electronics

MDPI

Article

Deep Transfer Learning for Image Classification of Phosphorus Nutrition States in Individual Maize Leaves

Manuela Ramos-Ospina [1,*], Luis Gomez [2], Carlos Trujillo [1] and Alejandro Marulanda-Tobón [1,*]

[1] School of Applied Sciences and Engineering, Universidad EAFIT, Medellin 050022, Colombia; catrujilla@eafit.edu.co

[2] Department of Electronic Engineering and Automatic (DIEA), University of Las Palmas de Gran Canaria, 35017 Las Palmas de Gran Canaria, Spain; luis.gomez@ulpgc.es

* Correspondence: mramoso@eafit.edu.co (M.R.-O.); amarula2@eafit.edu.co (A.M.-T.)

Abstract: Computer vision is a powerful technology that has enabled solutions in various fields by analyzing visual attributes of images. One field that has taken advantage of computer vision is agricultural automation, which promotes high-quality crop production. The nutritional status of a crop is a crucial factor for determining its productivity. This status is mediated by approximately 14 chemical elements acquired by the plant, and their determination plays a pivotal role in farm management. To address the timely identification of nutritional disorders, this study focuses on the classification of three levels of phosphorus deficiencies through individual leaf analysis. The methodological steps include: (1) using different capture devices to generate a database of images composed of laboratory-grown maize plants that were induced to either total phosphorus deficiency, medium deficiency, or total nutrition; (2) processing the images with state-of-the-art transfer learning architectures (i.e., VGG16, ResNet50, GoogLeNet, DenseNet201, and MobileNetV2); and (3) evaluating the classification performance of the models using the created database. The results show that the DenseNet201 model achieves superior performance, with 96% classification accuracy. However, the other studied architectures also demonstrate competitive performance and are considered state-of-the-art automatic leaf nutrition deficiency detection tools. The proposed method can be a starting point to fine-tune machine-vision-based solutions tailored for real-time monitoring of crop nutritional status.

Keywords: image classification; computer vision; transfer learning; image database; plant nutrition; leaf analysis

Citation: Ramos-Ospina, M.; Gomez, L.; Trujillo, C.; Marulanda-Tobón, A. Deep Transfer Learning for Image Classification of Phosphorus Nutrition States in Individual Maize Leaves. *Electronics* **2024**, *13*, 16. https://doi.org/10.3390/electronics13010016

Academic Editors: D. J. Lee and Dong Zhang

Received: 28 October 2023
Revised: 10 December 2023
Accepted: 14 December 2023
Published: 19 December 2023

1. Introduction

Today, agricultural production is facing challenges to meet increasing food demand as well as to mitigate the consequences of the gradual reduction in cultivated land area by enhancing agricultural productivity in a sustainable manner [1]. Additionally, there is a pressing need to meet the demand for more effective, more nutritious, and safer food production methods to ensure the well-being of both human health and the planet [2]. Agricultural automation is being raised as a solution that can boost productivity and improve quality and resource-use efficiency [1].

However, generating solutions for agricultural production is a complex task that requires the consideration of several variables. One critical variable is the nutritional status of crops, which is determined by approximately 14 fundamental nutrients that plants require for their growth [3]. Each of these nutrients is found in specific amounts and plays essential roles in crop metabolism. Among these nutrients, nitrogen, phosphorus, and potassium are needed in much more significant quantities [4]. In particular, phosphorus (P) plays a crucial role in various plant processes such as growth, reproduction, flowering, and environmental adaptation. A plant absorbs P in the form of inorganic phosphate (Pi). However, the concentration of Pi in the soil is typically quite low because it tends to strongly bind to the soil surface or form insoluble complexes, rendering more than 80% of it immobile and inaccessible for plant uptake [5]. To maintain high productivity levels, a continuous supply of Pi from fertilizers is required. The contribution of phosphorus, like other nutrients, needs to be carefully regulated according to the specific growth stage of the plant [4]. Therefore, it is crucial to assess and monitor the nutritional status of the crop throughout its entire life cycle. Traditionally, the assessment of nutritional status has relied on visual inspection, which has inherent limitations in terms of precision, as it is primarily a qualitative approach [3]. Alternatively, more accurate methods involve analyzing nutrient concentrations in either leaves or soil. However, these techniques can be costly, as they require not only chemical processes but also the transportation of samples and the interpretation of results [6].

Consequently, many types of technologies have been explored to overcome these problems. Given that nutritional deficiencies primarily manifest through visual characteristics, several explored options are based on automatic methods via image processing [7]. Within these options, artificial vision stands out as a competitive choice due to its versatility and autonomy. Specifically, deep learning techniques employing convolutional neural networks (CNNs) have shown remarkable performance, surpassing traditional approaches based on texture or color analysis of images [8].

The development of deep learning models typically involves a supervised process called end-to-end learning, which relies on known training data to make predictions on unknown data [9]. However, there are several limitations to the applicability of CNN-based methods. Perhaps the most important is the amount of data the network needs to learn the characteristics of the images. Obtaining the required number of high-quality images with accurate labeling is a significant challenge, even more so in the agricultural case, where the field environment is often difficult to access and visual signs of interest are not always present or isolated [6,10]. To address these challenges, a commonly employed strategy is to leverage transfer learning, which involves utilizing pre-trained networks that have been trained on extensive datasets. This technique not only reduces the amount of data and computational cost that is needed to train the network but also allows a model developed for one application domain to be relatively easy to transfer to another [9].

Many works that aim to recognize pathologies on plant leaves use transfer learning as the starting point to develop new models. These works usually propose a comparison between well-developed models and the new model to select the one that performs best for a specific problem. Regarding the recognition of maize diseases, Zhang et al. [11] proposed an improved model based on GoogLeNet and Cifar10 architectures to classify eight disease types using images collected from both the PlantVillage dataset [12] and other image search sites. Similarly, Bhatt et al. [13] classified three disease types with a combination of enhanced models (VGG16, InceptionV2, ResNet50, and Mobilenet), only using the PlantVillage data. Both studies achieved a maximum classification accuracy of 98%. Furthermore, Chen et al. [14] introduced INC-VGGN: composed of a VGGNet enhanced with the Inception module. The network was trained on a field-collected database composed with images of both maize and rice leaves. Results were subsequently compared with other common transfer learning models trained on PlantVillage, and it was shown that the proposed CNN performed the best. Likewise, Zeng et al. [15] classified several diseases of maize using a database acquired with a cellphone and a digital camera. They created a model that integrates the ResNet50 architecture with the SK unit (found in SKNet). The results of their method were then compared with the results of state-of-the-art multiscale network models (InceptionV3, InceptionV4, and Inception-ResNet-V2) and showed that their proposal produced competitive results. On the other hand, Verma and Bhowmik [16] created a new architecture named MDCNN (Maize Disease Detection CNN) and a database composed of publicly available databases and manually acquired leaf images. In this work, the results were also compared to those of several pre-trained networks, with the proposed model achieving the best results.

In the domain of maize leaf nutrition identification using artificial vision, various studies have explored the detection and analysis of nutritional deficiencies. For instance, Zúñiga and Bruno [7] developed a system that relies on texture and color analysis to recognize deficiency levels of essential nutrients such as nitrogen (N), phosphorus (P), potassium (K), magnesium (Mg), and sulfur (S). K-nearest neighbors algorithm, Naive Bayes, and Support Vector Machines (SVM) classifiers were used, and the best results were obtained using SVM, which achieved no more than 82% accuracy. Furthermore, Leena and Saju [3] classified macronutrient deficiency (N, P, and K) using an optimized multi-class SVM, with the highest classification accuracy being 90%. Similarly Guerrero et al. [17], recognized NPK deficiencies on banana leaves by preprocessing the images through linear and color space transformations and using them as input for a VGG16 model, obtaining 98% as the maximum accuracy. Also Jahagirdar and Budihal [18] utilized images with NPK nutrient-deficit maize leaves to train an Inception V3 model, reaching 80% training accuracy. In both cases, authentic datasets were used, but they are not publicly available.

Meanwhile, there are even fewer studies that specifically concentrate on the identification of single-nutrient deficiencies, such as the work conducted by de Fátima da Silva et al. [19], wherein magnesium nutrition was assessed with texture classifiers, reaching a maximum classification accuracy of 75%. In addition, Condori et al. [8] detected levels of nitrogen deficiency by comparing texture and transfer learning models. The main conclusion of this work was that the results of CNN-based models outperform those of texture methods in the majority of experiments.

