AI

MDPI

Article

Artificial Intelligence in Smart Farms: Plant Phenotyping for Species Recognition and Health Condition Identification Using Deep Learning

Anirban Jyoti Hati [1,2,*] and Rajiv Ranjan Singh [2]

[1] Department of Education and Training, National Institute of Electronics and Information Technology, Jadavpur University Campus, Kolkata 700032, India
[2] Department of Electronics and Communication Engineering, Presidency University, Yehalanka, Bengaluru, Karnataka 560064, India; rajivranjansingh@presidencyuniversity.in
* Correspondence: anirban.jyotihati@presidencyuniversity.in

Abstract: This paper analyses the contribution of residual network (ResNet) based convolutional neural network (CNN) architecture employed in two tasks related to plant phenotyping. Among the contemporary works for species recognition (SR) and infection detection of plants, the majority of them have performed experiments on balanced datasets and used accuracy as the evaluation parameter. However, this work used an imbalanced dataset having an unequal number of images, applied data augmentation to increase accuracy, organised data as multiple test cases and classes, and, most importantly, employed multiclass classifier evaluation parameters useful for asymmetric class distribution. Additionally, the work addresses typical issues faced such as selecting the size of the dataset, depth of classifiers, training time needed, and analysing the classifier's performance if various test cases are deployed. In this work, ResNet 20 (V2) architecture has performed significantly well in the tasks of Species Recognition (SR) and Identification of Healthy and Infected Leaves (IHIL) with a Precision of 91.84% and 84.00%, Recall of 91.67% and 83.14% and F1 Score of 91.49% and 83.19%, respectively.

Keywords: species recognition; health identification; plant phenotyping; deep learning

check for updates

Citation: Hati, A.J.; Singh, R.R. Artificial Intelligence in Smart Farms: Plant Phenotyping for Species Recognition and Health Condition Identification Using Deep Learning. *AI* **2021**, *2*, 274–289. https://doi.org/10.3390/ai2020017

Academic Editor: Yiannis Ampatzidis

Received: 11 April 2021
Accepted: 2 June 2021
Published: 5 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 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 maintain an environmental balance and nourish the atmosphere with their multidimensional contribution to nature. Looking at the possible food crisis in the near future, as reported by the Food and Agriculture Organization of the United Nations (FAO) [1], it is necessary to provide the plants with a better nourishing environment to have a sustainable life cycle. Smart farming helps human beings to have a better degree of control over the nourishment of plants. Plant phenotyping is a technique for quantitative formulation and analysis of complex plant traits, i.e., plant morphology, plant stress, crop yield, plant physiological and anatomical traits, etc. [2]. It is preferred in smart architecture based on efficient and high output farming platforms [3]. Computer vision-based plant phenotyping techniques offer a non-destructive and efficient representation of the complex plant traits [4]. Non-destructive methods have the potential to perform large-scale and high-throughput plant phenotyping experiments. Visible spectral imaging, fluorescence imaging, infrared imaging, hyperspectral imaging, three-dimensional imaging, and laser imaging are some of the popular methods used in these experiments [3,5]. Figure 1 represents different plant phenotyping categories where imaging technique plays an important role [3,6,7]. Visible spectral imaging has the advantages of affordability and quick measurement [8]. It can also model a wide range of plant traits. Before the comprehensive assessment of plant traits, computer vision-based recognition of plant species is required. Plant health condition analysis is also an integral part of the phenotypic analysis. This article has developed

a computer vision-based plant species recognition and health condition identification technique analysing plant leaf images. It is generally designed using standard classification methodologies. Figure 2 elaborates the process flow.

