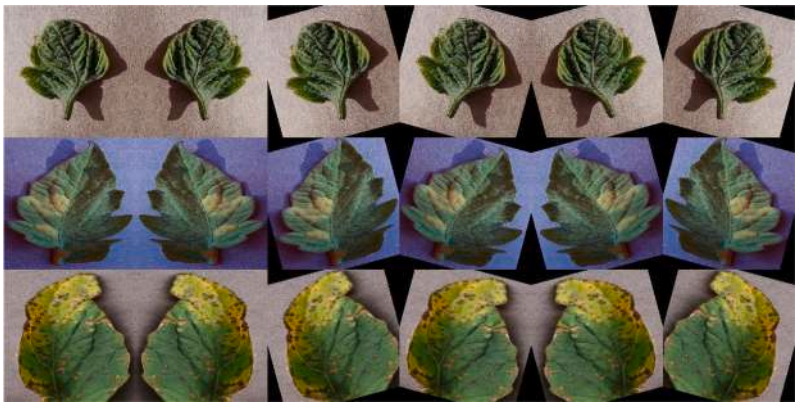

Fig. 2. Samples from data augmentation.

Table 2
Hyperparameters to perform training stage.

Hyperparameters	Values
classes	10
epochs	10
Batch size	24
Optimizer	Adam
learning rate	0.001
β_1, β_2	0.9, 0.999
$\in$	$1e^{-}7$

ImageNet training models were selected for initialization. Also, the last layers were replaced, so the feature map obtained at the end of the CNN is connected to a GlobalAveragePooling layer followed by a softmax layer with ten classes at the output. All CNN layers were labeled as trainable for a deep training.

3. Experimental results

In this section, the quality metrics to be used are presented first, followed by the results per-model and per-class, as well as examples of classification to obtain quantitative and qualitative overview of each model. Finally, we present the overall measures for the models, gathering in addition to the descriptors, possible causes, and areas of opportunity for improvement of results.

3.1. Quality criteria

Results presentation and comparison between models, the following quality measures were considered: Accuracy, which is the ratio of correctly labeled images to the total number of samples, as shown in (1). Precision is the probability given a positive label, how many of them are actually positive (2). Recall or Sensitivity is the accuracy of positive predicted instances describing how many were labeled correctly (3). And F_1 score, as an additional measure of classifier accuracy, which considers both precision and recall (4)

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

$$Precision = \frac{TP}{TP + FP}, \tag{2}$$

$$Recall = \frac{TP}{TP + FN}, \tag{3}$$

$$F_1 = 2^{*}\frac{Precision^{*}Recall}{Precision + Recall}, \tag{4}$$

where *TP* refers to true positives, *TN* to true negatives, *FP* to false positives and *FN* to false negatives. Also training times and confusion matrices are presented for the four models implemented in this work.

3.2. Evaluation results

First, the quality metrics for each of the ten classes are presented. There it is possible to observe in detail the performance of the classifiers used and the labels that represent a significant challenge within the dataset. Table 3 shows the quality measurements and the number of test images for each one.

Concerning quality metrics by class, Bacterial Spot for the MobileNetV2 model, contains a high precision value and low recall. This suggests that there were no false positives in its response, which would indicate that images from another class were labeled as Bacterial Spot. However, there were many false negatives, meaning that of the total images from Bacterial spot, less than half were correctly evaluated. Similar responses are observed for the Early Blight and Target Spot classes. The NasNetMobile model shows comparable behavior for Early Blight and Target Spot.

Table 3
Image quantity for tomato classes before and after data augmentation.