In the realm of datasets utilized for studying nutrient deficiencies in maize leaves, Peng et al. [20] stands out as a pioneering contribution. The dataset in [20] comprises UAV images characterized by extensive spatial coverage and long time series purposefully crafted for distribution analysis of maize in China. Additionally, [21] presents another noteworthy contribution, wherein images were sourced from well-known leaf databases, systematically curated, and categorized based on four common diseases. A distinctive feature of this work is the absence of self-generated images in the referenced dataset. As of our current understanding, there is no publicly available maize leaf database explicitly addressing single-nutrient deficiency, particularly phosphorus.

In the current research landscape, there is a conspicuous absence of studies exclusively dedicated to the classification of phosphorus deficiency in maize through the utilization of various deep transfer learning techniques. Furthermore, the absence of a well-established and publicly available database focused on this specific topic exacerbates the existing research gap. In light of these considerations, the primary objective of this study is to fill this void by addressing recent advancements in deep learning techniques. Specifically, our focus is on the classification of images derived from controlled environments featuring maize leaves that exhibit varying degrees of phosphorus deficiency: the complete absence of the nutrient, a half dose of the required phosphorus, and an adequate supply of phosphorus.

The structure of this work is as follows: Section 2 provides an overview of the process involved to build the dataset and details the transfer learning approach utilized. In Section 3, the results obtained from applying the transfer learning models to the created dataset are thoroughly reported. The paper finishes with Section 4.

2. Materials and Methods

The workflow employed in this study, approaching the use of deep learning techniques to classify three levels of phosphorus deficiency in maize leaves, is illustrated in Figure 1.

Firstly, we begin with a data preparation stage, which involves the collection, labeling, preprocessing, and splitting of data. This stage ends with the labeled samples divided into three data sets: train, validation, and test. In the second stage, a set of pre-trained models is chosen and implemented in MATLAB version 9.9.0 (R2020b), The MathWorks Inc., Natick, MA, USA. One is selected for a fine-tuning stage, for which the inputs are the training and validation sets and the output is a trained model. Hence, the fine-tuned model is used to classify new images from the test set. The prediction results are then evaluated with classification metrics, and the next model is chosen from the aforementioned set to restart the second stage. Once all pre-trained models have been tested, a comprehensive performance evaluation is conducted based on the metric scores, thereby concluding the workflow.

The following subsections present the procedures' details, providing a full overview of their specific information and methodologies.

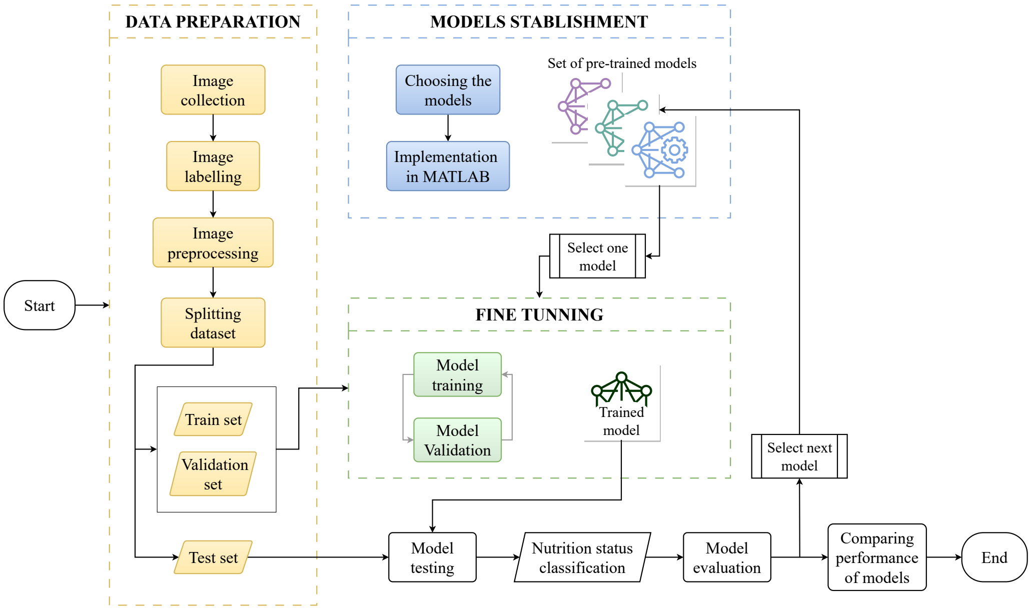


Figure 1. Workflow for phosphorous deficiency detection.

2.1. Dataset Building

The images of nutrition-deficient maize leaves (*Zea mays* L. improved variety ICA—V 109) used in this study were collected from mid-June to early August 2022 in a plastic shed from the area of Natural Systems and Sustainability of Universidad EAFIT, Medellin, Colombia (6°11′53.80″ N, 75°34′43.23″ W). The experimental design followed a 3 × 10 scheme comprising ten replications of three phosphorus levels: P absence (-P), half dose (-P50), and complete supply (C), resulting in a total of 30 plants (see Figure 2).

To induce the phosphorus deficiency levels, Hoagland's complete solution [4] was modified by taking into account only macronutrients and adjusting the net contribution of each nutrient according to the concentration of minerals in the solution.

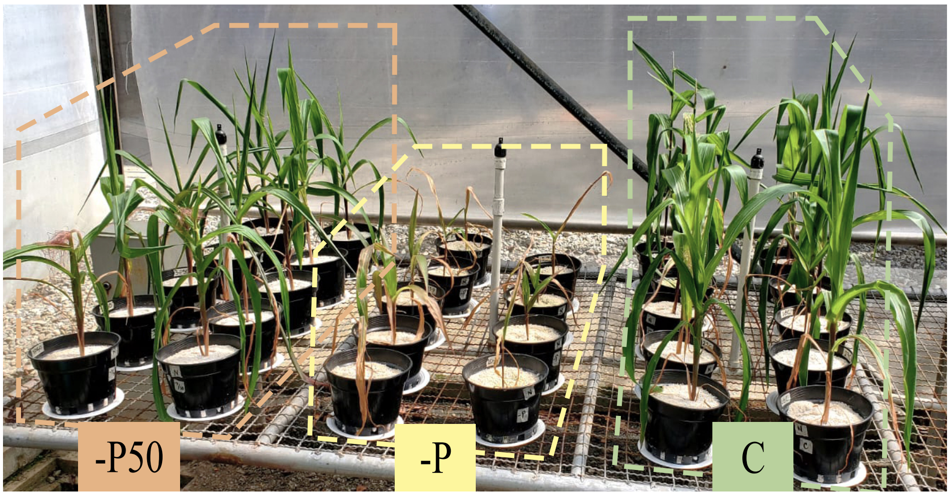


Figure 2. Location of experiments and treatment differentiation.

2.1.1. Image Collection

A total of 3934 images were acquired. Photographs included the growth stages of seedling, jointing, and flowering. The experiment involved natural illumination. Both sunny and cloudy days were considered in order to increase diversity in the illumination conditions.

Five acquisition devices were utilized, encompassing two types of regular smartphones, a digital camera, a single-lens reflex camera, and a compact scientific camera. In Table 1, the specifications of the tested cameras are presented.

Table 1. Specifications of cameras used to acquire original images.

Camera Type	Smartphone 1: Xiaomi Redmi 8T	Smartphone 2: Moto G (5) Plus	Digital	Single-Lens Reflex	Compact Scientific
Manufacturer	Omnivision	Motorola	Samsung	Nikon	ThorLabs
Model	OV02A10	Unknown	ES65	D3100	DCC1645C-HQ
Sensor type	CMOS	Unknown	CCD	CMOS	Color CMOS
Number of Active Pixels	1200 × 1200	3264 × 2448	2048 × 1536	3456 × 2304	1280 × 1020
Resolution (ppp)	96	72	96	300	144
Optical Format	0.2″	0.4″	0.24″	1.09″	0.33″
Maximum Aperture f /	2.4	1.7	3.5	3.8	1.4

Nevertheless, previous experiments have determined that images captured by the scientific camera consistently yield superior classification performance. The outcomes of image classification using the GoogLeNet architecture for each camera type are provided in Appendix A. Consequently, this study exclusively concentrates on the dataset comprising images acquired solely by the scientific camera.