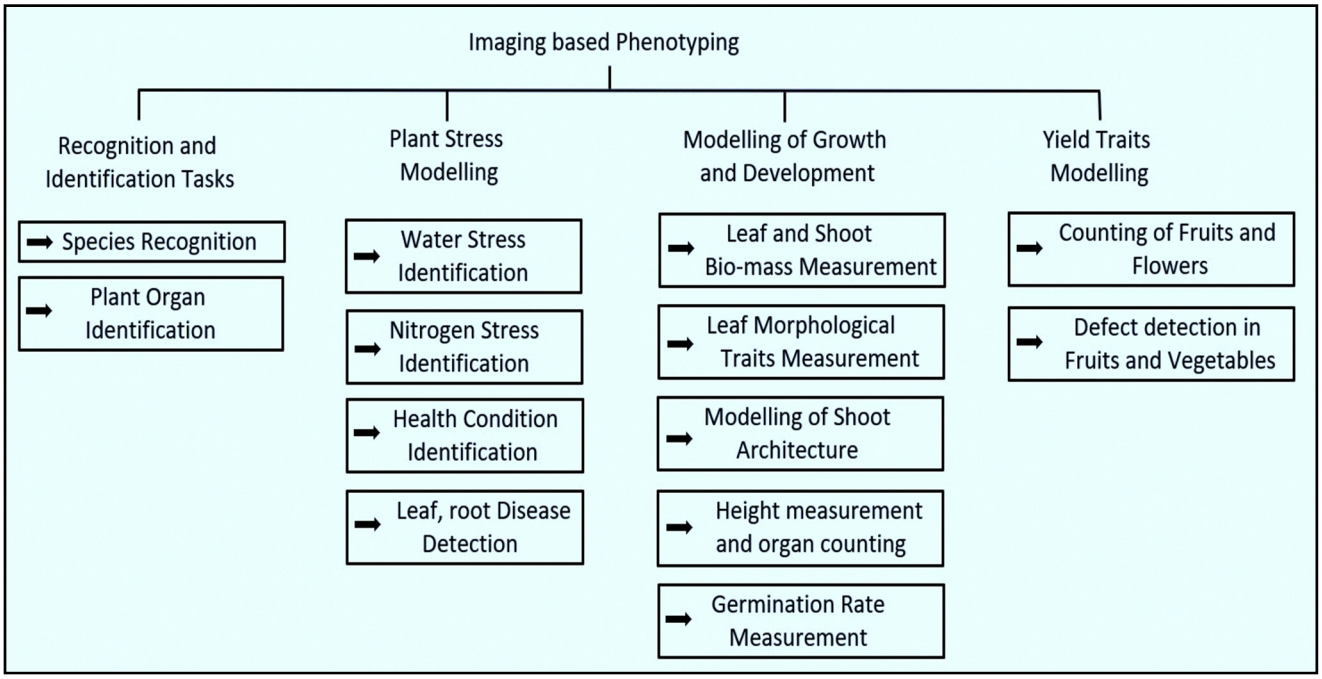

Figure 1. Computer-vision based plant phenotyping categories.

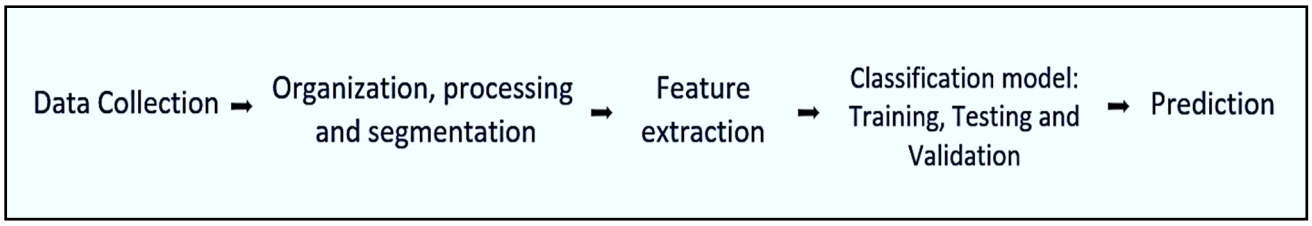

Figure 2. Process flow of computer vision-based classification methods.

