diversity

MDPI

Article

PTL-Inception: Integrating Deep Learning and Taxonomy for Desert Plant Classification

Yonis Gulzar [1,*], Zeynep Ünal [2,*], Kadir Şahbaz [2] and Mohannad Alkanan [1]

[1] Department of Management Information Systems, College of Business Administration, King Faisal University, Al-Ahsa 31982, Saudi Arabia
[2] Department of Biosystem Engineering, Niğde Ömer Halisdemir University, Central Campus, Niğde 51240, Türkiye
* Correspondence: ygulzar@kfu.edu.sa (Y.G.); zeynepunal@ohu.edu.tr (Z.Ü.)

Abstract

Accurate identification of plant species forms the basis of taxonomy, biodiversity assessment, and conservation planning. This requirement is especially urgent in arid ecosystems such as those of Saudi Arabia, where limited rainfall, fragile habitats, and high ecological stress create distinctive but poorly documented flora. In this study, a deep learning framework, termed PTL-Inception, was developed to classify desert plants and to provide reliable taxonomic data that can be integrated into biodiversity and phylogenetic studies. A dataset of ten native species was compiled and expanded through augmentation, and several state-of-the-art architectures were tested. InceptionV3 was found to be the most effective baseline, and the network was further modified by incorporating ten additional layers, transfer learning, and hyperparameter tuning. The proposed model achieved an accuracy of 99.46%, with precision and recall values of 99.46% and 99.44%, respectively. Reliability was confirmed through K-fold validation, while early stopping reduced training time with minimal loss of accuracy. Beyond these computational outcomes, the study demonstrates how deep learning can complement traditional taxonomy by producing consistent species-level identifications. The outputs can be combined with spatial and phylogenetic approaches to explore patterns of diversity, endemism, and adaptation in desert ecosystems, thereby supporting conservation strategies and biodiversity management.

Keywords: desert plants; taxonomy; biodiversity assessment; spatial phylogenetics; deep learning; endemism; conservation; image classification

check for updates

Academic Editors: Fazal Ullah, Hong-Hu Meng, Wajid Zaman and Saddam Saqib

Received: 8 September 2025
Revised: 25 October 2025
Accepted: 19 November 2025
Published: 20 November 2025

Citation: Gulzar, Y.; Ünal, Z.; Şahbaz, K.; Alkanan, M. PTL-Inception: Integrating Deep Learning and Taxonomy for Desert Plant Classification. *Diversity* **2025**, *17*, 806. https://doi.org/10.3390/d17110806

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Plants are widely regarded as the foundation of ecosystems, ranging from tall trees to small wildflowers. Their importance is not limited to ecological balance but extends to human well-being through food provision, landscape formation, and medicinal properties. More than 400,000 species have been documented globally, and they have long been classified to enable systematic study and organization [1,2]. In arid regions such as Saudi Arabia, the demand for accurate classification is particularly pressing. Limited rainfall, extreme heat, and fragile ecosystems make it essential that species be identified correctly so that their survival strategies and ecological adaptations can be understood.

Several practical outcomes depend on reliable classification. Biodiversity assessment cannot be undertaken without precise identification of species, as this forms the basis for evaluating the richness and condition of ecosystems [3]. Conservation programs likewise

depend on correct recognition, since rare or endemic plants can only be protected when they are clearly distinguished. Ecosystem services also rely on this foundation: desert plants stabilize soils, provide habitats for other organisms, and offer resources for human use. The sustainable management of these services requires accurate classification [4,5]. Monitoring climate change adds another dimension, as shifts in temperature and rainfall alter plant distributions, providing evidence of ecological response and adaptation [6].

Agriculture has also benefited from such work. Desert plants that tolerate drought have been identified, and their characteristics have been used to guide the development of crops suited to water-scarce environments [7]. Many desert species are further known for medicinal compounds used by local populations. Through systematic classification, this traditional knowledge can be preserved and scientifically validated [8].

In Saudi Arabia, classification is therefore not merely a record-keeping exercise but contributes directly to biodiversity conservation, agricultural sustainability, cultural knowledge, and climate research. Internationally, plant classification has been studied using a variety of datasets and methods. Computational approaches have increasingly taken a central role, with deep learning models applied to leaves, seeds, and other plant parts with considerable success. However, most investigations have relied on large global datasets collected under controlled conditions, while research focused on Saudi Arabian flora remains scarce. This limitation is significant, as the country's arid climate and sensitive ecosystems present distinctive challenges that require region-specific solutions.

Beyond its computational focus, this study bridges artificial intelligence and taxonomy by demonstrating how image-based deep learning can support plant identification, species documentation, and biodiversity monitoring in arid ecosystems. To address the lack of region-specific data, a new image dataset of ten desert plant families from Saudi Arabia was constructed under natural conditions to capture authentic variability. Building on comparative evaluation of state-of-the-art models, a new architecture—PTL-Inception (Plant Taxonomy Learning)—was designed by extending InceptionV3 with ten additional layers, transfer learning, and hyperparameter tuning to enhance feature extraction and classification accuracy.

The key contributions of this study are as follows:

- A region-specific image dataset was developed, comprising samples from ten desert plant families native to Saudi Arabia. The images were captured under natural environmental conditions to ensure realistic representation of morphological variability and background complexity.
- A customized deep learning architecture, PTL-Inception, was constructed by extending the InceptionV3 network with additional layers and optimized transfer learning strategies. This design enhanced the model's feature discrimination capability for desert plant classification, and its reliability and generalization were validated through ablation studies, K-fold cross-validation, and benchmarking against established deep learning architectures.
- The classification outputs generated by PTL-Inception were integrated into taxonomic identification workflows, underscoring the model's potential contribution to biodiversity documentation and ecological monitoring within desert ecosystems.

The remainder of this paper is organized as follows: Section 2 reviews the related work. Section 3 describes the materials and methods. Section 4 presents and discusses the results. Section 4.6 outlines the limitations of the study, and Section 5 concludes the paper.

2. Related Work

The use of artificial intelligence has been extended to several domains, where image classification has become a core application. In agriculture, it has been applied to crop

monitoring, disease detection, and yield prediction [9]. In healthcare, medical image analysis has benefited from these techniques, improving diagnostic accuracy and treatment planning. Other sectors, such as finance and education, have also reported significant advantages, with improvements in fraud detection, risk management, and personalized learning. These developments show that computational approaches, particularly deep learning, have assumed a central role in tasks requiring image-based recognition.

Within agriculture, deep learning methods have been widely studied for different purposes. Enhanced segmentation networks have been used for rice disease detection, incorporating dilated convolution and EfficientNetB4 to improve feature extraction, with modified UNet models reporting significant accuracy gains [10,11]. Transfer learning architectures such as InceptionV3, ResNet, MobileNet, and DenseNet have also been applied, with ensemble strategies showing higher robustness in disease diagnosis [12]. In post-harvest studies, deep learning models combined with Near-Infrared (NIR) imaging have been employed for fruit quality assessment, achieving around 99% accuracy when spectral data were included. Similarly, modified Inception networks have been tested for soybean seed classification, where additional layers improved recognition accuracy [13].

In the field of plant classification, deep learning has facilitated the identification of species for biodiversity assessment, environmental monitoring, and agricultural practices [14]. Several approaches have targeted medicinal plants [15], foliage-based species recognition [16], and public datasets such as PlantCLEF [17]. Segmentation methods have been integrated with CNNs to enhance feature learning from leaf images [18], while hybrid techniques, including CNN combined with GANs or Mask R-CNN with VGG16, have improved classification under complex conditions [19,20]. Automated dataset generation systems have been developed to address data scarcity, enabling large-scale training with diverse perspectives [21].

Recent advances have introduced alternative strategies. CNNs integrated with entropy impurity have been used for aromatic and medicinal plants [22], while hybrid CNN–SVM and CNN–kNN models have reported high accuracy across benchmark datasets [23]. Multi-task learning frameworks have been proposed to jointly predict plant species and diseases, reducing redundancy in training [24]. Deeper architectures such as DenseNet [25] and optimization-driven approaches like PB3C [26] have also been evaluated. More recently, ensemble models that combine CNNs with Vision Transformers (ViTs) have demonstrated superior performance on standard datasets, achieving accuracy rates close to 99% [27]. In the area of plant disease identification, EfficientNet variants have been adopted, with mobile-friendly systems designed for real-time use [28]. Ensemble-based approaches, such as MIV-PlantNet, have further improved classification, attaining accuracies above 99% [29].

Although considerable progress has been made, important limitations remain. Most studies rely on global datasets collected under controlled conditions, which may not reflect the variability of natural habitats. In addition, the majority of models have been validated on species from temperate regions, while little attention has been directed to biodiversity in arid and desert ecosystems. For Saudi Arabia in particular, studies are scarce, despite the ecological importance and the challenges posed by extreme climatic conditions and complex desert environments. These gaps underline the need for region-specific datasets and adapted deep learning models capable of reliable performance in such settings.

3. Materials and Methods

In the process of developing a system for plant recognition, the task has been performed in three stages, as shown in Figure 1. In the first stage, the original and augmented datasets were obtained. To construct the original dataset, the collected images were resized

and arranged in folders, allocating one folder for each class. Later, the augmented dataset was obtained by using image augmentation techniques and applied to the original dataset without modifying it. In the second we chose state-of-the-art models renowned for their exemplary performance in image classification, as documented in existing literature. These models include VGG19 [30], ResNet50 [31], EfficientNetB3 [32], Xception [33], and InceptionV3 [34] models were trained for 100 epochs using the same selected parameters. Using performance metrics, the models were evaluated, and the best model was selected. In the last stage, the best model was modified by adding ten layers. The obtained model was trained from scratch using ImageNet weights. To obtain better accuracy, hyperparameter tuning was performed. After obtaining targeted performance metrics, the early stop parameter was activated to find the optimum number of epochs needed for training, which aids in the efficient utilization of resources.