The image collection process was conducted according to the following steps: (1) One leave per plant exhibiting prominent visual symptoms, predominantly observed in older leaves, was selected for sampling. Specifically, the mid-leaf area, as depicted in Figure 3, was the focal region of interest. (2) A white background sheet was carefully positioned: the intent was to prevent the formation of shadows caused by the leaf and to minimize background-related noise. (3) The leaf was securely held, and a total of five photographs were captured for each leaf. Either the capture angle or the leaf section was adjusted between each shot, ensuring diverse perspectives. An illustration of this process is presented in Figure 4.

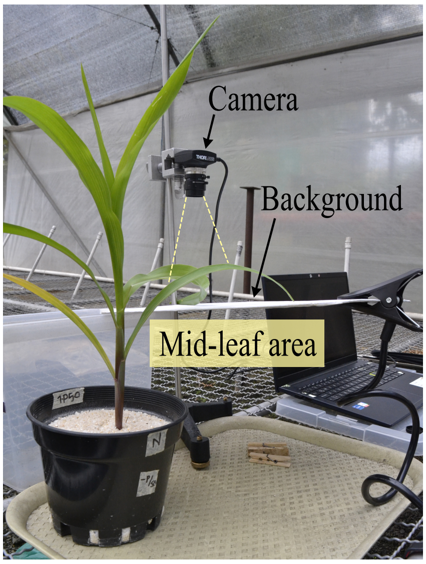
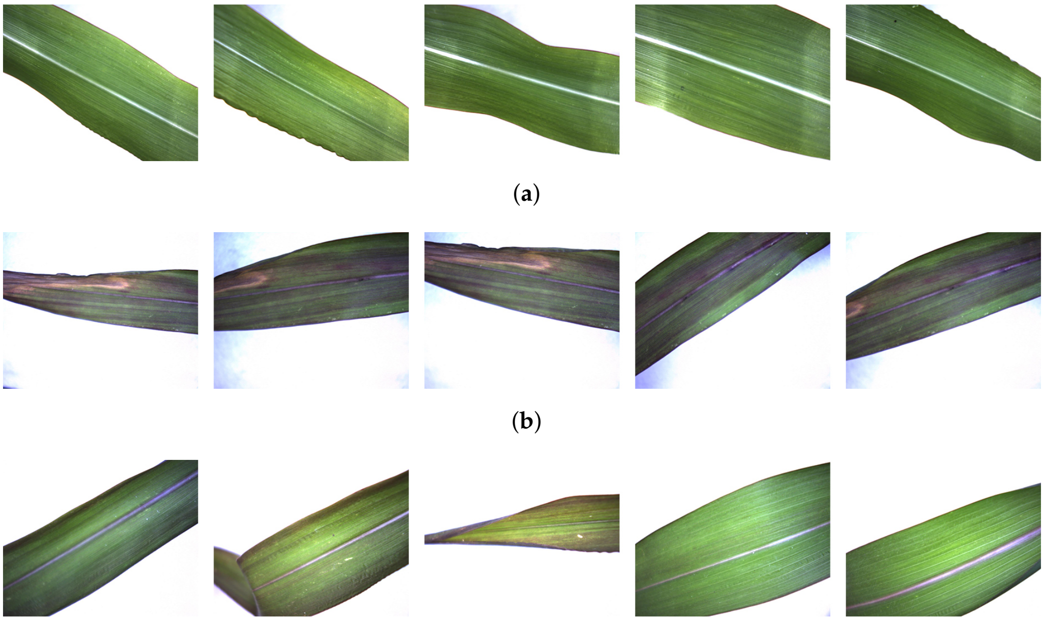


Figure 3. Illustration of an image acquisition example. The camera, the background sheet, and the leaf capture area are depicted.

The resulting images were saved and labeled according to the treatment and growth stage. Examples of images obtained using this method are shown in Figure 4.



(**a**)

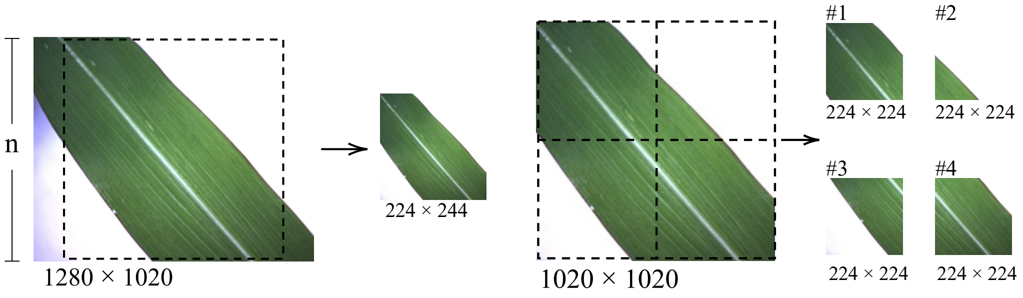
(**b**)

(**c**)

Figure 4. Example of five consecutive images taken of three different leaves with: (**a**) complete nutrition (C), (**b**) no phosphorus nutrition (-P), and (**c**) half-phosphorus nutrition (-P50).

2.1.2. Image Pre-Processing and Data Augmentation

The original images obtained with the scientific camera underwent automatic size processing using Python (version 3.8.8, Python Software Foundation, Wilmington, DE, USA) code with two concurrent methods: (1) All 1280 × 1020 pixel original images were cropped to a central square with sides equal to the smallest image side (n), i.e., 1020 px. Then, cropped images were resized to 224 × 224 pixels according to the method shown in Figure 5a. (2) All images cropped to a central square were subsequently divided into four individual images with a size of 510 × 510 pixels each. Similarly, these cropped images were resized to 224 × 224 pixels. The process is shown in Figure 5b.



(**a**)

(**b**)

Figure 5. Pre-processing based on cropping and resizing images using two methods: (**a**) crop to a centered square and (**b**) quadrant division. ’#1’ to ’#4’ are the resulting images after quadrant division.

After the above process, the number of images increased five-fold. However, the automated cropping mechanism introduced certain issues, including producing blank images or images capturing only a small portion of the leaf, which led to images with limited or irrelevant content. An example of this is seen in image #2 of Figure 5b. Based on supplementary experiments presented in Appendix B, these images introduce confusion to the neural network, hindering the accurate extraction of pertinent features and consequently resulting in a decline in performance metrics. To address this problem, an algorithm was developed to select valid images. The algorithm involved the following steps: First, the image was split into its RGB components. Based on the histogram analysis of the images, we determined that the blue (B) channel provided more contrast to distinguish the leaf from the background, so only the B channel was preserved. Next, a thresholding process was applied to distinguish leaf pixels (set to 255) from the background (set to 0). The algorithm then counted the number of leaf pixels and considered a minimum count of 15k pixels as indicative of a significant leaf presence. Finally, images with a pixel count below this threshold were excluded from further analysis. The effectiveness of the filtering process is illustrated in Figure 6.

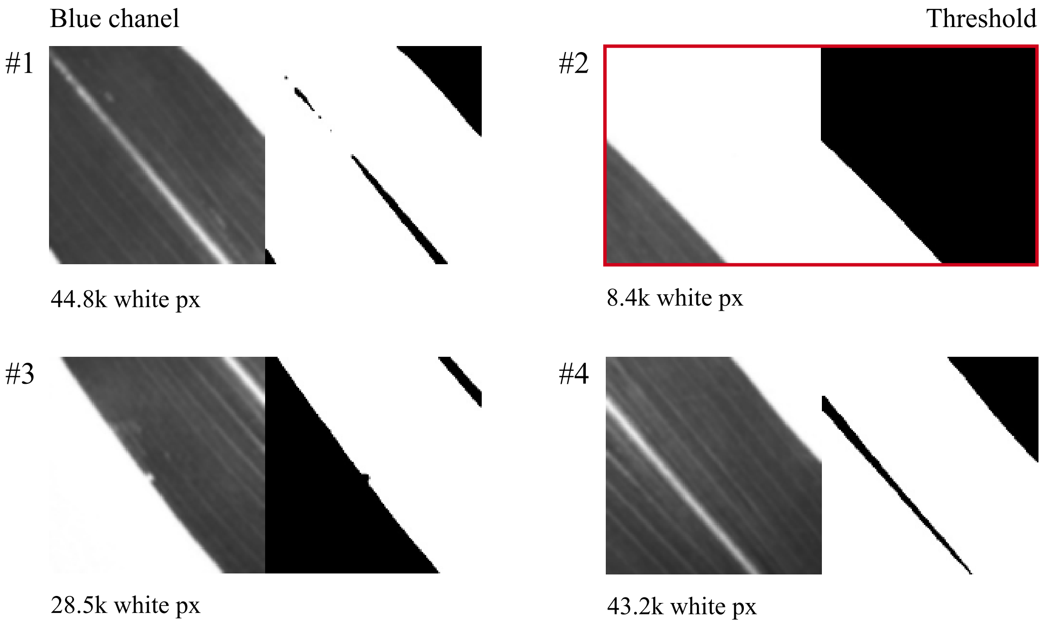


Figure 6. Selection process for images obtained through quadrant division. Each image shows the separated blue channel (**left**) and the result of thresholding (**right**). The number of white pixels at the bottom of each image represents the leaf content. A minimum count of 15k white pixels is considered as the threshold for determining the presence of relevant information. '#1' to '#4' are the resulting images after quadrant division. In this example, only image #2 would be filtered out.