MobileNetV2	Precision	Recall	F_1 score
Bacterial Spot	1.00	0.38	0.55
Early Blight	0.81	0.44	0.57
Healthy	1.00	0.97	0.99
Late Blight	0.97	0.90	0.94
Leaf Mold	0.64	0.82	0.72
Mosaic Virus	0.12	1.00	0.22
Septoria Leaf Spot	0.68	0.91	0.77
Two Spotted Spider Mites	0.83	0.76	0.80
Target Spot	0.83	0.68	0.75
Yellow Leaf Curl Virus	0.99	0.76	0.86
NasNetMobile	Precision	Recall	F_1 score
Bacterial Spot	0.98	0.81	0.88
Early Blight	0.83	0.47	0.60
Healthy	1.00	0.88	0.94
Late Blight	1.00	0.69	0.81
Leaf Mold	0.40	0.84	0.54
Mosaic Virus	0.87	0.82	0.84
Septoria Leaf Spot	0.69	0.98	0.81
Two Spotted Spider Mites	0.92	0.76	0.83
Target Spot	0.94	0.67	0.78
Yellow Leaf Curl Virus	0.91	1.00	0.95
Xception	Precision	Recall	F_1 score
Bacterial Spot	1.00	1.00	1.00
Early Blight	0.99	0.99	0.99
Healthy	1.00	1.00	1.00
Late Blight	0.99	1.00	1.00
Leaf Mold	0.97	1.00	0.99
Mosaic Virus	1.00	1.00	1.00
Septoria Leaf Spot	1.00	1.00	1.00
Two Spotted Spider Mites	1.00	0.99	0.99
Target Spot	1.00	1.00	1.00
Yellow Leaf Curl Virus	1.00	1.00	1.00
MobileNetV3	Precision	Recall	F_1 score
Bacterial Spot	0.99	0.97	0.98
Early Blight	0.94	0.95	0.94
Healthy	1.00	0.99	0.99
Late Blight	0.97	0.98	0.97
Leaf Mold	0.96	0.99	0.98
Mosaic Virus	0.98	0.99	0.99
Septoria Leaf Spot	1.00	0.94	0.97
Two Spotted Spider Mites	0.98	0.98	0.98
Target Spot	0.99	0.97	0.98
Yellow Leaf Curl Virus	0.98	1.00	0.99

On the other hand, a low precision value and high Recall value, as shown for the Leaf Mold and Septoria Leaf Spot classes for the MobileNetV2 and NasNetMobile models, or even more evident in the Mosaic Virus class of MobileNetV2, indicates that for those classes there is a high value of false positives, labeling images from other classes to Mosaic Virus. It is important to note the precision and Recall for the Healthy class from all four classifiers, as they indicate their performance in differentiating between healthy and diseased leaves is close to optimal with dataset-like images. Specifically, if there is a maximum precision, the number of false positives will remain at virtually zero. Therefore, precision is vital in early detection stages since, in case of error by the classifier, it is preferable to label diseases where there are none (low recall) than to miss a disease (low precision). Despite having a lower score for disease classification, MobileNetV2 separates better healthy leaves than diseased ones, compared to NasNetMobile. If it were necessary to establish the model for binary classification with the best relationship response and lower processing time, MobileNetV3 would be the one chosen. Xception, on the other hand, is the classifier with the best performance; however, its computational cost is higher due to its number of parameters. F_1 score presents a measure that combines pre-

cision and recall with equal weighting; this value is useful if, in the classifier application, a false negative has the same importance as a false positive.

Fig. 3 shows the confusion matrix charts, where it is easier to identify the classes that caused the most error in the trained models. There the number of images that form either true or false positives for each class can be obtained, which is a visual aid for a better understanding of the precision and recall values obtained for each disease and model.

As a complement to the confusion matrices, Fig. 3 shows the Precision-Recall curves obtained for each class, where each classifier is observed. It is essential to highlight the behavior of MobileNetV3 and Xception, which are close to the ideal behavior, considering that MobileNetV3 presents a smaller number of parameters. On the other hand, MobileNetV2 and NasNetMobile obtain a favorable result in the healthy class, but present a lower performance before classes with a low number of positive instances, which indicates that classes with a higher number of elements are favored (see Fig. 4).

Since the image quantity per class is not uniform, the average values for each quality metric should be adjusted by incorporating weights per class, according to (5). In Table 4, the total results for the classifiers are presented together with the training time required for their generation. The accuracy obtained from the four models, which is a value that tells us what proportion of true instances, positive and negative, were correctly classified by the model. For the MobileNetV2 and NasNetMobile architectures, the accuracy values are lower compared to the rest for two main reasons: first, the number of parameters to be trained; for this work, all layers were retrained, which achieves a greater generalization in the response compared to proposals that replace only fully connected layers (Brahimi et al., 2018). In consequence, training times are notably larger despite containing a lower number of parameters. Second, the low number of parameters is a preponderant factor when implemented low-power and low-performance devices are considered. The relationship between the number of parameters and the quality of the classifier results is evident. However, in this relationship and with a view to implementations in low-power devices, it must be established which are the least possible parameters capable of delivering acceptable results for a specific application and how they can be applied effectively for each architecture. Therefore, if the target hardware must offer the possibility of a wide adoption and lower cost, we can establish that the processing capacity will necessarily be lower, and a general comparison as shown in Table 4 is a first indicator that can aid to identify light architectures and their general performance for applications in agriculture, while the analysis in Table 3 indicates us in a detailed way the type of error to be expected, consequently the decision making and adjustments with a view to a classification system through artificial intelligence can be more specific and objective, through the design and implementation of architectures tailored to the application. It is possible to determine that the training performed is consistent with state-of-the-art results. State-of-the-art models AlexNet, GoogLeNet, and resNet18 were trained for 30 epochs, with ImageNet weights at initialization and replacing only top three layers (Maeda-Gutiérrez et al., 2020). Due to high number of parameters, were not considered for hardware implementation.

$$WeightedMetric = \frac{\sum_{class=1}^{T} I_{Class} * Q_{class}}{I_{total}} \tag{5}$$

where I_{class} denotes the image quantity per a given class, Q_{class} represents the quality metric obtained per-class and I_{total} is the total quantity images from the testing dataset.