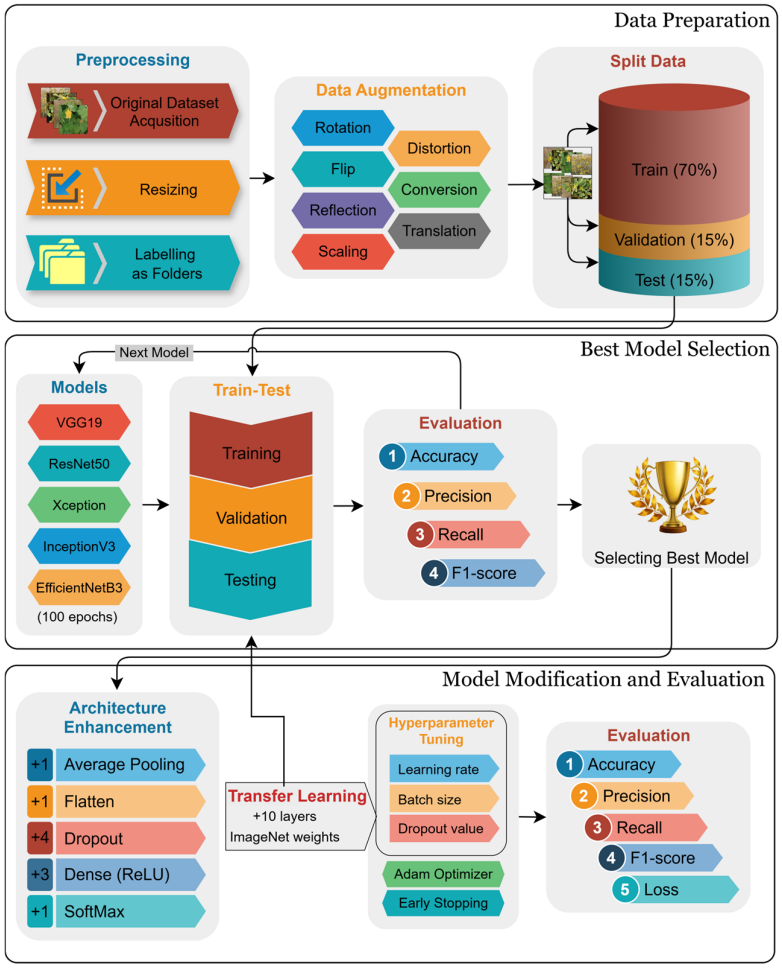


Figure 1. Flowchart of the proposed approach.

3.1. Data Description

The dataset prepared for this study is an important resource for plant recognition, consisting of RGB images from ten different plant families native to Saudi Arabia [29]. A total of more than 1050 images have been compiled, offering sufficient material for both training and evaluation. The photographs were taken with a Nikon D5100 DSLR camera (16.2 MP) (Tokyo, Japan) using an 18–55 mm VR lens, which ensured high image clarity. Each file carries a resolution of 3679 × 5439 pixels, allowing fine details of plant structures to be examined.

A key strength of the dataset lies in the variety of its natural settings. The images were collected from desert regions of Saudi Arabia, where the background often included

shifting sand, scattered rocks, and sparse vegetation, all under changing light conditions. Such elements create significant variation and introduce visual complexity, making it harder for models to isolate plant features in challenging environments. This difficulty, however, is valuable, as it provides a strong test of the reliability and adaptability of recognition methods.

Figure 2 presents a selection of sample images from the ten families, highlighting both the diversity of plant features and the demanding nature of their natural surroundings.

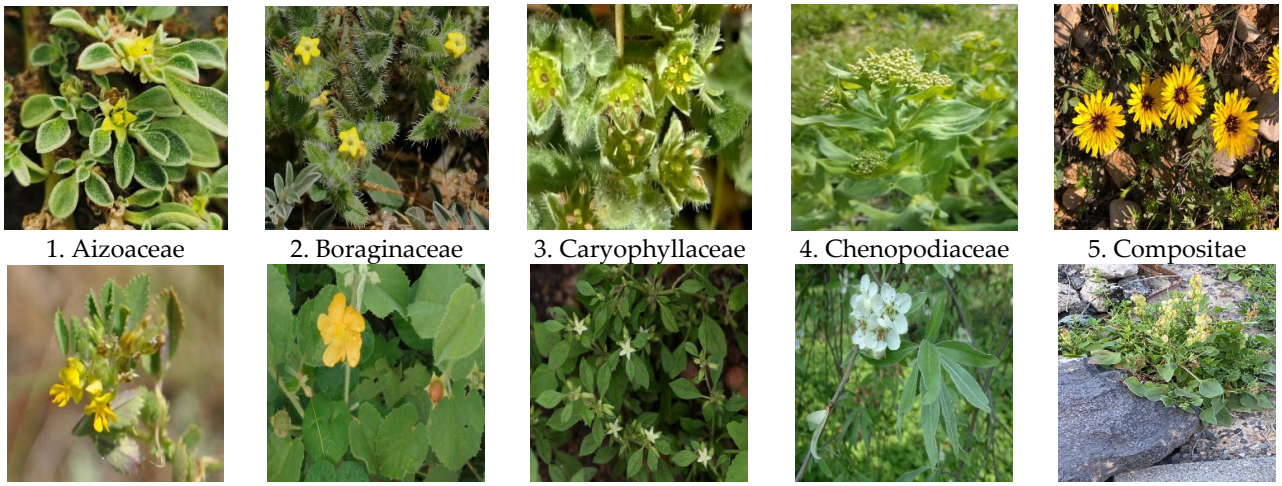


Figure 2. Samples from the dataset.

Detailed insights regarding the dataset, encompassing class count, original image count per class, and the distribution of images for training, validation, and testing, are presented in Table 1. To avoid bias, we carefully partitioned the dataset into these subsets, ensuring that no duplicate or overlapping images were present across them. This strict separation guaranteed that the model was evaluated only on unseen samples, thereby preventing data leakage and ensuring a fair and reliable assessment of performance.

Table 1. Details of the original dataset.

Class	Training	Validation	Test	Images Before Augmentation
Aizoaceae	73	16	16	105
Boraginaceae	70	15	16	101
Caryophyllaceae	73	16	16	105
Chenopodiaceae	71	15	16	102
Compositae	84	18	18	120
Leguminosae	70	15	15	100
Malvaceae	70	15	15	100
Molluginaceae	77	16	17	110
Moraceae	74	16	17	107
Polygonaceae	70	15	15	100
Total	732	157	161	1050

Desert ecosystems impose strong habitat filtering, which often leads to convergent adaptive traits such as reduced leaf size, thick cuticles, or succulence. While these similarities can make automated recognition more challenging, our dataset included finer taxonomic distinctions such as variation in reproductive structures, branching forms, and leaf morphology. These features provided sufficient variability for discrimination, while augmentation under diverse natural backgrounds and lighting conditions further reduced the risk of the model relying solely on shared desert-adaptive traits.

Although the present study reported classification at the family level, the dataset was constructed from multiple representative species within each family. This design ensured that intra-family morphological diversity was captured, rather than restricting the dataset to single exemplars. The family-level grouping was deliberately adopted to establish a broad and challenging baseline for evaluating the robustness of PTL-Inception in the context of Saudi Arabia's desert flora.

3.2. Data Augmentation

When dealing with CNN models, the quantity of images plays an important role in the training success of the architectures. The insufficient quantity of images in the training dataset exposes the model to learning patterns with very high variation, resulting in poor predictions on test data. This case is recognized in the literature as overfitting [35]. There are functional techniques in the literature to solve the overfitting problem, such as batch normalization, dropout regularization, and early stopping and transfer learning, which have been developed and applied to training on small datasets. Data Augmentation is not just reducing overfitting; it also addresses the core cause of poor generalization of the problem by artificially boosting the training dataset size through data warping or oversampling [35]. The image augmentation algorithms are mainly classified into three main categories such as model-free, model-based, and policy-based algorithms. In our study, we used a model-free approach because it does not need any pre-trained model to do image augmentation and can be used even for a single image. Geometrical transformation applied to a single image is one of the model-free approaches, which include operations such as rotation, flip, conversion, translation, reflection, elastic distortion, and scaling. Intensity transformation is also a model-free approach that includes techniques such as blurring, cutout, random erasing, and grid mask. In our study, we applied model-free geometrical transformation using built-in preprocessing functions in the Keras Library. For creating new instances of images, each original sample was rotated by an angle of 45, shifted by 20% horizontally and vertically, with a zoom range of 20%. The gaps formed after these geometrical transformations were filled with neighbor pixel values using reflection parameters, as shown in Figure 3. Later, the augmented dataset was split into training, validation, and testing datasets using a ratio of 70/15/15 as shown in Table 2.

A similar precaution was applied during augmentation. Augmented samples were generated only within the designated split of each class, and we verified that no identical or duplicate images were shared between training, validation, and testing sets. This design ensured that performance metrics reported for the test set reflected generalization to genuinely unseen data.