Following the preprocessing and data augmentation procedures, the resulting dataset contained the number of images indicated in Table 2.

Table 2. Dataset details.

Class	Description	Number of Images
-P	No phosphorus nutrition	656
-P50	Half of complete phosphorus nutrition	850
C	Complete nutrition	927

Finally, the training, validation, and testing image sets were composed with a ratio of 7:2:1 and in such a way that the five subimages obtained from each original image belonged to a single set, thus ensuring the independence between the image sets. The correspondence of the totals of the images for each set is detailed in Table 3.

Table 3. Dataset division.

Image Set	Number of Samples
Total	2433
Train (70%)	1703
Validation (20%)	487
Test (10%)	243

2.2. Transfer Learning Approach

A deep learning approach is employed to classify the three levels of phosphorus deficiencies. Given the challenges associated with acquiring an ample supply of images and the potential scarcity of publicly available datasets for training convolutional neural networks (CNNs), it is a common practice to adopt transfer learning. Transfer learning is a powerful machine learning technique that involves repurposing an existing trained model for a new—often related—problem. This approach capitalizes on the capability of the initial layers in the original model to detect general features. Subsequently, the output of the last layer is adapted to the specific requirements of the new task. This adjustment is achieved by replacing the last fully connected layer with a new one representing the classes relevant to the new problem. Additionally, it is possible to fine-tune the transfer learning process by selectively freezing or updating specific weights in the initial layers [22].

The models used for transfer learning in this study are primarily associated with the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC) [23], which has produced some of the most accurate models. These models have served as inspiration for numerous versions and improvements, as well as being the foundation for other models. Considering the existing literature, five architectures were selected based on their frequent utilization and high accuracy. Therefore, the following models were included in this study.

2.2.1. VGG16

These models were introduced in 2014 by Oxford's Visual Geometry Group [24] but are still popular today. The VGG networks consist of multiple blocks of stacked convolutional layers with smaller filters (i.e., 3 × 3 layers) combined with a max-pooling and another fully connected layer. This set of layers is used instead of a larger filter size (such as 7 × 7) in order to increase efficiency and make the decision function more discriminative [22]. The latter ultimately means that this model type generalizes well to a wide range of tasks [24]. One of the most popular variants is VGG16, which is composed of 16 layers in weight and is available pre-trained on the ImageNet dataset [25].

2.2.2. ResNet50

Residual networks were first introduced in 2015 by He et al. [26] and consist of blocks with two or three sequential convolutional layers with a parallel but separate identity layer that connects the input of the first layer to the output of the last one [22]. These identity layers, called 'skip connections', solve errors generated during training and testing when the model goes deeper. Furthermore, they can mitigate the vanishing gradient problem when placed before the activation function [27]. This study utilizes ResNet50, one of the evolved versions of ResNet. ResNet50 is chosen as it is a 50-layer deep architecture known for its remarkable performance and effectiveness at various tasks.

2.2.3. GoogLeNet

The GoogLeNet model is a special manifestation of the Inception architecture. This type of block splits the input into parallel and multiple pillars containing convolutional layers with a different-sized filter and a pooling layer. Those are followed or preceded by a downsampling convolution to reduce the output depth, which is finally concatenated. This enables the saving of computing resources [22]. The GoogLeNet structure uses nine

Inception modules accompanied by pooling, regularization, and fully connected layers. For additional information, refer to the original paper [28].

2.2.4. DenseNet201

The creators of Dense Convolutional Network (DenseNet) [29] took some inspiration from the residual network's idea to introduce dense blocks. These are modules of sequential convolutional layers for which any layer has a connection to every other layer in a feed-forward way in terms of concatenation operation. In this way, successive layers receive information from preceding ones, including feature maps, for better feature propagation and reuse. This process causes the number of channels to grow, despite reducing the number of parameters as compared to conventional CNN [30]. Three versions are highlighted: DenseNet121, DenseNet169, and DenseNet201, which are differentiated by the number of layers. The latter is used in this study.

2.2.5. MobileNetV2

MobileNet was first introduced by Howard et al. [31] using the concept of depthwise separable blocks and consists of (1) depthwise convolution: performing a single convolutional filter per input channel, followed by (2) pointwise convolution: computing a linear 1 × 1 convolution of the input channels. Later, Sandler et al. [32] improved the original version by incorporating bottleneck blocks between input and output layers; bottleneck blocks are similar to residual connections but are considerably more memory efficient [33].

2.3. Model Implementations

The transfer learning models are implemented using MATLAB's Deep Learning Toolbox™ release R2020a, The MathWorks Inc., Natick, MA, USA [34]. This package provides access to the pre-trained models mentioned earlier, which have been specifically trained on the ImageNet dataset. Some specifications of these models are presented in Table 4.

Table 4. Parameters of pre-trained CNN models.

Model	Depth (Layers)	Total Parameters (in Millions)	Size (MB)	Birth Year
VGG16	16	138.0	515	2014
ResNet50	50	25.6	96	2015
GoogLeNet	22	7.0	27	2014
DenseNet201	201	20.0	77	2017
MobileNetV2	53	3.5	13	2018

The computer code is executed on a machine equipped with an i7-9700K 3.6 GHz processor, 64 GB RAM (Intel, Santa Clara, CA, USA), and an NVIDIA GeForce RTX 2080 40 GB GPU (Santa Clara, CA, USA). To apply the transfer learning approach for each model, the following process is performed (see Figure 7):

(1) Data with its ground-truth labels are read. The five sub-images are randomly chosen to form the train set with 70% of available samples, 20% for the validation set, and the remaining 10% images as a test set, ensuring the independence of sets. (2) Each model is loaded separately, and its initial layers are frozen to reuse the already learned general features. Moreover, the last fully connected layer is substituted to match the three classes' outputs. (3) Hyperparameters are predefined with specific values as outlined in Table 5. These hyperparameters control various aspects of the deep learning model and its training process. The details and explanations of these hyperparameters are provided in the following paragraph. (4) The training process involves training each model on the training set and validating it at each epoch using the validation set. The training continues until either the 'maximum number of epochs' requirement is met or the validation patience is satisfied. (5) The fine-tuned model is used to classify new images from the testing set. Consequently, the predicted labels are obtained. (6) The predicted labels are compared with the ground-truth labels to validate the model's performance.

In Table 5, the Solver represents the optimizer used for the loss function, which is stochastic gradient descent with a momentum of 0.9. The batch size indicates the number of images processed by the network in each batch for error computation and weight updates. The initial learning rate is used at the beginning of training and decreases by 0.96 per epoch in a stepwise manner. To prevent overfitting, the training process considers both a maximum number of epochs and a validation patience method. The validation patience method monitors the validation error for consistent behavior within a certain number of epochs to determine when to stop training.