Collection of relevant images is the primary challenge for which digital cameras, charged couple device (CCD) cameras, mobile cameras, cameras with portable spectro-radiometers, etc., are used [9]. In the pre-processing step, inappropriate data images are filtered out, and the relevant images are resized, denoised and segmented to get a more accurate classification result. Sometimes the classifier models error and noise present in the data as the original concept and causes overfitting. Data augmentation, i.e., enlargement of the original dataset by adding synthetically generated data, is an accepted approach by the researchers to overcome this problem [10]. The next step is feature extraction of the images. Generally, plant parts affected with some disease show deformation in their colour, texture and shape. Hue histogram, Speeded Up Robust Features (SURF), Histogram of Oriented Gradients (HOG), Scale Invariant Feature Transform (SIFT), etc., are the features used for this purpose [11]. Local descriptors like Bags of Visual Words (BOVW), Histogram of Oriented Gradients (HOG) are used for plant recognition using deep learning (DL) [12]. Classification algorithm learns from the input data features and fits a model which can predict target classes. The whole input dataset is divided into training, testing and validation. Initially, the model parameters are fit based on training data. Validation data helps to tune the model's hyperparameters, and finally, the test data provides an evaluation methodology of the model. Researchers have used supervised, unsupervised and other classification techniques for plant recognition and disease identification [9].

2. Related Work

This article analyses two particular applications (i.e., plant species recognition and health condition identification) of visual spectrum imaging-based plant phenotyping using deep learning (DL) methods. Researchers prefer fours type of deep learning-based methodologies (i.e., convolutional neural network (CNN), deep belief network, recurrent neural network (RNN), and stacked autoencoder) for this purpose [13]. Lee et al. (2017) demonstrated that the use of DL for harvesting important leaf features is effective and can be successfully used for plant identification purposes [14]. In other research, Zhang, S., and Zhang (2017) showed that plant species recognition using a deep convolutional neural network (DCNN) solves the problem of weak convergence and generalisation [15]. Thus, DL algorithms perform better than generic classification algorithms, which use colour, shape, and texture-based features. DL-based plant disease severity assessment achieved good accuracy and was used to predict yield loss [16]. Table 1 presents state-of-the-art research on this field. It shows there is a scope of research and analysis of the following aspects in the context of plant species recognition and health identification:

i. There is a scope to analyse the performance of DL models when fed with an imbalanced dataset, especially when there is a significant difference in the number of leaf images present in each class.
ii. The performance change of the model with the size of the leaf image dataset requires analysis.
iii. Further research is required to map the change in classification accuracy with differences in the DL classifier's depth.
iv. A potential analysis is required to record the change in DL model's performance with an increased number of classes or increased number of leaf images in each class.
v. Computational time in an affordable experimental setup will better visualise application platforms where the model can be deployed.

This paper addresses state-of-the-art issues by organising the imbalanced dataset, tuning the depth of a DL classifier based on performance and computational time, fixing the training data size, and including a number of multiclass classifier's evaluation parameters.

Table 1. State-of-the-art research.

Article	Research Area	Dataset	Methodology	Remarks
[17]	Plant species identification with small datasets	32 species from FLAVIA dataset, 20 species from CRLEAVES dataset of leaf images	Convolutional Siamese network (CSN) and a CNN with three convolutional blocks and a convolutional layer with 32 filters	Classifiers trained with small training samples (5 to 30 per species) and got accuracy of 93.7% and 81% by CSN at two different experimental scenarios
[18]	Plant identification in a natural scenario	BJFU100 dataset of 10,000 images of 100 plant species. The images were collected using mobile devices	Residual network (ResNet) classifier with 26-layer architecture	91.78% accuracy was achieved
[19]	Plant species classification	Dataset with 43 plant species with 30 image samples each	Feature extraction using pre-trained AlexNet, fine-tuned AlexNet, a proposed CNN model (D-leaf), and vein morphometric. Classification using artificial neural network (ANN), support vector machine (SVM), k-nearest neighbours (KNN)	The proposed method with ANN classifier achieved the highest accuracy of 94.88%