3.3. PTL-Inception Model

Deep learning, which is built on the principles of artificial neural networks, has been increasingly applied in agricultural research. The rapid growth of digital agricultural data has provided a suitable foundation for the development and evaluation of advanced learning architectures. Convolutional neural networks (CNNs), which process images through hierarchical feature extraction, have become the most widely adopted models in this domain. Their success can be traced to benchmark competitions such as the ImageNet Large-Scale Visual Recognition Challenge (ILSVRC), where architectures including Inception achieved state-of-the-art performance. InceptionV3, in particular, introduced inception blocks that combine multiple convolutional filter sizes within parallel branches, allowing the simultaneous extraction of local and global features at reduced computational cost.

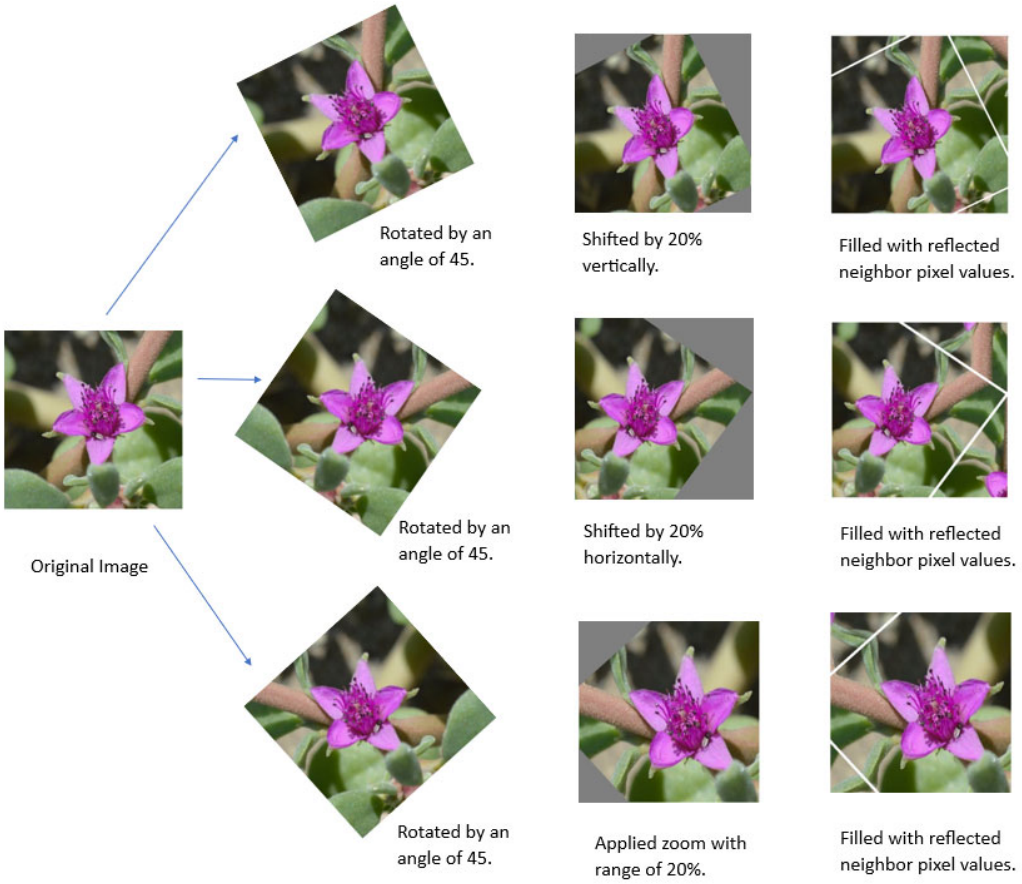


Figure 3. Examples of transformation applied to images.

Table 2. Details of the augmented dataset.

Class	Training	Validation	Test	# Images After Augmentation
Aizoaceae	779	167	168	1114
Boraginaceae	751	161	162	1074
Caryophyllaceae	779	167	168	1114
Chenopodiaceae	758	163	163	1084
Compositae	884	190	190	1264
Leguminosae	744	160	160	1064
Malvaceae	744	160	160	1064
Molluginaceae	814	175	175	1164
Moraceae	793	170	171	1134
Polygonaceae	744	160	160	1064
Total	7790	1673	1677	11,140

In this study, InceptionV3 was empirically selected as the baseline model and subsequently modified to produce a new architecture, referred to as PTL-Inception. The original InceptionV3 network contains 1000 output nodes designed for large-scale classification. For the present dataset, the final dense layer was removed and replaced with a customized classification head containing ten nodes corresponding to the plant families under study. To improve learning capacity and generalization, ten additional layers were introduced, consisting of one average pooling layer, one flattening layer, four dropout layers $p = 0.3$, three fully connected layers activated by ReLU, and a final SoftMax layer. The overall architecture of PTL-Inception is presented in Figure 4, where the integration of the new layers on top of InceptionV3 is illustrated.

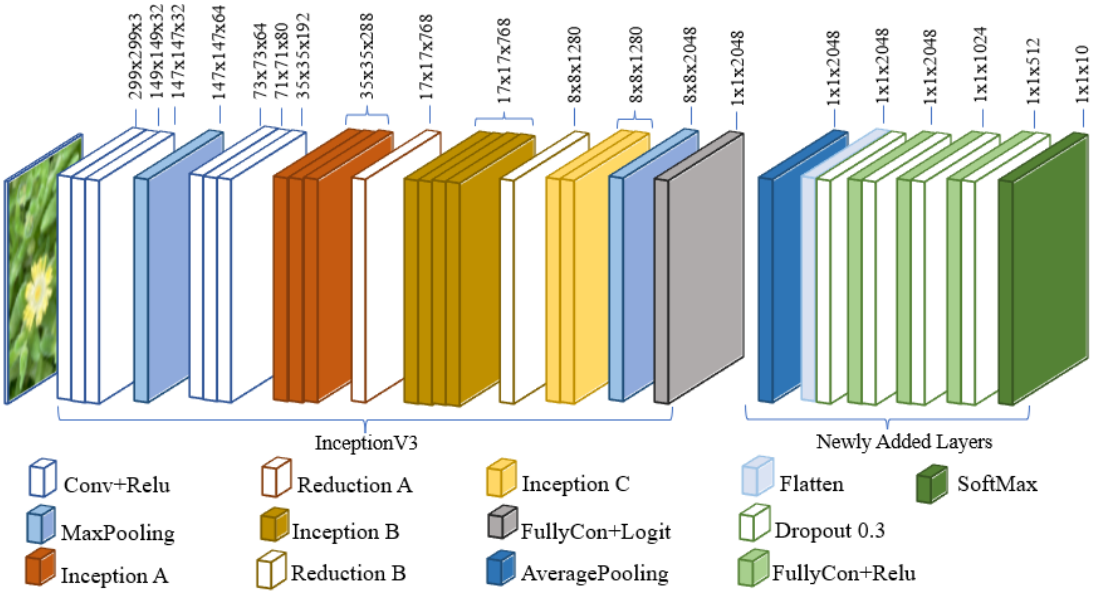


Figure 4. The architecture of the PTL-Inception model.

For clarity and completeness, the key computational steps of PTL-Inception are presented in mathematical form. Equations (1)–(8) describe the convolutional operation, inception block, dropout regularization, fully connected layers, SoftMax output, loss function, and optimization process.

Convolutional Operation

Each convolutional layer applies a kernel W to an input feature map X as

$$h_{i,j}^{(l)} = \sigma\left(\sum_{m=1}^{M}\sum_{n=1}^{N} W_{m,n}^{(l)} \cdot X_{i+m,j+n}^{(l-1)} + b^{(l)}\right), \tag{1}$$

where $h_{i,j}^{(l)}$ denotes the activation at position (i,j) in layer l, $W^{(l)}$ is the convolutional kernel of size $M \times N$, $b^{(l)}$ is the bias, and $\sigma(\cdot)$ is the activation function (ReLU).

Inception Block

An inception block concatenates feature maps obtained from parallel convolutions with different receptive fields:

$$F^{(l)} = \text{Concat}\left(f_{1\times1}\left(X^{(l-1)}\right), f_{3\times3}\left(X^{(l-1)}\right), f_{5\times5}\left(X^{(l-1)}\right), f_{\text{pool}}\left(X^{(l-1)}\right)\right). \tag{2}$$

This mechanism enables the network to learn both fine-grained and global patterns simultaneously.

Dropout Regularization

Dropout layers were introduced to reduce overfitting by randomly deactivating neurons during training:

$$\widetilde{h}^{(l)} = h^{(l)} \cdot r, r \sim \text{ Bernoulli } (1-p), \tag{3}$$

where $p = 0.3$ denotes the dropout probability and $\widetilde{h}^{(l)}$ represents the resulting activation.

Fully Connected Layers

The fully connected layers combine extracted features into class-discriminative representations:

$$z^{(l)} = W^{(l)}h^{(l-1)} + b^{(l)}, h^{(l)} = \text{ReLU}(z^{(l)}). \tag{4}$$

SoftMax Output Layer

For the ten-class classification problem, the SoftMax function is defined as:

$$P(y = c \mid z) = \frac{\exp(z_c)}{\sum_{j=1}^{10} \exp(z_j)}, c = 1, 2, \ldots, 10 \quad (5)$$

where $P(y = c)$ represents the probability of assigning an input to class c. The predicted label is obtained as:

$$\hat{y} = \arg\max_c P(y = c \mid z). \quad (6)$$

Loss Function

The network was optimized using categorical cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{10} y_{i,c} \log P(y_i = c), \quad (7)$$

where N is the batch size and $y_{i,c}$ denotes the one-hot encoded ground truth.