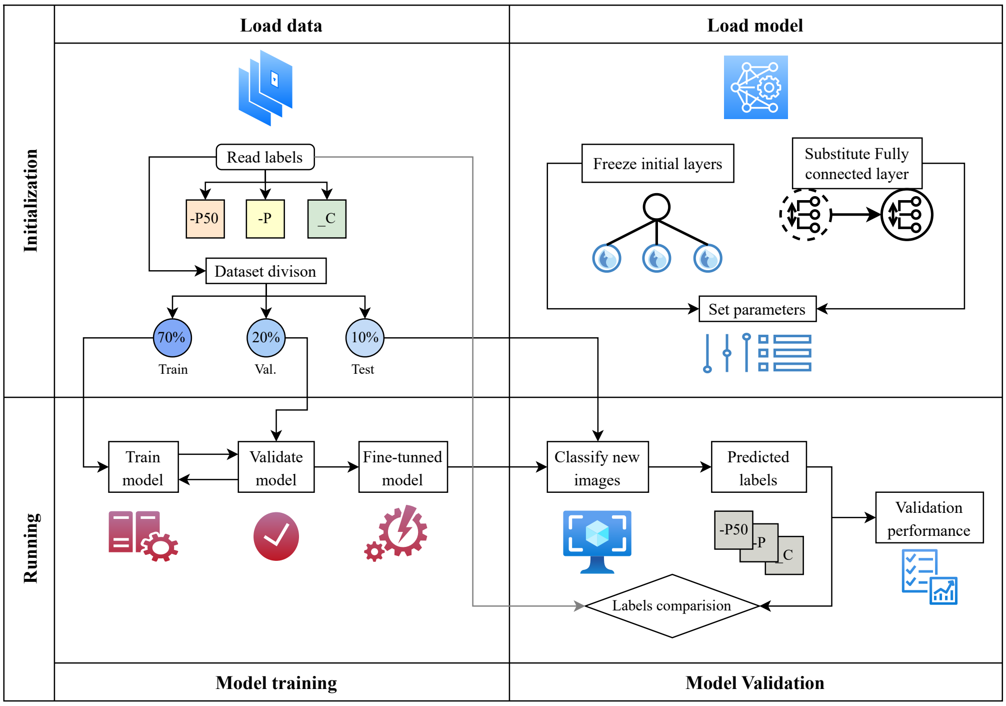


Figure 7. Framework of transfer learning approach for each model.

Table 5. Hyperparameter specifications.

Solver	SGDM
Momentum	0.9
Batch size	32
Initial learning rate	0.001
Learning rate policy	Step
Learning rate decay	0.96
Decay period	1 epoch
Max epochs	30
Validation patience	8

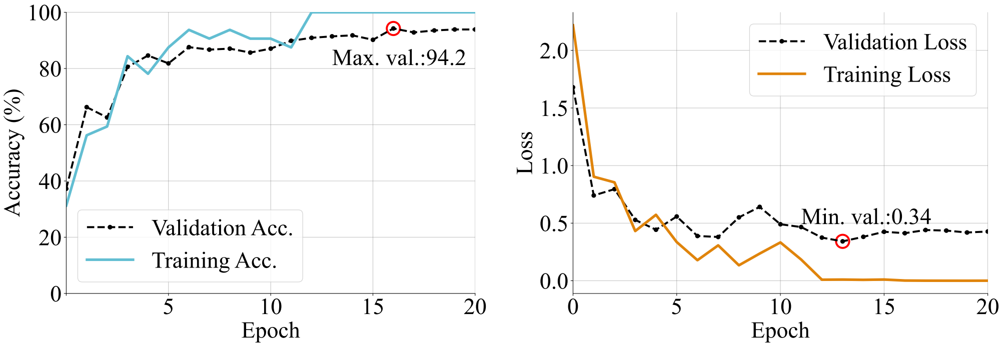
The hyperparameters presented in Table 5 were determined based on findings reported in the existing literature for similar studies: Mohanty et al. [35], Barbedo [36], Zhang et al. [11], Maeda-Gutiérrez et al. [37], and Nagaoka [38]. These values have been widely used and are recognized as effective choices for achieving good performance using deep learning models.

3. Results

The proposed transfer learning approach was employed to classify three levels of maize leaf phosphorous deficiency using the aforementioned deep learning models (VGG16, ResNet50, GoogLeNet, DenseNet201, and MobileNetV2). The following subsections describe the evolution of accuracy and loss values during the training stage and present the results obtained to evaluate the overall performance of the studied models on the dataset built specifically for this study.

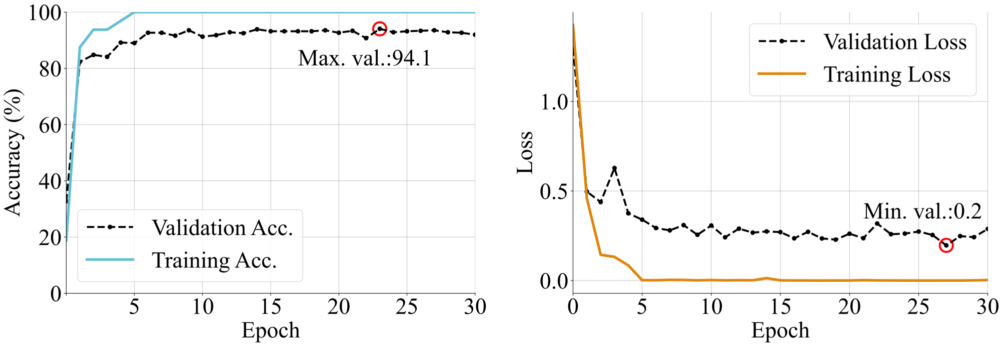
3.1. Learning Curves

To evaluate the training performance, accuracy and loss curves are examined for each epoch. Figures 8–12 depict the training progress with the training set and visual represent how well each model learns. On the other hand, the validation curves in those figures provide insight into how well the models generalize.



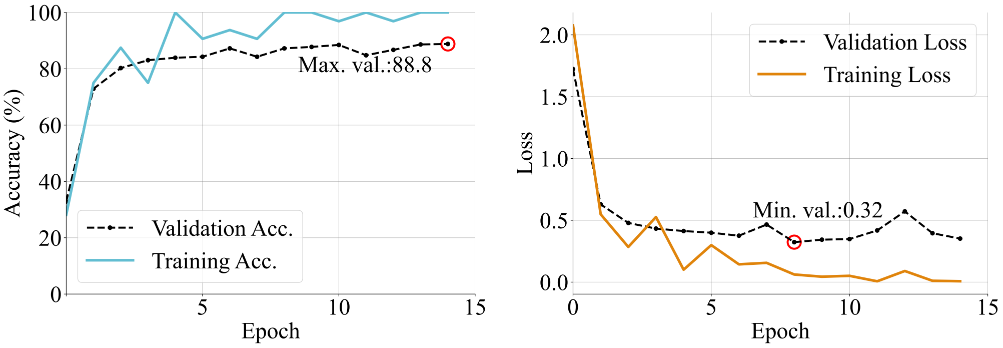
(**a**) Accuracy curve. (**b**) Loss curve.

Figure 8. Training and validation curves for the VGG16 architecture specifying (**a**) accuracy and (**b**) loss in every epoch. The maximum accuracy value and the minimum loss are also reported.



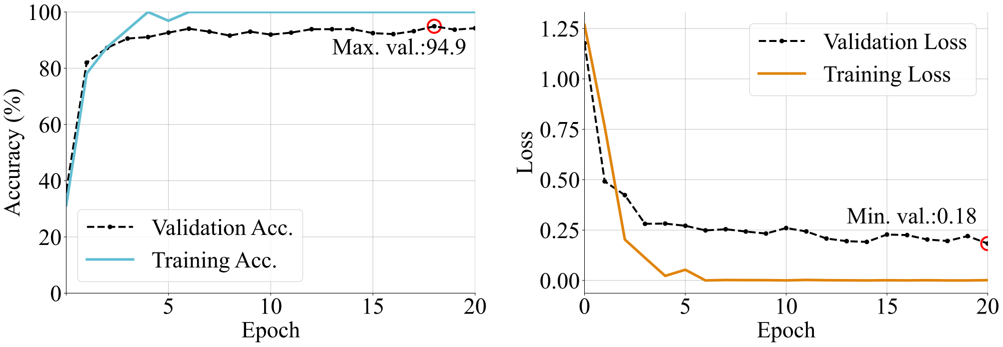
(**a**) Accuracy curve. (**b**) Loss curve.

Figure 9. Training and validation curves for the ResNet50 architecture specifying (**a**) accuracy and (**b**) loss in every epoch. The maximum accuracy value and the minimum loss are also reported.



(**a**) Accuracy curve. (**b**) Loss curve.