The following are qualitative results, where saliency maps are presented to show the activation areas for some of the trained classes. The objective of this visualization is to corroborate that the model has learned relevant patterns in the training images. Saliency maps are extracted using a CNN on the image labels, so no additional annotation is required. The computation of the image-specific saliency map is

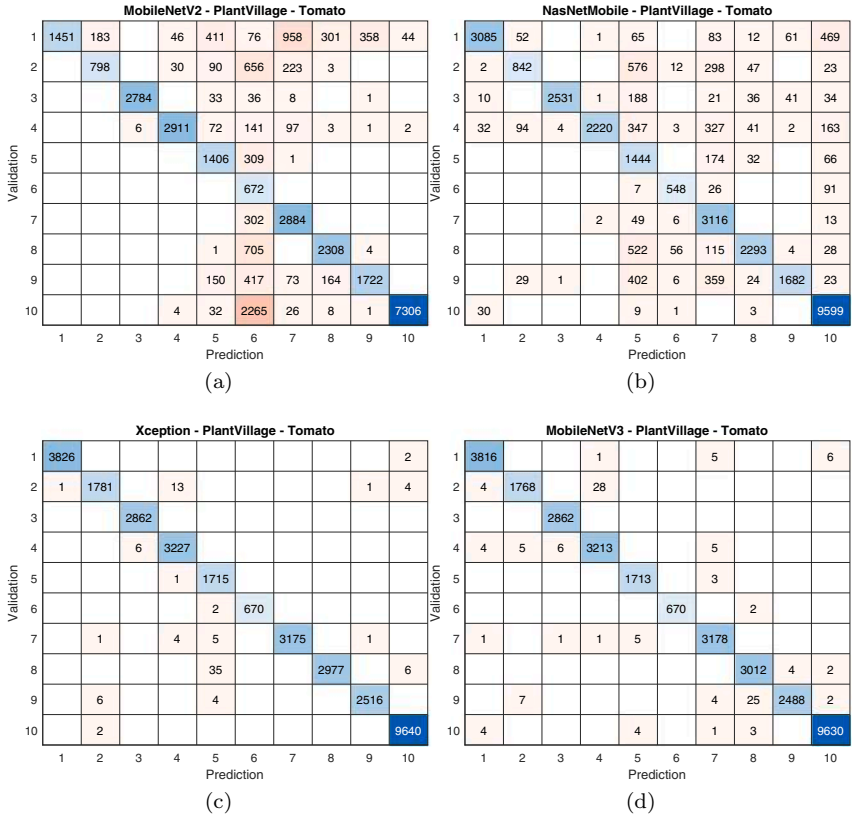

Fig. 3. Confusion matrix charts generated with PlantVillage tomato dataset and deep transfer learning with, a) MobileNetV2 b) NasnetMobile c) Xception and, d) MobileNetV3 CNN architecture.