Optimization

The parameters were updated with the Adam optimizer, which combines momentum and adaptive learning rates:

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon}, \quad (8)$$

where $\hat{m}_t$ and $\hat{v}_t$ are bias-corrected first and second moment estimates, η is the learning rate, and ϵ is a small constant for numerical stability.

3.4. Transfer Learning

CNN structures can be trained from scratch or using predefined weights. The knowledge gained by the network trained for image classification speeds up the more specific training process when used as a predefined weight. This technique is called transfer learning [36]. In cases where the current labeled training data is scarce, it accelerates the training process by making fewer calculations since it starts the training with feasible weights. The first layers of CNN architectures determine the basic elements, such as edges, corners, or brightness, and the next layers capture task-specific details. The pre-trained architecture will be familiar with the basic elements, making it easier to focus on the task-specific details.

3.5. Hyperparameter Tuning

Given today's software and hardware platforms, it appears that implementing and using CNNs is not difficult. However, to achieve the best classification results, hyperparameters for each task must be determined properly. Even when using proven architectures with a defined number of activation functions, number of neurons, and layers, many combinations of hyperparameters such as learning rate, number of epochs, and batch size should be tried. There are many techniques for hyperparameter selection in the literature, which generally require a lot of computing power. Therefore, in practice, it is typical that the research is based on experience. The researcher, who has done similar studies, can reach the result faster by starting to try the most possible values of the hyperparameters that will lead to success in the shortest time. Modification performed by adding layers to the SOTA model also needs to be applied with careful selection of the dropout value to avoid overfitting. The other approach to overcome the underfitting and overfitting paradigm is to use an early stopping parameter. The selection of optimization functions also affects training speed as well as accuracy. The speed at which the optimization function can reach

the global minima without being stuck in local minima plays a decisive role in terms of the suitability of the optimization function.

In this research, we compared the results of the Adam optimizer to other available optimizers in the Keras library for the obtained dataset. Steps applied for hyperparameter tuning are given in Figure 5.

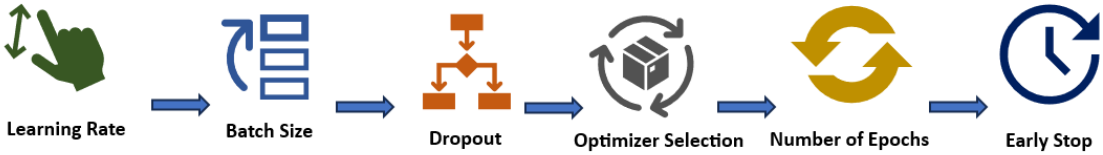


Figure 5. Steps applied for hyperparameter tuning.

A systematic approach was employed for hyperparameter tuning in this study. Initially, all hyperparameters were kept constant, and a series of trials were conducted with different learning rates, ranging from 0.1 to subsequent decreases. The trials continued until the optimal learning rate of 0.00001 was identified. Once the optimal learning rate was determined, it was fixed, and various batch sizes (8, 16, 32, 64) were explored for further training, ultimately identifying 16 as the optimal batch size, as mentioned in Table 3. Subsequently, experimentation with different dropout values revealed the optimal dropout rate to be 0.3. Various optimizers available in the Keras library, including RMSprop, Adam, Adadelta, Adagrad, Adamax, Adafactor, and Nadam, were employed to assess their impact on accuracy. The number of training epochs was set to 100, but early stopping was implemented. After each training iteration, the results were comprehensively evaluated using model evaluation metrics.

Table 3. Identifying Optimal Learning Rate.

Learning Rate	Batch Size	Optimizer	Accuracy	Loss	Duration
0.1	16	Adam	0.9277	0.3028	5:41:24
0.01	16	Adam	0.9546	0.1921	5:42:12
0.001	16	Adam	0.9731	0.1367	5:39:43
0.0001	16	Adam	0.9899	0.0649	5:41:40
0.00001	**16**	**Adam**	**0.9976**	**0.0036**	**5:41:24**
0.000001	16	Adam	0.9904	0.0367	5:42:11
0.00001	8	Adam	0.9499	0.2954	7:32:05
0.00001	**16**	**Adam**	**0.9976**	**0.0036**	**5:41:24**
0.00001	32	Adam	0.9844	0.0624	4:25:43
0.00001	64	Adam	0.9821	0.0797	3:57:32
0.00001	16	RMSprop	0.9384	0.2379	**5:38:54**
0.00001	16	Adadelta	0.9480	0.1667	**5:39:23**
0.00001	16	Adagrad	0.9767	0.0952	**5:40:12**
0.00001	16	Adamax	0.9791	0.1124	**5:38:47**
0.00001	16	Adafactor	0.9868	0.0423	**5:39:38**
0.00001	**16**	**Adam**	**0.9976**	**0.0036**	**5:41:24**
0.00001	**16**	**Nadam**	**0.9976**	**0.0036**	**5:42:59**

3.6. Model Evaluation and System Configuration

Model Evaluation metrics such as accuracy, precision, recall, and F1-score were employed to evaluate the performance of classifying plant images. The metrics were calculated using the Equations given in Table 4.

Table 4. Model evaluation metrics.

Accuracy	$\frac{(TP+TN)}{(TP+FP+TN+FN)}$	The ratio of correctly predicted samples to the total number of samples.
RecallorSensitivity	$\frac{TP}{(TP+FN)}$	The proportion of correctly predicted positive samples out of all actual positives.
Precision	$\frac{TP}{(TP+FP)}$	The proportion of correctly predicted positives out of all predicted positives.
$F1-score$	$2\frac{Precision*Recall}{Precision+Recall}$	The harmonic mean of precision and recall balanced both metrics.
Cohen's Kappa coefficient (κ)	$\frac{P_o-P_e}{1-P_e}$	Measures the agreement between predicted and true class labels while accounting for the agreement occurring by chance. Higher values indicate stronger reliability in multi-class classification.

The other metric used in model evaluation is Loss Function. The Loss Function is a metric used for adjusting the weights of the network to fit the data. Although loss is connected to model accuracy, it also indicates how much we may rely on accuracy value. During training each pass-through batch of data produces outputs showing class probability, which are compared with the target classes, and a penalty is calculated for the deviation between actual and predicted classes. Considering the penalty, the weights are updated before the next batch or epoch. Each weight update aims to minimize loss. As the dataset contains more than 2 classes, categorical cross-entropy loss was used.

Models were tested on Windows 10 Pro OS (Redmond, WA, USA) installed on a computer with an Intel i5 processor running at 2.9 GHz, 16 GB RAM (Santa Clara, CA, USA), and Nvidia GeForce GTX 1660 Graphical Processing Unit (Santa Clara, CA, USA). The proposed model implementation was performed in Python (v. 3.8) environment using OpenCV (v. 4.7) and Keras Library (v. 2.8).

4. Results and Discussion

This section delves into the results and discussion of our study. Initially, we determine the optimal model among the state-of-the-art (SOTA) models by training and validating them on both the original and augmented datasets. Employing various techniques, including adjustments to batch sizes, learning rates, epoch numbers, and optimizer types, we assess the SOTA models' performance under diverse conditions to identify the optimal model (InceptionV3) for our plant classification problem. After identifying the optimal model, a modified version of InceptionV3, referred to as PTL-Inception, was introduced and tested using different evaluation metrics. An ablation study was carried out to examine the contribution of individual components, and the performance was further validated through K-fold cross-validation.

4.1. Evaluation of Model Selection Process

In this subsection, we elucidate the outcomes derived from deploying SOTA models on the said dataset. Several state-of-the-art models were selected based on their strong reputation and proven effectiveness reported across different domains in the literature. The overarching goal of this process is to discern the optimal model that aligns most effectively with our dataset.

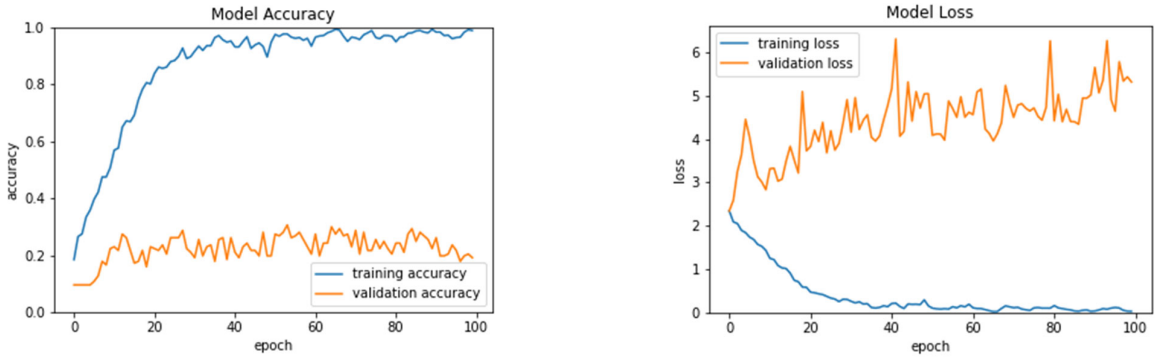
To ensure a fair and comprehensive comparison, all selected models, including VGG19, ResNet50, EfficientNetB3, Xception, and InceptionV3, underwent training on the original dataset. This training spanned 100 epochs, and no architectural modifications or methods to enhance accuracy were employed. The standardized training approach provides a robust foundation for evaluating the intrinsic capabilities of each model under consistent conditions. Table 5 presents the training, validation and test results of SOTA models based on our plant dataset.