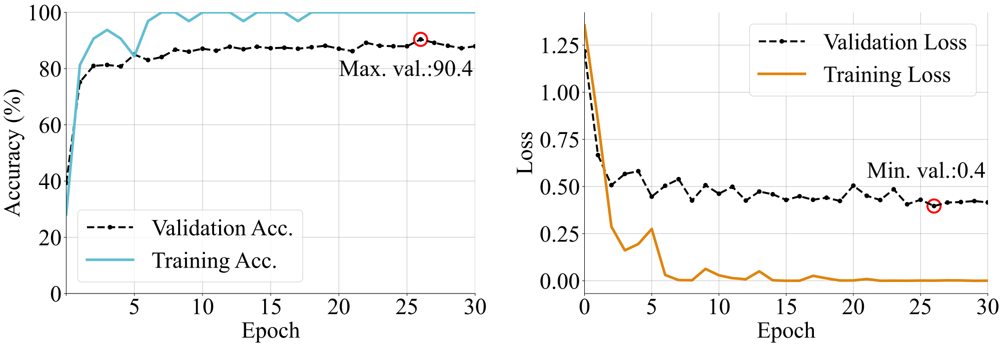
Figure 10. Training and validation curves for the GoogLeNet architecture specifying (**a**) accuracy and (**b**) loss in every epoch. The maximum accuracy value and the minimum loss are also reported.



(**a**) Accuracy curve.

(**b**) Loss curve.

Figure 11. Training and validation curves for the DenseNet201 architecture specifying (**a**) accuracy and (**b**) loss in every epoch. The maximum accuracy value and the minimum loss are also reported.



(**a**) Accuracy curve.

(**b**) Loss curve.

Figure 12. Training and validation curves for the MobileNetV2 architecture specifying (**a**) accuracy and (**b**) loss in every epoch. The maximum accuracy value and the minimum loss are also reported.

Each model was run for 30 epochs, and it was found that at around ten epochs, the models started to converge with high accuracy. Specifically, DenseNet, VGG, and ResNet achieved more than 94% validation accuracy. They were followed by MobileNet, which obtained an accuracy of more than 90% on the validation sets. In the same way, as is expected, losses seem to be lower as accuracy increases and had values ranging from 0.18 to 0.4.

In addition, model behavior can be diagnosed by the shape of the learning curves. One common dynamic that can be concluded by observing the graphs is overfitting. Overfitting refers to a model that has learned the training dataset too exactly. This causes it to be less able to generalize unseen data. Based on Figure 9, the ResNet loss validation curve continues to increase after a minimum validation point. Similarly, the GoogLeNet loss curve (Figure 10) has two peaks where the loss increases, causing training to stop at only 14 epochs.

Another consideration to be seen on the learning curves is the gap between the validation and training loss curves, which indicates an insufficient dataset size. It can be observed on Figure 12 that the MobileNet loss curve has a more considerable gap distance, followed by ResNet and VGG (Figures 8 and 9, respectively).

Finally, the most consistent performance is obtained by the DenseNet model in Figure 11 since both curves reach a point of stability with a minimal gap between the final values. In addition, the training stops in 20 epochs, indicating good learning and generalization of features in the images.

3.2. *Performance Analysis*

Once each model is trained, it can further be used to infer features of interest in unknown data to test its generalization. In order to both assess the effectiveness of the studied models and to determine the superiority of one model over the others, four performance metrics were utilized, as described below:

- Accuracy: This is the most common classification metric. This metric describes the ratio between the number of correct predictions and the size of the data. The metric is defined in Equation (1).

$$\text{Accuracy} = \frac{\text{Total correctly classified samples}}{\text{Dataset size}} \tag{1}$$

- Precision: This is a performance metric that measures the proportion of correct predictions for a specific class out of all the predictions made by the model for that class. It provides insight into the model's ability to accurately classify instances for a particular class, regardless of the overall accuracy. Precision focuses on the relevance of the model's predictions compared to the actual ground truth. This metric is defined in Equation (2).

$$\begin{aligned}\text{Precision} &= \frac{\text{Correctly classified samples by class}}{\text{correctly classified samples} + \text{incorrectly classified samples}} \\ &= \frac{\text{Correctly classified samples by class}}{\text{Total predictions by class}}\end{aligned} \tag{2}$$

- Recall: Also known as sensitivity or true positive rate, recall is a performance metric that measures the proportion of correctly predicted instances for a specific class out of all the instances that actually belong to that class. It quantifies the model's ability to identify and capture the positive instances, or true positives, in relation to the actual ground truth. Recall emphasizes the model's capability to recognize and recall the relevant instances of a particular class, without considering the incorrect predictions. This metric is computed as presented in Equation (3).

$$\text{Recall} = \frac{\text{Correctly classified samples by class}}{\text{Number of samples by class}} \tag{3}$$

- F1-score: The F1-score is a performance metric that combines precision and recall into a single value by taking their harmonic mean. By incorporating both precision and recall, the F1-score provides a comprehensive evaluation of the model's ability to achieve both high precision and high recall, promoting a balanced trade-off between the two measures. The metric is defined by Equation (4).

$$\text{F1-score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{4}$$

Since the metrics of precision, recall, and F1-score are performance measures for n classes, there are different ways to combine these scores to have an overall value. One way to do this is to calculate the simple arithmetic mean, which is known as the macro-averaged score and is defined by Equation (5). With this technique, all classes contribute equally to the final averaged metric.

$$\text{Macro-averaged score} = \frac{\text{Class}_1\ \text{score} + \cdots + \text{Class}_n\ \text{score}}{\text{Total of classes}} \tag{5}$$

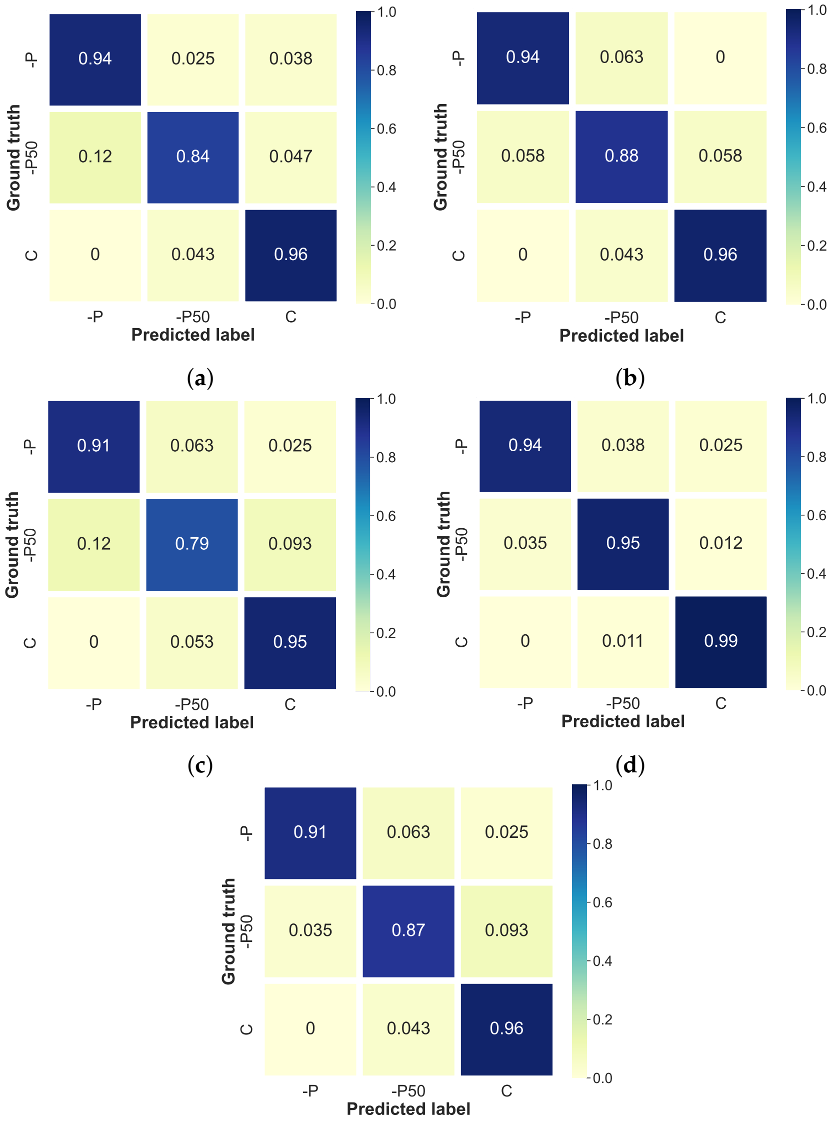
Table 6 presents a comparison of the performance metrics, including macro-averaged precision, recall, F1-score, and accuracy, on the testing set. DenseNet is the model with the best results and is highlighted.