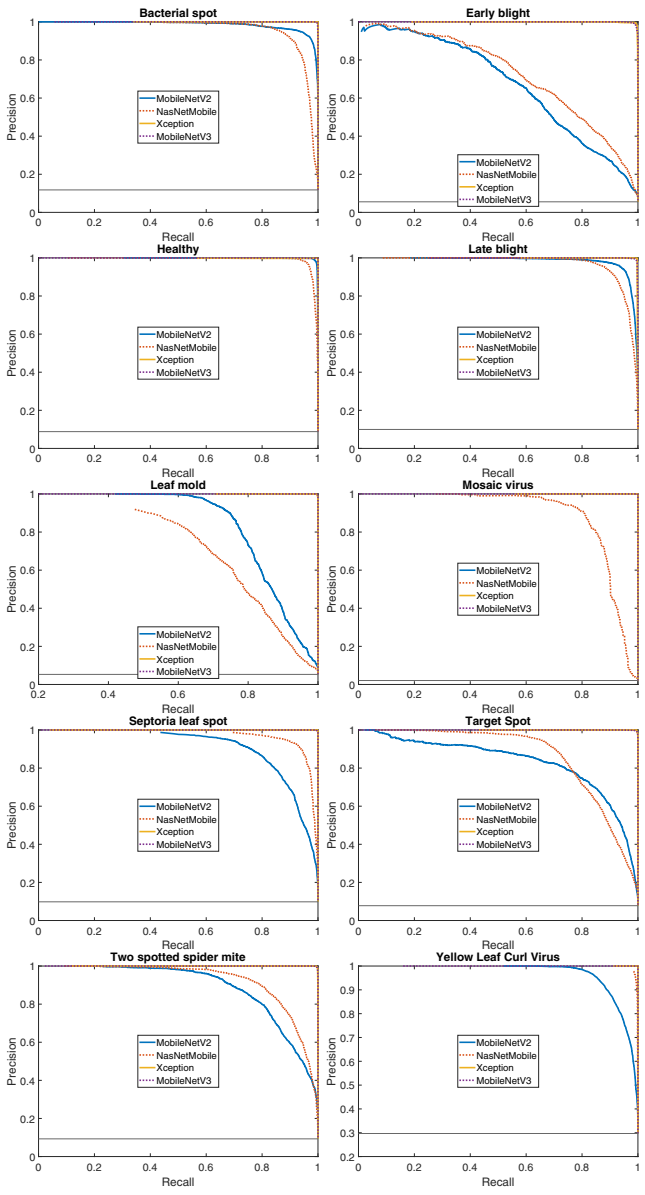

Fig. 4. Precision-Recall curves generated with PlantVillage tomato dataset and deep transfer learning.

estimated for a single class, and only requires a single back-propagation pass (Simonyan et al., 2014). In Fig. 5 some examples are shown, which were taken from the test data set, there, the regions of the leaves that activate each model for its respective class are exposed. In particular the Xception column shows that the higher activation is done in areas where the leaves have anomalies, while the other two models have some inconsistencies, this translates into classification errors.

3.3. Hardware implementation

The models were trained and evaluated on a CPU intel i7-7700 with 16 Gb RAM and GPU using an NVIDIA GTX 1070 with 1920 CUDA cores and a memory interface of 8 Gb GDDR5. The process was implemented on the IDE Spyder 4.1.3, and the main stages were programmed using OpenCV 3.4.2, Keras 2.3.1, TensorFlow 2.1, tf-keras-vis 0.3.1 for saliency maps (Kotikalapudi, 2017), and Tensorflowlite-bin (Hyodo, 2019) for hardware implementation. Raspberry Pi 4 contains a Broadcom BCM2711, Quad-core Cortex-A72 (ARM v8) at 1.5 GHz, 4 Gb SDRAM, and a maximum of 15 W power consumption, including the webcam used in this work, which is a Logitech C920.

For the implementation in raspberry pi 4, Operating System Raspbian 10 (Buster) was installed. First, a graphic user interface (GUI) was designed to allow the capture of photographs. This interface was developed through Tkinter to avoid third party libraries and to avoid compatibility issues, proposing the use of two main windows. In the first one, located on the left, captured from the webcam is shown; in the right one, the last screenshot within the region of interest frame is displayed. In addition, there are three buttons at the bottom for capturing (screenshot and obtain ROI), analyzing (Classify with CNN architecture), and saving images, all saved images are stored in SD memory card where the operative system is installed. Since the CNN models are exclusively for classification from leaves images, the proposal is to use of a reference frame within the display. The saved image is selected from the original screenshot. At the same time, the detection, date and time are recorded in the name of the generated file. Concerning the CNN architectures implementation, a reduction of the models is made to be compatible with TensorFlowLite; this entails the quantization of parameters and reduction in numerical accuracy, decreasing the computational cost required to make the predictions. The GUI is shown in Fig. 6. In those examples, the proposed implementation and a screenshot from a mobile device is presented. Using a remote desktop client adds funcionality with a touch screen and mobile devices, the remote desktop software used was Real VNC, client and server. For images evaluation from tomato crops, a confidence value of 0.8 was proposed for the top class predicted.

Qualitative results were obtained from 100 images captured from a tomato crop that showed visible deterioration due to diseases. A labeling session was performed to manually select the leaves from the captured images in real-life conditions and classify on the trained models. The four trained models were used to evaluate the images. Some examples are presented in Fig. 7, showing the class selected by the model and the saliency map obtained. It should be noted that the Xception model showed the highest consistency when visually inspecting the results, which is consistent with the quantitative results presented above. The classification shown is derived from the Xception model in the evaluation of the images. The crop consisted of 12 tomato plants exposed to open air, after five months of growth began to develop visible signs of abnormalities in the leaves.

Table 4
Weighted quality metrics for the trained and state-of-the-art models.