Table 5. Training, Validation, and Test Results of SOTA Models.

Dataset	Models	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Test Accuracy	Test Loss
Original Dataset	VGG19	1.0000	0.0035	0.1274	9.8873	0.1304	8.0128
	ResNet50	0.9317	0.2353	0.1592	6.9275	0.1863	6.2616
	EffiecientNetB3	0.8907	0.3570	0.1975	4.9838	0.1987	4.8269
	Xception	0.8456	0.4713	0.1274	5.8887	0.1242	6.0942
	InceptionV3	0.9877	0.0337	0.1911	5.3141	0.2671	5.2396

Table 5 reveals notable patterns in the performance of the models. While all models demonstrated high accuracy and low loss values during the training process, there is a noticeable decline in performance during validation, indicating the presence of an overfitting issue. However, it is noteworthy that InceptionV3 outperformed the other models in terms of validation performance, as evidenced by the results presented in the table.

Additionally, we provide the training curves for the InceptionV3 model (Figure 6) to offer a transparent understanding of its performance throughout each epoch in both training and validation phases.



(**a**) (**b**)

Figure 6. Training curves InceptionV3 trained on the original dataset: (**a**) Accuracy, (**b**) Loss.

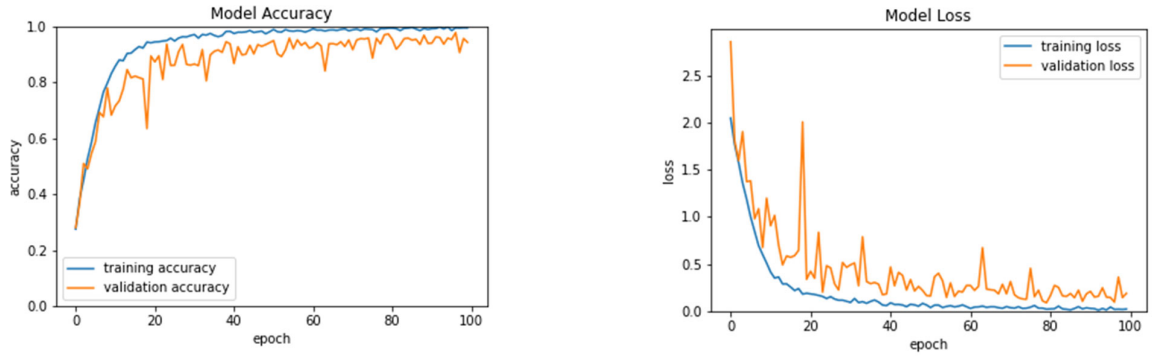
Figure 6a illustrates a gradual increase in accuracy during model training with each epoch. Conversely, in the case of validation accuracy, a consistent downward trajectory is observed with each epoch. This trend is further evident when examining the training and validation loss values depicted in Figure 6b, where low loss values are observed during training, contrasting with an increase during the validation process. These patterns suggest a potential overfitting challenge, emphasizing the need for strategies to mitigate this issue.

In pursuit of enhancing the performance of state-of-the-art (SOTA) models, various trials were conducted, manipulating batch sizes, learning rates, the number of epochs, and optimizer types. The discerned optimal learning rate was 0.0001, coupled with an optimal batch size of 16. The choice of the Adam optimizer proved effective, as other optimizers exhibited minimal impact on the learning process, even across 100 epochs. Regrettably, none of the hyperparameters yielded a significant improvement, except for image augmentation.

Utilizing the configurations outlined in Table 2, we proceeded to train the SOTA models. The outcomes of models trained on the augmented dataset are detailed in Table 6. Notably, the test accuracy of all models demonstrated relative improvement when trained on the augmented dataset compared to the original dataset. Among them, the InceptionV3 model showcased the highest accuracy during this augmented training process. Figure 7 visually depicts the training curves of the InceptionV3 model trained on the augmented dataset, offering insights into its performance dynamics.

Table 6. Training, Validation, and Test Results of SOTA Models on Augmented Dataset.

Dataset	Models	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Test Accuracy	Test Loss
Augmented Dataset	VGG19	0.9724	0.0992	0.5282	6.4004	0.5375	5.6017
	ResNet50	0.9736	0.0767	0.6031	1.8564	0.6195	1.8080
	EffiecientNetB3	0.9140	0.2451	0.6396	1.4203	0.6100	1.5228
	Xception	0.9752	0.0447	0.8770	0.9562	0.8615	1.0678
	InceptionV3	0.9937	0.0241	0.9420	0.1905	0.9481	0.1679



(**a**) (**b**)

Figure 7. Training curves InceptionV3 trained on the augmented dataset: (**a**) Accuracy, (**b**) Loss.

Figures 6 and 7 provide a comparative analysis, showcasing the evolution in learning performance as measured by accuracy and loss parameters. While improvements are evident, it is crucial to note the pick point on the validation loss curve in Figure 7b, indicating a degree of instability in the model's performance. Even after 100 training epochs, the learning curve remained uneven, indicating the need for further refinement of the models. To allow fair comparison, all models were extensively tested for precision, recall, F1-score, accuracy, loss, and training duration. The findings, summarized in Table 7, present the results for both the original and the augmented datasets. From these results, it is clear that InceptionV3 consistently achieved the best outcomes across both datasets. It showed higher precision, recall, F1-score, accuracy, Cohen's Kappa (κ), and lower loss compared with other models. Based on this comprehensive evaluation, InceptionV3 was chosen as the base model for the study.

Table 7. Overall test performance of compared models.

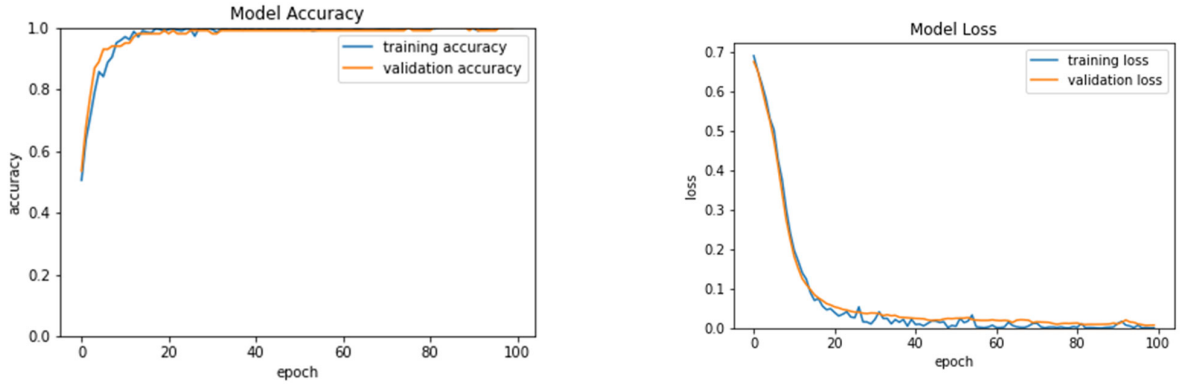
Dataset	Models	Precision	Recall	F1-Score	Accuracy	Loss	Cohen's Kappa (κ)	Duration
Original Dataset	VGG19	0.1212	0.1290	0.1225	0.1304	18.0128	0.0340	0:28:50
	ResNet50	0.1863	0.1863	0.1863	0.1863	6.2616	0.0943	0:22:19
	EffiecientNetB3	0.1836	0.1979	0.1874	0.1987	4.8269	0.1081	1:11:10
	Xception	0.1437	0.1226	0.1242	0.1242	6.0942	0.0246	0:30:12
	InceptionV3	0.2635	0.3596	0.2739	0.2671	5.2396	0.1913	0:30:23
Augmented Dataset	VGG19	0.5379	0.5394	0.5321	0.5375	5.6178	0.5012	4:51:30
	ResNet50	0.6400	0.6213	0.6037	0.6195	1.8080	0.5773	4:10:13
	EffiecientNetB3	0.6339	0.6107	0.6114	0.6100	1.5228	0.5666	9:03:10
	Xception	0.8615	0.8615	0.8615	0.8615	1.0678	0.8523	8:03:46
	InceptionV3	0.9504	0.9487	0.9482	0.9481	0.1679	0.9424	5:08:32

4.2. Evaluation of PTL-Inception Model

Following the model selection process, where InceptionV3 was identified as the most suitable baseline, the architecture was further extended to develop PTL-Inception. Ten additional layers were incorporated, as described in Section 3.3, and several refinement

techniques were applied, including data augmentation, transfer learning, and hyperparameter adjustment, to enhance classification performance.

Figure 8 presents the training and validation behavior of PTL-Inception over 100 epochs. In Figure 8A, the training and validation accuracies are shown. The model began with a training accuracy of 48%, indicating limited alignment between initial predictions and the true labels. Accuracy increased rapidly during the early iterations, reaching its maximum within 32 epochs, which demonstrates quick adaptation to the dataset. Validation accuracy, starting from 51%, followed a similar trend and achieved 99.46% by the 31st epoch. After approximately the 38th epoch, both curves stabilized and remained relatively steady until the 100th epoch. This behavior indicates that the model had reached convergence, with consistent predictive performance on both training and validation sets. The stabilization also suggests the possibility of overfitting if training were to continue for substantially longer, as further specialization could reduce generalization to new data.