In these terms, GoogLeNet obtained the lowest scores, followed by the VGG and MobileNet architectures. As was discussed based on observing the learning curves, these models had problems with training in terms of overfitting and insufficient dataset size. Both aspects would negatively impact the model's performance. In the same way, the most consistent training was done by DenseNet and ResNet, and this is depicted by their high performance.

Table 6. Comparative performance analysis of validation macro-averaged metrics for each model.

Network Model	Accuracy	Precision	Recall	F1-Score
DenseNet201	**96.1**	**96.1**	**96.0**	**96.0**
MobileNet	91.5	91.8	91.4	91.5
ResNet50	92.7	92.6	92.6	92.6
GoogleNet	88.4	88.3	88.3	88.2
VGG16	91.1	91.1	91.1	90.9

To finish the evaluation of the studied models, the confusion matrix is used. This tool records all the predictions made on the test set, allowing the visualization of the performance for each class. On one side of the matrix, the ground truth is arranged against predictions of the model. The confusion matrices for all models are shown in Figure 13.



(**a**) (**b**) (**c**) (**d**) (**e**)

Figure 13. Normalized confusion matrices to evaluate the accuracy of prediction results for: (**a**) VGG16, (**b**) ResNet50, (**c**) GoogLeNet, (**d**) DenseNet, and (**e**) MobileNet models.

Based on the graphics, it can be seen that in almost all cases, the prediction of the $-P50$ class has the lowest performance values, except for DenseNet (Figure 13d), which has lower recognition rates for the $-P$ label. Concerning this architecture, the color map shows high homogeneity of correct classifications for all classes, i.e., this model has no strong inclination to recognize one class more than another. In the opposite case, it can be seen in Figure 13c that the GoogLeNet model has a classification weakness for the $-P50$ label (79% accuracy), although all other classes are identified with good accuracy (both higher than 90%). This same behavior is traced by MobileNet in Figure 13e but with a slightly higher accuracy rate. Finally, both the VGG and ResNet models had similar behavior (Figure 13a and 13b, respectively).

This difficulty in recognizing the $-P50$ label is observed in almost all models and is explained by the overlap of visual characteristics between this class and the other two. This makes it as difficult for a human as it is for a machine to recognize the differences between a leaf with sufficient nutrition or low nutrition and a leaf with medium nutrition.

In addition to nutrition evaluation being relevant to ensure good agricultural production, other issues, such as maize diseases, are also important and been studied using the same deep learning framework, so it is possible to compare our results to the results obtained by other studies from the literature; this comparison is presented in Table 7. It was observed from the analysis that this work places within the state-of-the-art results and also that only a few studies have attempted to acquire their own images.

Table 7. Results comparison with other studies from the literature.

Reference	Dataset	Multi-Classes	Pre-Trained Model	Metric	Value
[14]	PlantVillage	4	VGG19 ResNet50 DenseNet201 Proposed "INC-VGGN"	Training accuracy	74.20 70.41 84.13 97.57
[16]	Self-created	4	VGG16 ResNet50 DenseNet169 Proposed "MDCNN"	Test Accuracy	97.35 99.21 99.51 99.54
[15]	Self-created	6	VGG16 ResNet50 Proposed "SKPSNet-50"	Average F1-Score	81.4 82.5 91.9
[11]	Various sources	9	Improved GoogLeNet Improved Cifar10	Test accuracy	98.8 97.1

4. Conclusions

The detection and identification of plant leaf issues is a relevant task in farm management. The care of each plant leads to a healthy plantation, which results in high production and excellent quality. Despite the development of many deep learning methods for the classification of plant diseases, including leaf nutrition deficiency, they do not respond the same to all situations. For this reason, there is an existing need to test deep learning model performance for specific tasks, as learned features are extracted uniquely depending on image characteristics.

In this study, five transfer deep learning architectures pre-trained on the ImageNet database (i.e., VGG16, ResNet50, GoogLeNet, DenseNet201, and MobileNetV2) were trained to classify three phosphorus deficiency levels. The training was conducted on a self-made database comprising images taken by five different acquisition devices (but just one camera's images were selected for this analysis). It was found that DenseNet201 performed the best for this specific problem: giving the most consistent training performance and best recognition metrics and leading with an overall accuracy score of 96.1% as well as correct prediction rates uniformly distributed among all classes. This performance is placed within the state-of-the-art results, so further investigations can be done focusing on the performance of this architecture against more current models.

The second best-behaved model was ResNet50: reaching an accuracy of 92.7% but with better recognition rates for $-P$ and C labels than for those with $-P50$. Finally, GoogLeNet and VGG16 models had lower overall accuracy (88.4% and 91.1%, respectively). The first, probably due its huge number of layers, either does not learn enough features from the database or requires more data to correctly adjust the network weights. Meanwhile, the dataset is possibly unrepresentative for the second model, judging by the shape of the loss curve.

Moreover, the analysis of learning curves supports this hypothesis and allows us to understand that for most architectures, either the amount or the quality of training data is not sufficient for the models to make a perfect generalization of the features in the images. Therefore, it is necessary to increase the size of the database in the future. In the same way, different regularization techniques can be explored to avoid overfitting.

This study generates a comprehensive evaluation of the performance of the mentioned models that contributes to the understanding of deep learning models applied to detection of single-nutrient deficiency on plants. This also contributes to faster and economical identification of nutritional phosphorus issues so that a crop's fertilization schedule can be focused on specific plants and thus make more rational use of resources, taking care of both the farmer's budget as well as the health of the environment. We remark that the setup proposed in this research can be extended easily to real-time monitoring of other crop types and even to analyzing different kinds of leaf issues that can be inferred from visual inspection.

In future work, we aim to explore the development of a novel model directly from our dataset. This endeavor will involve experiments without pre-training to assess the model's performance and the potential advantages of a more specific dataset tailored to address agricultural challenges. Additionally, we will investigate the implications of this approach for broader applications in similar agricultural problems.

Author Contributions: Conceptualization, M.R.-O. and A.M.-T.; methodology, investigation, formal analysis, software, and visualization, M.R.-O.; validation, M.R.-O., A.M.-T., L.G. and C.T.; resources, A.M.-T. and C.T.; Project administration, supervision, and funding acquisition, A.M.-T.; writing—original draft, M.R.-O.; writing—review and editing, A.M.-T., C.T. and L.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Universidad EAFIT (grant number 1059-000027).

Data Availability Statement: The original imageset and the preprocessed dataset presented in this work are freely available at Zenodo at https://zenodo.org/records/10279042 (accessed on 7 December 2023). Also, the codes used to preprocess images as well as the MATLAB scripts are available at https://zenodo.org/records/10290643 (accessed on 7 December 2023).

Acknowledgments: M. Ramos-Ospina and A. Marulanda-Tobón acknowledge the support from María Isabel Hernández-Pérez, Head of the Undergraduate Program in Agricultural Engineering, School of Applied Sciences and Engineering, Universidad EAFIT, for the scientific assistance provided during the realization of this work.

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

Appendix A. Previous Results Using Different Camera Types

Table A1. Results of classification of images using GoogLeNet architecture according to camera type.

Metric	Accuracy	Average Precision	Average Recall
Smartphone 1	0.84	0.84	0.84
Smartphone 2	0.91	0.92	0.92
Digital	0.80	0.81	0.80
Reflex	0.92	0.92	0.92
Scientific	**0.93**	**0.92**	**0.93**

Appendix B. Previous Results of Model Behavior with and without Images with Spurious Content

The GoogLeNet model was trained with both a database containing all the preprocessed subimages and with another database wherein the blank images were removed. The results are listed in Table A2 and show a significant decrease in performance when the blank images are included.

Table A2. Results of classification of two databases using GoogLeNet architecture: Comp. stands for the database with all images, and Drop. means the database with blank images removed.