Table 1. *Cont.*

Article	Research Area	Dataset	Methodology	Remarks
[20]	Leaf species recognition using DL	Plant leaf dataset of 240 images of 15 different species	AlexNet architecture, fine-tuning of hyperparameters has been done.	Research achieved an accuracy of 95.56%
[21]	Grape plant species identification	Two vineyard image datasets of six varieties of grape with 23 and 14 images each	AlexNet architecture with transfer learning	An accuracy of 77.30% was achieved
[22]	Plant disease diagnosis	Open dataset of 87,848 images with 25 different plants	Five CNN based architecture, i.e., AlexNet, AlexNetOWTBn, GoogLeNet, Overfeat and VGG have been used to classify 58 different classes of healthy and diseased plants	Achieved an accuracy of 99.53%
[23]	Plant Disease Recognition	Experimented on 8 plants and 19 diseases from a dataset of 40,000 leaf images	AlexNet, VGG16, ResNet, Inception V3 used for feature extraction and proposed two-head network for classification.	Achieved 98.07% accuracy on plant species recognition and 87.45% accuracy on disease classification
[24]	Recognition of disease and pests of tomato plants	Dataset contained 5000 images. The method applied for 10 different classes.	Deep learning meta-architectures such as faster region-based CNN, region-based fully convolutional network (R-FCN) and single shot multibox detector (SSD) used for detection and VGG net, ResNet based feature extraction	The research reports the highest average precision of 85.98% with ResNet50 and R-FCN
[25]	Identification of plant disease	14 plant species with 26 diseases from the PlantVillage dataset were used for recognition. Total number of images is 54306.	DL classifiers such as VGG net, ResNet, Inception V4, DenseNet were used. The DL models have been fine-tuned for the process of Disease Identification.	An accuracy of 99.75% has been achieved using DenseNet

3. Materials and Methods

This section discusses the dataset, pre-processing of the images, organisation of the dataset, and the classification methodology adopted for the species recognition and health condition identification task.

3.1. The Dataset

A data repository [26] of segmented leaf images of 12 different plant species was selected for this purpose. The presence of a wide variety of species in the dataset increases the variability among it. The acquisition of images was made in a smart enclosed environment using a Nikon D5300 camera. It has 4503 images in which 2278 images are of healthy leaves, and 2225 images are of leaves infected with different diseases. Table 2 reflects the details of the dataset. Figure 3 shows a healthy and diseased leaf image sample of each species.

Table 2. The number of image samples present in the dataset.

Plant Species	Number of Images (H: Healthy, I: Infected)	Plant Species	Number of Images (H: Healthy, I: Infected)
Mango (*Mangifera indica*)	H:170, I:265	Jatropha (*Jatropha curcas* L.)	H:133, I:124
Arjun (*Terminalia arjuna*)	H:220, I:232	Sukh chain (*Pongamia Pinnata* L.)	H:322, I:276
Alstonia (*Alstonia scholaris*)	H:179, I:254	Basil (*Ocimum basilicum*)	H:149, I:0
Guava (*Psidium guajava*)	H:277, I:142	Pomegranate (*Punica granatum* L.)	H:287, I:272
Bael (*Aegle marmelos*)	H:0, I:118	Lemon (*Citrus limon*)	H:159, I:77
Jamun (*Syzgium cumini*)	H:279, I:345	Chinar (*Platanus orientalis*)	H:103, I:120