(**A**) (**B**)

Figure 8. Training curves of PTL-Inception model trained for 100 epochs: (**A**) Accuracy, (**B**) Loss.

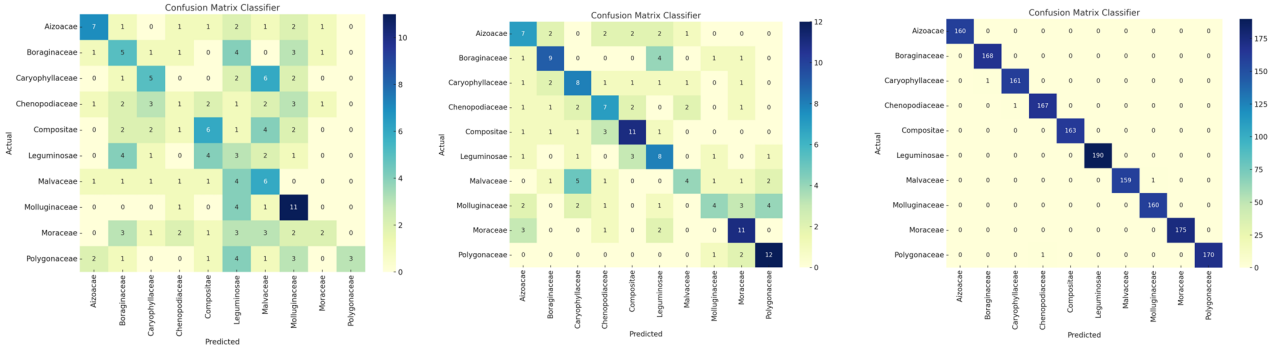
Figure 8B illustrates the training and validation loss values. As expected, the loss curves show an inverse relationship to the accuracy graphs. Both training and validation loss declined sharply during the initial stages, reaching their lowest values by the 18th epoch. This reduction confirms that the model was effectively minimizing the difference between predicted and actual outputs. From the 25th epoch onward, the loss values remained consistently low, coinciding with the plateau observed in the accuracy curves. This stable phase indicates that the model had attained a balanced configuration, where errors were minimized without substantial signs of overfitting.

In addition to overall accuracy and macro-level evaluation, we report per-class precision, recall, F1-score, and support values (Table 8). These results provide a more detailed view of classification robustness across the ten families.

Figure 9 illustrates the confusion matrices of both the InceptionV3 and PTL-Inception models across various datasets. In Figure 9A, the performance of the Inception model is depicted when evaluated on the original dataset. Conversely, Figure 9B showcases the outcomes of the PTL-Inception model assessed on the same original dataset. Notably, Figure 9C provides insights into the performance of the PTL-Inception model when tested on an augmented dataset.

Table 8. Per-class performance of PTL-Inception model.

Class Name	Precision	Recall	F1-Score	Number of Samples
Aizoacae	1.000000	0.993750	0.996865	160
Boraginaceae	1.000000	1.000000	1.000000	168
Caryophyllaceae	0.993865	1.000000	0.996923	162
Chenopodiaceae	0.988166	0.994048	0.991098	168
Compositae	1.000000	0.993865	0.996923	163
Leguminosae	1.000000	1.000000	1.000000	190
Malvaceae	1.000000	1.000000	1.000000	160
Molluginaceae	1.000000	0.993750	0.996865	160
Moraceae	0.994318	1.000000	0.997151	175
Polygonaceae	1.000000	1.000000	1.000000	171



(**A**) (**B**) (**C**)

Figure 9. Confusion matrices of (**A**) InceptionV3 tested on the original dataset, (**B**) PTL-Inception Model tested on the original dataset, (**C**) PTL-Inception Model tested on the augmented dataset.

Analysis of Figure 9B reveals instances of incorrect predictions attributed to the limited dataset size. However, upon subjecting the proposed model to a substantial amount of unseen data (augmented data), as depicted in Figure 9C, notable improvements in performance are observed. Specifically, the results indicate that the model exhibited commendable performance with the augmented dataset, demonstrating a slight enhancement over its performance with the original dataset.

4.3. Ablation Study

An ablation study was conducted to examine the contribution of successive refinements made to the base InceptionV3 model. The process evaluated the effect of data augmentation, additional layers, transfer learning, hyperparameter tuning, and early stopping. The results are presented in Tables 9 and 10.

Table 9. Ablation Study of PTL-Inception model.

Model	Tuning Technique	Training Accuracy	Training Loss	Validation Accuracy	Validation Loss	Test Accuracy	Test Loss
Model I	InceptionV3	0.9877	0.0337	0.1911	5.3141	0.2671	5.2396
Model II	Model I + Augmentation	0.9937	0.0241	0.9420	0.1905	0.9481	0.1679
Model III	Model II + Additional Layers	0.9901	0.0362	0.9588	0.1619	0.9660	0.1073
Model IV	Model III + Transfer Learning	0.9946	0.0231	0.9851	0.0805	0.9899	0.0649
Model V	Model IV + Hyperparameter Tuning	0.9976	0.0094	0.9958	0.0179	0.9976	0.0036
PTL-Inception	Model V + Early Stop (in 31 Epochs)	0.9947	0.0180	0.9904	0.0282	0.9946	0.0209

Table 10. Performance metrics of test results.

Model	Modification	Accuracy	Precision	Recall	F1-Score	Loss	Cohen's Kappa (κ)
Model I	InceptionV3	0.2671	0.3596	0.2739	0.2635	5.2396	0.1913
Model II	Model I + Augmentation	0.9481	0.9504	0.9487	0.9483	0.1679	0.9424
Model III	Model II + Additional Layers	0.9660	0.9665	0.9667	0.9662	0.1073	0.9525
Model IV	Model III + Transfer Learning	0.9899	0.9902	0.9897	0.9899	0.0649	0.9834
Model V	Model IV + Fine Tuning	0.9976	0.9976	0.9975	0.9976	0.0036	0.9967
PTL-Inception	Model V + Early Stop	0.9946	0.9946	0.9944	0.9945	0.0209	0.9940

Table 9 shows that the baseline InceptionV3 (Model I) provided limited accuracy and high loss, highlighting the need for improvement. With the inclusion of data augmentation (Model II), performance improved substantially across training, validation, and test sets. Adding ten layers (Model III) led to further gains, with validation accuracy rising from 0.9420 to 0.9588 and test accuracy from 0.9481 to 0.9660.

Transfer learning applied to this configuration (Model IV) increased accuracy to 0.9851 on the validation set and 0.9899 on the test set, demonstrating the benefit of knowledge transfer from pretrained weights. Hyperparameter tuning (Model V), particularly with the Adam optimizer and adjusted learning rate, improved results further, achieving 0.9958 validation accuracy and 0.9976 test accuracy.

Training InceptionV3 for 100 epochs typically required about five hours. The introduction of additional layers extended this time only marginally, by approximately ten minutes. To reduce computational cost, early stopping with a patience parameter of 20 was employed, ending training after 31 epochs in 1.5 h. The resulting PTL-Inception model achieved 0.9904 validation accuracy and 0.9946 test accuracy, while reducing training time by 3.5 h compared with full-length training.

Table 10 summarizes test performance across all configurations. The progression from Model I through Model V illustrates the incremental effect of each modification. The final PTL-Inception model combined these refinements and retained high performance while improving efficiency, with balanced accuracy (0.9946), precision (0.9946), recall (0.9944), F1-score (0.9945) and Cohen's Kappa (0.9940). These results confirm that the combination of augmentation, architectural changes, transfer learning, fine-tuning, and early stopping produced a model capable of robust and reliable plant classification.

To further interpret model predictions and confirm that PTL-Inception relied on taxonomically relevant features, we generated explainability visualizations using Grad-CAM (Figure 10). The results clearly indicated that the model consistently focused on plant morphology (leaf arrangement, floral structures, branching) while minimizing attention to non-relevant background elements such as soil, rocks, or shadows. This provided additional confidence that classification decisions were biologically meaningful rather than being driven by environmental cues.

4.4. K-Fold Cross Validation

Classification models are also examined using k-fold cross-validation to see whether classification algorithm performance is data-dependent. In this study we used 5-fold cross-validation to evaluate classification models. The augmented dataset was randomly split into training and testing sets. Later the training dataset was divided into 5 folds, with 1 fold reserved for validation and the remaining 4 folds used for training. We have run the model 5 times, but each time a different folder is selected for validation. After each run model was tested and performance metrics were recorded to assess the dependence of accuracy on different folds. The model's performance was determined by calculating the average of

performance criteria from these iterations. The obtained result from the 5-fold validation experiment is given in Table 11.