Model	Accuracy	Precision	Recall	F_1 score	$\frac{time}{epoch}$ (seconds)	Parameters (Millions)
MobileNetV2	0.75	0.89	0.75	0.78	1172	3.53
NasNetMobile	0.84	0.88	0.84	0.85	1694	5.32
Xception	1.00	1.00	1.00	1.00	2512	22.91
MobileNetV3	0.98	0.98	0.98	0.98	462	3.05
AlexNet	0.98	0.98	0.98	0.98	188	61
GoogLeNet	0.99	0.99	0.99	0.99	266	7
ResNet18	0.99	0.98	0.98	0.98	294	11.7

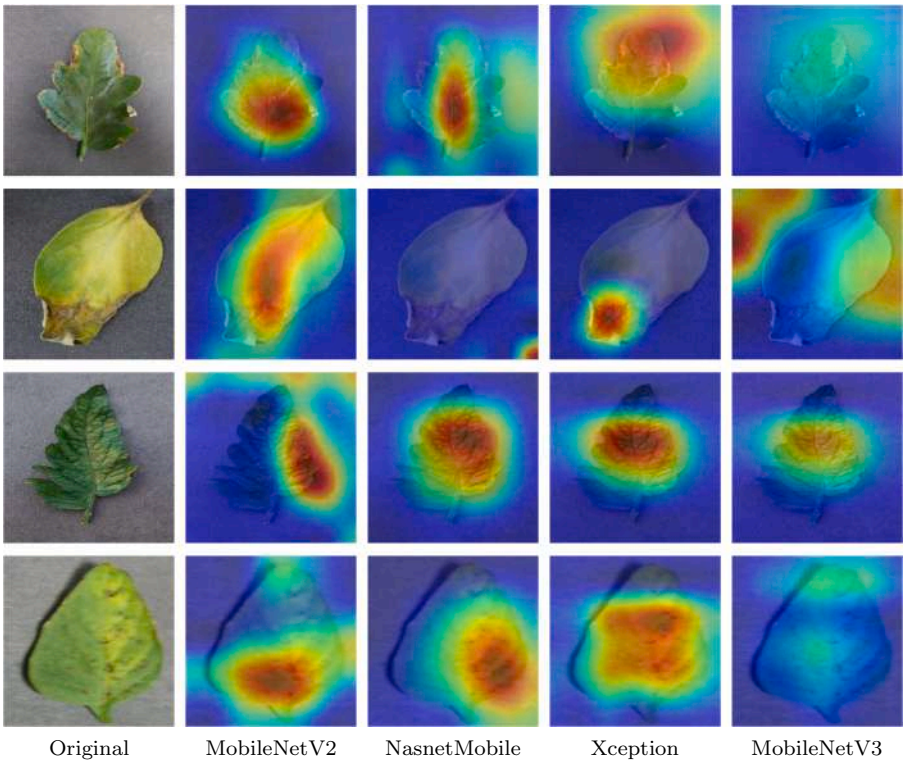

Fig. 5. Saliency maps for Bacterial Spot (first row), Late blight (second row) Leaf mold (third row) and Yellow leaf curl virus (fourth row).

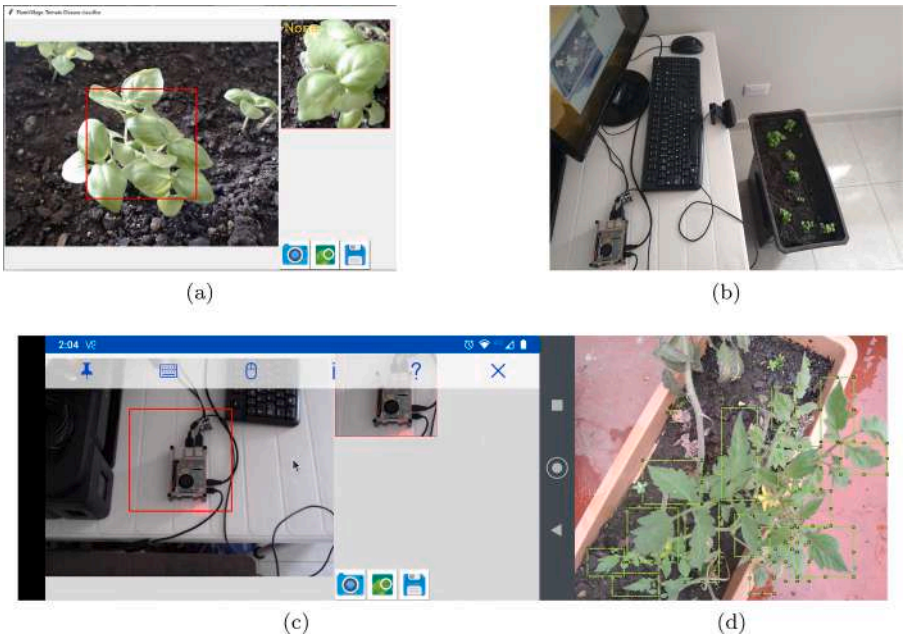

Fig. 6. Hardware implementation a) GUI designed, b) working system, c) GUI screenshot from mobile device and d) image from labeling session.