Metric	Accuracy	Class Precision			Mean	Class Recall			Mean
		-P	-P50	C	Precision	-P	-P50	C	Recall
Comp.	0.91	0.91	0.88	0.94	0.91	0.86	0.91	0.95	0.91
Drop.	0.93	0.91	0.91	0.96	0.92	0.95	0.91	0.92	0.93

References

1. FAO. *The State of Food and Agriculture 2022. Leveraging Automation in Agriculture for Transforming Agrifood Systems*; FAO: Rome, Italy, 2022. [CrossRef]
2. Tian, H.; Wang, T.; Liu, Y.; Qiao, X.; Li, Y. Computer vision technology in agricultural automation—A review. *Inf. Process. Agric.* **2020**, *7*, 1–19. [CrossRef]
3. Leena, N.; Saju, K.K. Classification of macronutrient deficiencies in maize plants using optimized multi class support vector machines. *Eng. Agric. Environ. Food* **2019**, *12*, 126–139. [CrossRef]
4. Taiz, L.; Zeiger, E. *Plant Physiology*, 4th ed.; Sinauer Associates, Inc.: Sunderland, MA, USA, 2006; Volume 1, pp. 1–764.
5. White, P.J.; Hammond, J.P. *The Ecophysiology of Plant-Phosphorus Interactions*; Springer: Dordrecht, The Netherlands, 2008; Volume 7, p. 296. [CrossRef]
6. Barbedo, J.G.A. Detection of nutrition deficiencies in plants using proximal images and machine learning: A review. *Comput. Electron. Agric.* **2019**, *162*, 482–492. [CrossRef]
7. Zúñiga, A.M.G.; Bruno, O.M. Sistema de visão Artificial para Identificação do Estado Nutricional de Plantas. Master's Thesis, Sciences of Computation and Mathematical Computation, Universidade de São Paulo, São Carlos, Brazil, 2012.
8. Condori, R.H.M.; Romualdo, L.M.; Bruno, O.M.; Luz, P.H.D.C. Comparison between Traditional Texture Methods and Deep Learning Descriptors for Detection of Nitrogen Deficiency in Maize Crops. In Proceedings of the 13th Workshop of Computer Vision, WVC 2017, Natal, Brazil, 30 October–1 November 2017; pp. 7–12. [CrossRef]
9. Smith, M.L.; Smith, L.N.; Hansen, M.F. The quiet revolution in machine vision—A state-of-the-art survey paper, including historical review, perspectives, and future directions. *Comput. Ind.* **2021**, *130*, 103472. [CrossRef]
10. Barbedo, J.G. Factors influencing the use of deep learning for plant disease recognition. *Biosyst. Eng.* **2018**, *172*, 84–91. [CrossRef]
11. Zhang, X.; Qiao, Y.; Meng, F.; Fan, C.; Zhang, M. Identification of maize leaf diseases using improved deep convolutional neural networks. *IEEE Access* **2018**, *6*, 30370–30377. [CrossRef]
12. Hughes, D.P.; Salathe, M. An open access repository of images on plant health to enable the development of mobile disease diagnostics. *arXiv* **2015**, arXiv:1511.08060. Available online: http://arxiv.org/abs/1511.08060 (accessed on 31 November 2023).
13. Bhatt, P.; Sarangi, S.; Shivhare, A.; Singh, D.; Pappula, S. Identification of Diseases in Corn Leaves Using Convolutional Neural Networks and Boosting. In Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2019), Prague, Czech Republic, 19–21 February 2019; pp. 894–899. [CrossRef]
14. Chen, J.; Chen, J.; Zhang, D.; Sun, Y.; Nanehkaran, Y.A. Using deep transfer learning for image-based plant disease identification. *Comput. Electron. Agric.* **2020**, *173*, 105393. [CrossRef]
15. Zeng, W.; Li, H.; Hu, G.; Liang, D. Identification of maize leaf diseases by using the SKPSNet-50 convolutional neural network model. *Sustain. Comput. Informatics Syst.* **2022**, *35*, 100695. [CrossRef]
16. Verma, A.; Bhowmik, B. *Automated Detection of Maize Leaf Diseases in Agricultural Cyber-Physical Systems*; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2022; pp. 841–846. [CrossRef]
17. Guerrero, R.; Renteros, B.; Castaneda, R.; Villanueva, A.; Belupu, I. Detection of nutrient deficiencies in banana plants using deep learning. In Proceedings of the 2021 IEEE International Conference on Automation/24th Congress of the Chilean Association of Automatic Control, ICA-ACCA, Online, 22–26 March 2021. [CrossRef]
18. Jahagirdar, P.; Budihal, S.V. Framework to Detect NPK Deficiency in Maize Plants Using CNN. *Adv. Intell. Syst. Comput.* **2021**, *1199*, 366–376. [CrossRef]
19. de Fátima da Silva, F.; Luz, P.H.C.; Romualdo, L.M.; Marin, M.A.; Zúñiga, A.M.G.; Herling, V.R.; Bruno, O.M. A Diagnostic Tool for Magnesium Nutrition in Maize Based on Image Analysis of Different Leaf Sections. *Crop Sci.* **2014**, *54*, 738–745. [CrossRef]
20. Peng, Q.; Shen, R.; Li, X.; Ye, T.; Dong, J.; Fu, Y.; Yuan, W. A twenty-year dataset of high-resolution maize distribution in China. *Sci. Data* **2023**, *10*, 1–18. [CrossRef]

21. Smaranjit, G. Corn or Maize Leaf Disease Dataset. 2020. Available online: https://www.kaggle.com/datasets/smaranjitghose/corn-or-maize-leaf-disease-dataset/data (accessed on 31 November 2023).
22. Vasilev, I.; Slater, D.; Spacagna, G.; Roelants, P.; Zocca, V. *Python Deep Learning: Exploring Deep Learning Techniques and Neural Network Architectures with PyTorch, Keras, and TensorFlow*, 2nd ed.; Packt Publishing: Birmingham, UK, 2019; Volume 1, p. 379.
23. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* **2014**, *115*, 211–252. [CrossRef]
24. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. In Proceedings of the 3rd International Conference on Learning Representations, ICLR, San Diego, CA, USA, 7–9 May 2015; Conference Track Proceedings, 2014.
25. 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; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2010; pp. 248–255. [CrossRef]
26. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 770–778. [CrossRef]
27. Andrew, J.; Eunice, J.; Popescu, D.E.; Chowdary, M.K.; Hemanth, J. Deep Learning-Based Leaf Disease Detection in Crops Using Images for Agricultural Applications. *Agronomy* **2022**, *12*, 2395. [CrossRef]
28. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going Deeper with Convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
29. Huang, G.; Liu, Z.; Maaten, L.V.D.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269. [CrossRef]
30. Pradhan, P.; Kumar, B.; Mohan, S. Comparison of various deep convolutional neural network models to discriminate apple leaf diseases using transfer learning. *J. Plant Dis. Prot.* **2022**, *129*, 1461–1473. [CrossRef]
31. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861. Available online: https://arxiv.org/abs/1704.04861v1 (accessed on 31 November 2023).
32. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [CrossRef]
33. Hassan, S.M.; Maji, A.K.; Jasiński, M.; Leonowicz, Z.; Jasińska, E. Identification of Plant-Leaf Diseases Using CNN and Transfer-Learning Approach. *Electronics* **2021**, *10*, 1388. [CrossRef]
34. The MathWorks Inc. Deep Learning Toolbox. 2020. Available online: https://la.mathworks.com/products/deep-learning.html (accessed on 14 August 2023).
35. Mohanty, S.P.; Hughes, D.P.; Salathé, M. Using deep learning for image-based plant disease detection. *Front. Plant Sci.* **2016**, *7*, 1419. [CrossRef]
36. Barbedo, J.G.A. Impact of dataset size and variety on the effectiveness of deep learning and transfer learning for plant disease classification. *Comput. Electron. Agric.* **2018**, *153*, 46–53. [CrossRef]
37. 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.; Méndez, C.A.G.; Olvera-Olvera, C.A. Comparison of Convolutional Neural Network Architectures for Classification of Tomato Plant Diseases. *Appl. Sci.* **2020**, *10*, 1245. [CrossRef]
38. Nagaoka, T. Hyperparameter Optimization for Deep Learning-based Automatic Melanoma Diagnosis System. *Adv. Biomed. Eng.* **2020**, *9*, 225–232. [CrossRef]

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.