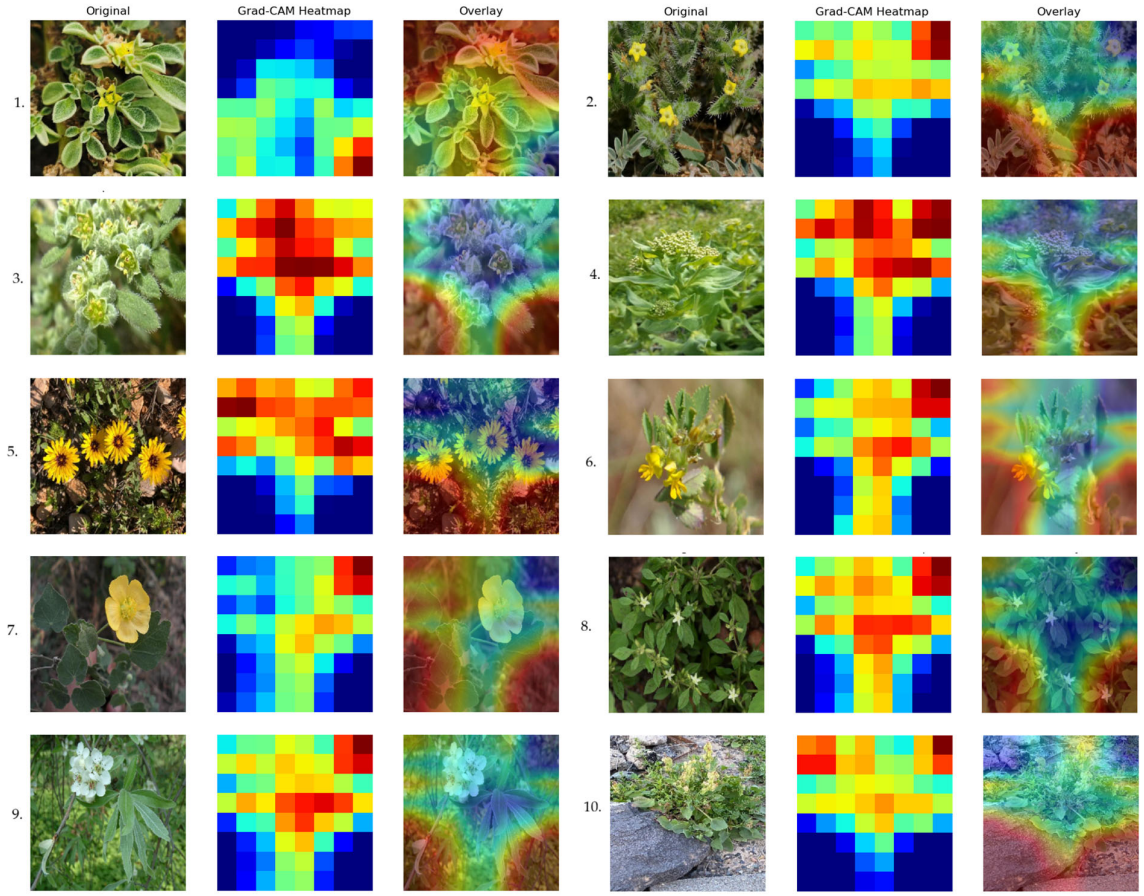


Figure 10. Grad-CAM visualizations of PTL-Inception across 10 representative samples.

Table 11. K-Fold Validation results.

Models	Loop 1	Loop 2	Loop 3	Loop 4	Loop 5	Mean
Training Accuracy	0.9917	0.9921	0.9921	0.9917	0.9929	0.9921
Training Loss	0.0267	0.0247	0.0259	0.0271	0.0199	0.0249
Validation Accuracy	0.9926	0.9931	0.9905	0.9931	0.9942	0.9927
Validation Loss	0.0206	0.0185	0.0215	0.0217	0.0208	0.0206
Test Accuracy	0.9922	0.9922	0.9946	0.9911	0.9899	0.9920
Test Loss	0.0220	0.0216	0.0208	0.0365	0.0339	0.0269
Test Precision	0.9925	0.9923	0.9946	0.9910	0.9900	0.9921
Test Recall	0.9923	0.9922	0.9945	0.9910	0.9897	0.9919
Test F1-Score	0.9923	0.9922	0.9945	0.9910	0.9899	0.9920

Table 11 reports the results of the K-Fold validation, presenting performance measures obtained from the training, validation, and test phases across the different folds. The evaluation includes accuracy, loss, precision, recall, and the F1-score, offering a complete view of the model's predictive behavior.

The training phase showed stable performance across the five folds, with accuracy averaging 0.9921. This consistency indicates that the model was able to learn effectively from the training data in each split. The mean training loss was 0.0249, confirming a steady reduction in the difference between predicted and actual values. In validation, the average accuracy was 0.9927, suggesting strong generalization to unseen data, while the corresponding loss of 0.0206 indicated reliable convergence without signs of overfitting.

On the test data, the model reached an average accuracy of 0.9920, supported by a mean test loss of 0.0269, reflecting stable error minimization. Further assessment using precision, recall, and F1-score confirmed the balance of predictive performance. Precision averaged 0.9921, recall was 0.9919, and the mean F1-score was 0.9920. These values demonstrate that the model maintained an even trade-off between accuracy and sensitivity, providing confidence in its robustness across the folds.

4.5. Comparative Analysis: Proposed Model Versus Existing Approaches

The performance of the PTL-Inception model was compared with results from earlier studies, as summarized in Table 11. The table reports representative works on plant classification, including the datasets used, the number of classes, the applied models, and the evaluation metrics of accuracy, precision, and recall.

As shown in Table 12, PTL-Inception was trained on a dataset containing ten plant categories and was developed as an extension of the InceptionV3 architecture. The model achieved an accuracy of 99.46%, with precision and recall values of 99.46% and 99.44%, respectively. When examined alongside previous studies, PTL-Inception produced more consistent results across the reported metrics. The improvement may be linked to the introduction of ten additional layers, which strengthened feature learning and improved the separation of plant types. These findings suggest that PTL-Inception can be regarded as a reliable and competitive approach for plant classification when compared with established methods.

Table 12. Comparison of Existing Studies with PTL-Inception Model.

Study	Dataset	Classes	Model	Accuracy	Precision	Recall
Qi et al. [36]	Leaves	4	Xception	70%	70.1%	57.3%
Alirezazadeh et al. [37]	Weed	5	MobileNet	89.77%	86.3%	86.29%
Islam et al. [38]	Plants	7	ResNet50	99.60%	99.60%	99.66%
Quan et al. [39]	Leaves	4	MobileNetV2	97.80%	97.78%	97.61%
PTL-Inception	**Plants**	**10**	**InceptionV3**	**99.46%**	**99.46%**	**99.44%**

4.6. Limitations of the Study

Several limitations of the study should be noted. Although the dataset included ten classes, its overall size remained limited, which may restrict the ability of the model to generalize across a wider variety of species. The number of images per class was also modest, and this may not fully reflect the variability present within each category, particularly for less-represented classes. In addition, PTL-Inception, while extending InceptionV3, still carries structural limitations of the original model that may influence its application in broader classification tasks.

Another limitation arises from the dataset itself, which reflects plants from the ecological conditions of Saudi Arabia. The model's performance in other regions with different vegetation, environmental factors, or background characteristics has not been established. Such variations could influence classification accuracy and therefore require further investigation.

Future research may address these issues by constructing larger and more diverse datasets that include species from a wider range of ecological contexts. The exploration of alternative architectures could also help strengthen robustness and improve the transferability of the approach. In addition, testing under external datasets or domain-shift conditions remains an open task. For practical deployment, future work will also examine the balance between accuracy and computational efficiency by evaluating reduced-resolution inputs,

quantization strategies, and lightweight backbones to enable use on mobile platforms in field surveys.

5. Conclusions

Accurate recognition of desert plant species remains challenging due to limited datasets, complex natural backgrounds, and high similarity among species. In this study, these limitations were addressed through the development of PTL-Inception, a modified InceptionV3 architecture enhanced with ten additional layers, transfer learning, hyperparameter tuning, and early stopping. Trained on an augmented dataset of ten native Saudi Arabian species, the model outperformed baseline approaches and reached 99.46% accuracy, with precision, recall, and F1-score values above 99%. The framework was also confirmed to be efficient, as early stopping reduced training time by more than half while preserving predictive performance.

Beyond computational performance, the study contributes to taxonomy and biodiversity research by providing reliable species-level identifications. These outputs can be integrated with phylogenetic mapping and spatial analyses to examine patterns of diversity, endemism, and ecological adaptation. In addition, the dataset and model provide a foundation for assessing biodiversity across different spatial and ecological scales, and can be extended toward field-based applications where species distribution and richness are monitored in real time. Such integration is of particular value in arid ecosystems, where baseline data remain scarce and conservation challenges are pressing. By complementing traditional taxonomy, PTL-Inception offers a digital tool that can support biodiversity monitoring, ecological research, and conservation planning in fragile desert environments, with future deployment aimed at mobile platforms for practical use in the field.

Author Contributions: Conceptualization, Y.G., Z.Ü., K.Ş. and M.A.; Methodology, Y.G., Z.Ü., K.Ş. and M.A.; Software, Y.G. and Z.Ü.; Validation, Y.G. and Z.Ü.; Formal analysis, Y.G. and Z.Ü.; Investigation, Y.G. and Z.Ü.; Resources, Y.G. and Z.Ü.; Data curation, Y.G.; Writing—original draft, Y.G. and Z.Ü.; Writing—review & editing, Y.G., K.Ş. and M.A.; Visualization, Y.G., Z.Ü., K.Ş. and M.A.; Supervision, Y.G.; Project administration, Y.G.; Funding acquisition, Y.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Deanship of Scientific Research, Vice Presidency for Graduate Studies and Scientific Research, King Faisal University, Saudi Arabia, under Project KFU253481.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: The authors gratefully acknowledge the use of language-editing tools, including Grammarly and ChatGPT, which were employed solely for improving the clarity and readability of the English text. All scientific content, data analyses, and interpretations presented in this manuscript are entirely the work of the authors.

Conflicts of Interest: The author declares that there is no conflict of interest regarding the publication of this paper.