4. Conclusions

In this paper, a comparison and evaluation of four of the most popular models were presented, with better performance with state-of-the-art datasets and a lower number of parameters than other architectures due to the incorporation of depth-wise separable convolutions and light architectures. This comparison was made with the Plantvillage dataset to perform an analysis and select the most suitable one for a particular application. The four models were implemented in a Raspberry Pi 4 microcomputer to demonstrate its capacity to be incorporated in the future in more exhaustive fieldwork. The main contributions are: first, the deep transfer learning carried out for the selected models in an exhaustive way. Second, the evaluation and analysis with different quality metrics to describe the behavior of the models quantitatively and qualitatively to a major extent. Third, quantification of the models for their implementation in Raspberry Pi 4 and finally, development of a GUI for the use of the models trained in PC, Raspberry Pi 4, or in mobile devices.

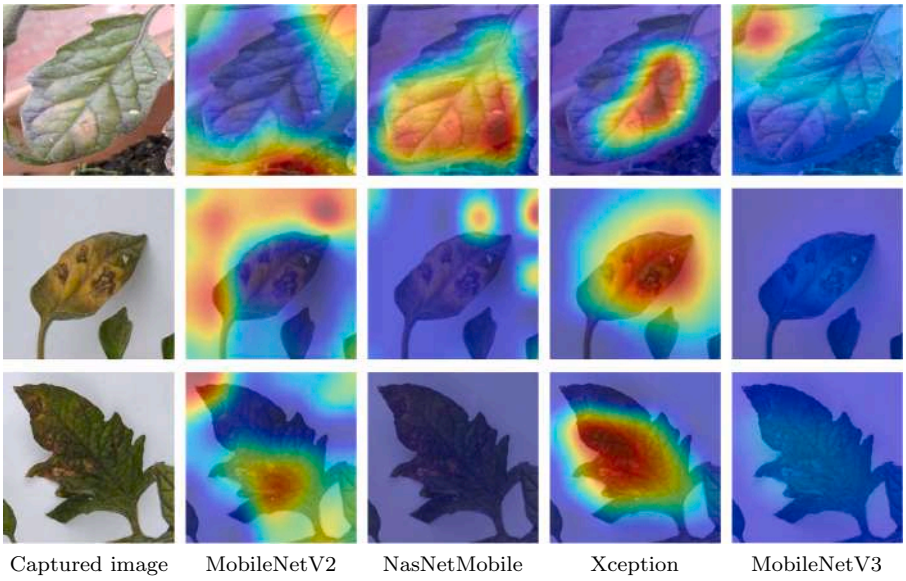

Fig. 7. Saliency maps for Late Blight (first row), Septoria (second row) Early Blight (third row).

CRediT authorship contribution statement

Victor Gonzalez-Huitron: Conceptualization, Methodology, Writing - original draft, Writing - review & editing. **José A. León-Borges:** Software, Validation. **A.E. Rodriguez-Mata:** Formal analysis, Investigation. **Leonel Ernesto Amabilis-Sosa:** Formal analysis, Investigation. **Blenda Ramírez-Pereda:** Data curation, Resources. **Hector Rodriguez:** Conceptualization, Resources, Funding acquisition.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at https://doi.org/10.1016/j.compag.2020.105951.

References

Almaraz Sánchez, A., Ayala Escobar, V., Landero Valenzuela, N., Tlatilpa Santamaría, I. F., Nieto Angel, D., 2019. First report of colletotrichum truncatum of solanum lycopersicum in mexico. Plant Dis. 103 (7), 1782. https://doi.org/10.1094/PDIS-10-18-1809-PDN.

Barbedo, J.G., 2018. Factors influencing the use of deep learning for plant disease recognition. Biosyst. Eng. 172, 84–91. https://doi.org/10.1016/j.biosystemseng.2018.05.013.

Basak, J., 2016. Tomato yellow leaf curl virus: A serious threat to tomato plants world wide. J. Plant Pathol. Microbiol. 7 (4) https://doi.org/10.4172/2157-7471.1000346.

Brahimi, M., Arsenovic, M., Laraba, S., Sladojevic, S., Boukhalfa, K., Moussaoui, A., 2018. Deep Learning for Plant Diseases: Detection and Saliency Map Visualisation. Springer International Publishing, Cham, pp. 93–117. doi:10.1007/978-3-319-90403-0_6.

Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1800–1807. https://doi.org/10.1109/CVPR.2017.195.

da Costa, A.Z., Figueroa, H.E.H., Fracarolli, J.A., 2020. Computer vision based detection of external defects on tomatoes using deep learning. Biosyst. Eng. 190, 131–144. https://doi.org/10.1016/j.biosystemseng.2019.12.003.

Durmus, H., Günes, E.O., Kirci, M., 2017. Disease detection on the leaves of the tomato plants by using deep learning. In: 2017 6th International Conference on Agro-Geoinformatics, pp. 1–5.

Elgueta, S., Valenzuela, M., Fuentes, M., Meza, P., Manzur, J.P., Liu, S., Zhao, G., Correa, A., 2020. Pesticide residues and health risk assessment in tomatoes and lettuces from farms of metropolitan region chile. Molecules 25 (2), 355. https://doi.org/10.3390/molecules25020355.

Elvanidi, A., Katsoulas, N., Kittas, C., 2018. Automation for water and nitrogen deficit stress detection in soilless tomato crops based on spectral indices. Horticulturae 4 (4), 47. https://doi.org/10.3390/horticulturae4040047.

Gutierrez, A., Ansuategi, A., Susperregi, L., Tubío, C., Rankic, I., Lenža, L., 2019. A benchmarking of learning strategies for pest detection and identification on tomato plants for autonomous scouting robots using internal databases. J. Sensors 2019, 1–15. https://doi.org/10.1155/2019/5219471.

Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H., 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR abs/1704.04861. URL http://arxiv.org/abs/1704.04861.

Howard, A., Sandler, M., Chen, B., Wang, W., Chen, L., Tan, M., Chu, G., Vasudevan, V., Zhu, Y., Pang, R., Adam, H., Le, Q., 2019. Searching for MobileNetV3. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 1314–1324.

Hsu, T.C., Yang, H., Chung, Y.C., Hsu, C.H., 2018. A Creative IoT agriculture platform for cloud fog computing. Sustainable Comput.: Informat. Syst. https://doi.org/10.1016/j.suscom.2018.10.006.

Hughes, D.P., Salathé, M., 2015. An open access repository of images on plant health to enable the development of mobile disease diagnostics through machine learning and crowdsourcing. CoRR abs/1511.08060. URL http://arxiv.org/abs/1511.08060.

Hyodo, K., 2019. Tensorflowlite-bin. https://github.com/PINTO0309/TensorflowLite-bin.

Jameel, S.M., Rehman Gilal, A., Hussain Rizvi, S.S., Rehman, M., Hashmani, M.A., 2020. Practical implications and challenges of multispectral image analysis. In: 2020 3rd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), pp. 1–5.

Jirón-Rojas, R.L., Nava-Camberos, U., Jiménez-Díaz, F., Alvarado-Gómez, O.G., Ávila Rodríguez, V., García-Hernández, J.L., 2016. Densidades debactericera cockerelli (sulc) e incidencia del "permanente del tomate en diferentes condiciones de producción del tomate. Southwestern Entomol. 41 (4), 1085–1094. https://doi.org/10.3958/059.041.0408.

KC, K., Yin, Z., Wu, M., Wu, Z., 2019. Depthwise separable convolution architectures for plant disease classification. Comput. Electron. Agric. 165, 104948. doi:10.1016/j.compag.2019.104948.

Khan, S., Narvekar, M., 2020. Disorder Detection in Tomato Plant Using Deep Learning. Springer, Singapore, pp. 187–197. doi:10.1007/978-981-15-3242-9_19.

Khanna, A., Kaur, S., 2018. Evolution of Internet of Things (IoT) and its significant impact in the field of Precision Agriculture. Comput. Electron. Agric. 157(November 2018), 218–231. doi:10.1016/j.compag.2018.12.039.

Kotikalapudi, R., 2017. Contributors: keras-vis. https://github.com/raghakot/keras-vis.

Kumar, A., Vani, M., 2019. Image based tomato leaf disease detection. In: 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), pp. 1–6.

Kussul, N., Lavreniuk, M., Skakun, S., Shelestov, A., 2017. Deep learning classification of land cover and crop types using remote sensing data. IEEE Geosci. Remote Sens. Lett. 14 (5), 778–782.

Lahiri, S., Orr, D., 2018. Biological Control in Tomato Production Systems. Elsevier, p. 253–267. doi:10.1016/b978-0-12-802441-6.00011-5.

Maeda-Gutiérrez, V., Galván-Tejada, C.E., Zanella-Calzada, L.A., Celaya-Padilla, J.M., Galván-Tejada, J.I., Gamboa-Rosales, H., Luna-García, H., Magallanes-Quintanar, R., Guerrero Méndez, C.A., Olvera-Olvera, C.A., 2020. Comparison of convolutional neural network architectures for classification of tomato plant diseases. Appl. Sci. 10 (4), 1245. https://doi.org/10.3390/app10041245.

Maes, W.H., Steppe, K., 2019. Perspectives for remote sensing with unmanned aerial vehicles in precision agriculture. Trends Plant Sci. 24 (2), 152–164. https://doi.org/10.1016/j.tplants.2018.11.007.

Mahlein, A.K., 2016. Plant disease detection by imaging sensors – parallels and specific demands for precision agriculture and plant phenotyping. Plant Dis. 100 (2), 241–251. https://doi.org/10.1094/PDIS-03-15-0340-FE.

Marconi, T.G., Oh, S., Ashapure, A., Chang, A., Jung, J., Landivar, J., Enciso, J., 2019. Application of unmanned aerial system for management of tomato cropping system. In: Thomasson, J.A., McKee, M., Moorhead, R.J. (Eds.), Autonomous Air and Ground Sensing Systems for Agricultural Optimization and Phenotyping IV. SPIE. doi: 10.1117/12.2518955.

Mohanty, S.P., Hughes, D.P., Salathé, M., 2016. Using deep learning for image-based plant disease detection. Front. Plant Sci. 7, 1419. https://doi.org/10.3389/fpls.2016.01419. URL https://www.frontiersin.org/article/10.3389/fpls.2016.01419.

Morais, R., Silva, N., Mendes, J., Adão, T., Pádua, L., López-Riquelme, J.A., Pavón-Pulido, N., Sousa, J.J., Peres, E., 2019. mySense: A comprehensive data management environment to improve precision agriculture practices. Comput. Electron. Agric. 162 (March), 882–894. https://doi.org/10.1016/j.compag.2019.05.028.

Osroosh, Y., Khot, L.R., Peters, R.T., 2017. Economical thermal-RGB imaging system for monitoring agricultural crops. Comput. Electron. Agric. 147(August 2017), 34–43. doi:10.1016/j.compag.2018.02.018.

Petrellis, N., 2018. A review of image processing techniques common in human and plant disease diagnosis. Symmetry 10 (7), 270. https://doi.org/10.3390/sym10070270.

Picon, A., Seitz, M., Alvarez-Gila, A., Mohnke, P., Ortiz-Barredo, A., Echazarra, J., 2019. Crop conditional convolutional neural networks for massive multi-crop plant disease classification over cell phone acquired images taken on real field conditions. Comput. Electron. Agric. 167, 105093. https://doi.org/10.1016/j.compag.2019.105093.

Rangarajan, A.K., Purushothaman, R., Ramesh, A., 2018. Tomato crop disease classification using pre-trained deep learning algorithm. Procedia Comput. Sci. 133, 1040–1047. https://doi.org/10.1016/j.procs.2018.07.070.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al., 2015. Imagenet large scale visual recognition challenge. Int. J. Comput. Vision 115 (3), 211–252. https://doi.org/10.1007/s11263-015-0816-y.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L., 2018. Mobilenetv 2: Inverted residuals and linear bottlenecks. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4510–4520.

Simonyan, K., Vedaldi, A., Zisserman, A., 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. In: Workshop at International Conference on Learning Representations.

Szegedy, C., Liu, Wei, Jia, Yangqing, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A., 2015. Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9.

Tangarife, H.I., Díaz, A.E., 2017. Robotic applications in the automation of agricultural production under greenhouse: A review. In: 2017 IEEE 3rd Colombian Conference on Automatic Control (CCAC), pp. 1–6.

Taqi, F., Al-Langawi, F., Abdulraheem, H., El-Abd, M., 2017. A cherry-tomato harvesting robot. In: 2017 18th International Conference on Advanced Robotics (ICAR), pp. 463–468.

Verma, S., Chug, A., Singh, A.P., 2018. Prediction models for identification and diagnosis of tomato plant diseases. In: 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1557–1563.

Xue, W., Hu, X., Wei, Z., Mei, X., Chen, X., Xu, Y., 2019. A fast and easy method for predicting agricultural waste compost maturity by image-based deep learning. Bioresour. Technol. 290, 121761. https://doi.org/10.1016/j.biortech.2019.121761.

Zhang, L., Jia, J., Gui, G., Hao, X., Gao, W., Wang, M., 2018. Deep learning based improved classification system for designing tomato harvesting robot. IEEE Access 6, 67940–67950.

Zhang, Z., Liu, H., Meng, Z., Chen, J., 2019. Deep learning-based automatic recognition network of agricultural machinery images. Comput. Electron. Agric. 166, 104978. https://doi.org/10.1016/j.compag.2019.104978.

Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V., 2018. Learning transferable architectures for scalable image recognition. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 8697–8710.