References

1. Kellman, M.C. *Plant Geography*; Routledge: London, UK, 2023; pp. 1–181. [CrossRef]
2. Kaya, A.; Keceli, A.S.; Catal, C.; Yalic, H.Y.; Temucin, H.; Tekinerdogan, B. Analysis of Transfer Learning for Deep Neural Network Based Plant Classification Models. *Comput. Electron. Agric.* **2019**, *158*, 20–29. [CrossRef]

3. Wang, R.; Gamon, J.A. Remote Sensing of Terrestrial Plant Biodiversity. *Remote Sens. Environ.* **2019**, *231*, 111218. [CrossRef]
4. Gaba, S.; Lescourret, F.; Boudsocq, S.; Enjalbert, J.; Hinsinger, P.; Journet, E.P.; Navas, M.L.; Wery, J.; Louarn, G.; Malézieux, E.; et al. Multiple Cropping Systems as Drivers for Providing Multiple Ecosystem Services: From Concepts to Design. *Agron. Sustain. Dev.* **2015**, *35*, 607–623. [CrossRef]
5. Malik, I.; Ahmed, M.; Gulzar, Y.; Baba, S.H.; Mir, M.S.; Soomro, A.B.; Sultan, A.; Elwasila, O. Estimation of the Extent of the Vulnerability of Agriculture to Climate Change Using Analytical and Deep-Learning Methods: A Case Study in Jammu, Kashmir, and Ladakh. *Sustainability* **2023**, *15*, 11465. [CrossRef]
6. Parmesan, C.; Hanley, M.E. Plants and Climate Change: Complexities and Surprises. *Ann. Bot.* **2015**, *116*, 849–864. [CrossRef] [PubMed]
7. Sowers, J.; Vengosh, A.; Weinthal, E. Climate Change, Water Resources, and the Politics of Adaptation in the Middle East and North Africa. *Clim. Change* **2011**, *104*, 599–627. [CrossRef]
8. Jamshidi-Kia, F.; Lorigooini, Z.; Amini-Khoei, H. Medicinal Plants: Past History and Future Perspective. *J. Herbmed Pharmacol.* **2017**, *7*, 1–7. [CrossRef]
9. Khan, F.; Ayoub, S.; Gulzar, Y.; Majid, M.; Reegu, F.A.; Mir, M.S.; Soomro, A.B.; Elwasila, O. MRI-Based Effective Ensemble Frameworks for Predicting Human Brain Tumor. *J. Imaging* **2023**, *9*, 163. [CrossRef]
10. Sharma, M.; Kumar, C.J.; Singh, T.P.; Talukdar, J.; Sharma, R.K.; Ganguly, A. Enhancing Disease Region Segmentation in Rice Leaves Using Modified Deep Learning Architectures. *Arch. Phytopathol. Plant Prot.* **2023**, *56*, 1555–1580. [CrossRef]
11. Sharma, M.; Kumar, C.J.; Talukdar, J.; Singh, T.P.; Dhiman, G.; Sharma, A. Identification of Rice Leaf Diseases and Deficiency Disorders Using a Novel DeepBatch Technique. *Open Life Sci.* **2023**, *18*, 20220689. [CrossRef]
12. Sharma, M.; Kumar, C.J. Improving Rice Disease Diagnosis Using Ensemble Transfer Learning Techniques. *Int. J. Artif. Intell. Tools* **2022**, *31*, 2250040. [CrossRef]
13. Gulzar, Y. Enhancing Soybean Classification with Modified Inception Model: A Transfer Learning Approach. *Emir. J. Food Agric.* **2024**, *36*, 1–9. [CrossRef]
14. Kolhar, S.; Jagtap, J. Plant Trait Estimation and Classification Studies in Plant Phenotyping Using Machine Vision—A Review. *Inf. Process. Agric.* **2023**, *10*, 114–135. [CrossRef]
15. Rao, M.S.; Kumar, S.P.; Rao, K.S. A Review on Detection of Medical Plant Images. *Int. J. Recent Innov. Trends Comput. Commun.* **2023**, *11*, 54–64. [CrossRef]
16. Azlah, M.A.F.; Chua, L.S.; Rahmad, F.R.; Abdullah, F.I.; Alwi, S.R.W. Review on Techniques for Plant Leaf Classification and Recognition. *Computers* **2019**, *8*, 77. [CrossRef]
17. Van Hieu, N.; Hien, N.L.H. Recognition of Plant Species Using Deep Convolutional Feature Extraction. *Int. J. Emerg. Technol.* **2020**, *11*, 904–910.
18. Huixian, J. The Analysis of Plants Image Recognition Based on Deep Learning and Artificial Neural Network. *IEEE Access* **2020**, *8*, 68828–68841. [CrossRef]
19. Li, Y.; Chao, X. Ann-Based Continual Classification in Agriculture. *Agriculture* **2020**, *10*, 178. [CrossRef]
20. Yang, K.; Zhong, W.; Li, F. Leaf Segmentation and Classification with a Complicated Background Using Deep Learning. *Agronomy* **2020**, *10*, 1721. [CrossRef]
21. Beck, M.A.; Liu, C.-Y.; Bidinosti, C.P.; Henry, C.J.; Godee, C.M.; Ajmani, M. An Embedded System for the Automated Generation of Labeled Plant Images to Enable Machine Learning Applications in Agriculture. *PLoS ONE* **2020**, *15*, e0243923. [CrossRef]
22. Bahri, A.; Bourass, Y.; Badi, I.; Zouaki, H.; El Moutaouakil, K.; Satori, K. Dynamic CNN Combination for Morocco Aromatic and Medicinal Plant Classification. *Int. J. Comput. Digit. Syst.* **2022**, *11*, 239–249. [CrossRef]
23. Ghosh, S.; Singh, A.; Kavita; Jhanjhi, N.Z.; Masud, M.; Aljahdali, S. SVM and KNN Based CNN Architectures for Plant Classification. *Comput. Mater. Contin.* **2022**, *71*, 4257–4274. [CrossRef]
24. Keceli, A.S.; Kaya, A.; Catal, C.; Tekinerdogan, B. Deep Learning-Based Multi-Task Prediction System for Plant Disease and Species Detection. *Ecol. Inform.* **2022**, *69*, 101679. [CrossRef]
25. Shelke, A.; Mehendale, N. A CNN-Based Android Application for Plant Leaf Classification at Remote Locations. *Neural Comput. Appl.* **2023**, *35*, 2601–2607. [CrossRef]
26. Ghosh, S.; Singh, A.; Kumar, S. PB3C-CNN: An Integrated Parallel Big Bang-Big Crunch and CNN Based Approach for Plant Leaf Classification. *Intel. Artif.* **2023**, *26*, 15–29. [CrossRef]
27. Lee, C.P.; Lim, K.M.; Song, Y.X.; Alqahtani, A. Plant-CNN-ViT: Plant Classification with Ensemble of Convolutional Neural Networks and Vision Transformer. *Plants* **2023**, *12*, 2642. [CrossRef]
28. Shah, S.A.; Lakho, G.M.; Keerio, H.A.; Sattar, M.N.; Hussain, G.; Mehdi, M.; Vistro, R.B.; Mahmoud, E.A.; Elansary, H.O. Application of Drone Surveillance for Advance Agriculture Monitoring by Android Application Using Convolution Neural Network. *Agronomy* **2023**, *13*, 1764. [CrossRef]
29. Amri, E.; Gulzar, Y.; Yeafi, A.; Jendoubi, S.; Dhawi, F.; Mir, M.S. Advancing Automatic Plant Classification System in Saudi Arabia: Introducing a Novel Dataset and Ensemble Deep Learning Approach. *Model. Earth Syst. Environ.* **2024**, *10*, 2693–2709. [CrossRef]

30. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
31. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
32. Tan, M.; Le, Q. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 6105–6114.
33. Chollet, F. Xception: Deep Learning with Depthwise Separable Convolutions. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR, Honolulu, HI, USA, 21–26 July 2017; pp. 1800–1807.
34. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the Inception Architecture for Computer Vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
35. Gulzar, Y.; Ünal, Z.; Aktaş, H.A.; Mir, M.S. Harnessing the Power of Transfer Learning in Sunflower Disease Detection: A Comparative Study. *Agriculture* **2023**, *13*, 1479. [CrossRef]
36. Qi, M.; Du, F.K.; Guo, F.; Yin, K.; Tang, J. Species Identification through Deep Learning and Geometrical Morphology in Oaks (*Quercus* spp.): Pros and Cons. *Ecol. Evol.* **2024**, *14*, e11032. [CrossRef] [PubMed]
37. Alirezazadeh, P.; Schirrmann, M.; Stolzenburg, F. A Comparative Analysis of Deep Learning Methods for Weed Classification of High-Resolution UAV Images. *J. Plant Dis. Prot.* **2024**, *131*, 227–236. [CrossRef]
38. Islam, M.T.; Rahman, W.; Hossain, M.S.; Roksana, K.; Azpiroz, I.D.; Diaz, R.M.; Ashraf, I.; Samad, M.A. Medicinal Plant Classification Using Particle Swarm Optimized Cascaded Network. *IEEE Access* **2024**, *12*, 42465–42478. [CrossRef]
39. Quan, S.; Wang, J.; Jia, Z.; Yang, M.; Xu, Q. MS-Net: A Novel Lightweight and Precise Model for Plant Disease Identification. *Front. Plant Sci.* **2023**, *14*, 1276728. [CrossRef] [PubMed]

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.