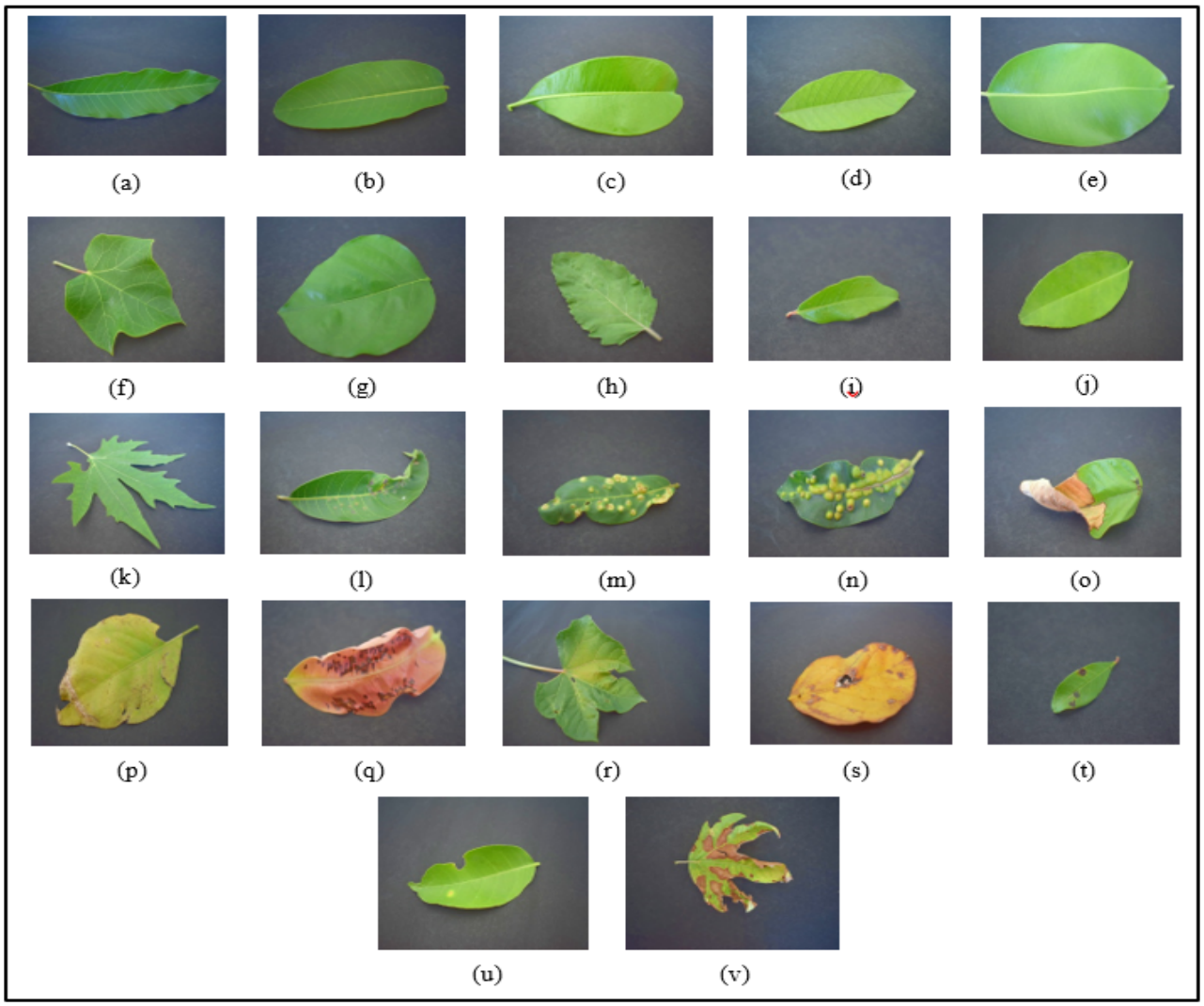

Figure 3. Image samples post pre-processing stage (**a**) Mangifera indica-H, (**b**) Terminalia arjuna-H, (**c**) Alstonia scholaris-H, (**d**) Psidium guajava-H, (**e**) Syzgium cumini-H, (**f**) Jatropha curcas L.-H, (**g**) Pongamia Pinnata L.-D, (**h**) Ocimum basilicum-H, (**i**) Punica granatum L.-H, (**j**) Citrus limon-H, (**k**) Platanus orientalis-H, (**l**) Mangifera indica-D, (**m**) Terminalia arjuna-D, (**n**) Alstonia scholaris-D, (**o**) Psidium guajava-D, (**p**) Aegle marmelos-D, (**q**) Syzgium cumini-D, (**r**) Jatropha curcas L.-D, (**s**) Pongamia Pinnata L.-D, (**t**) Punica granatum L.-D, (**u**) Citrus limon-D, (**v**) Platanus orientalis-D, where H indicates Healthy leaf sample and D indicates Diseased leaf.

3.2. Pre-Processing of the Dataset

Figure 4a,b shows samples of healthy and infected leaves before feeding into the pre-processing stage. The images of the dataset were represented in the red, green and blue (RGB) colour model. In the pre-processing stage, representation was changed from RGB to the HSV (hue saturation value) colour space. A threshold value is added to the V-value, which increases the brightness level of the images. Enhancing brightness makes the dark spots on leaves and the infection patches easily differentiable. Figure 4c shows an image sample after enhancing brightness. Images were resized to $224 \times 224 \times 3$ before feeding them to the classifier.

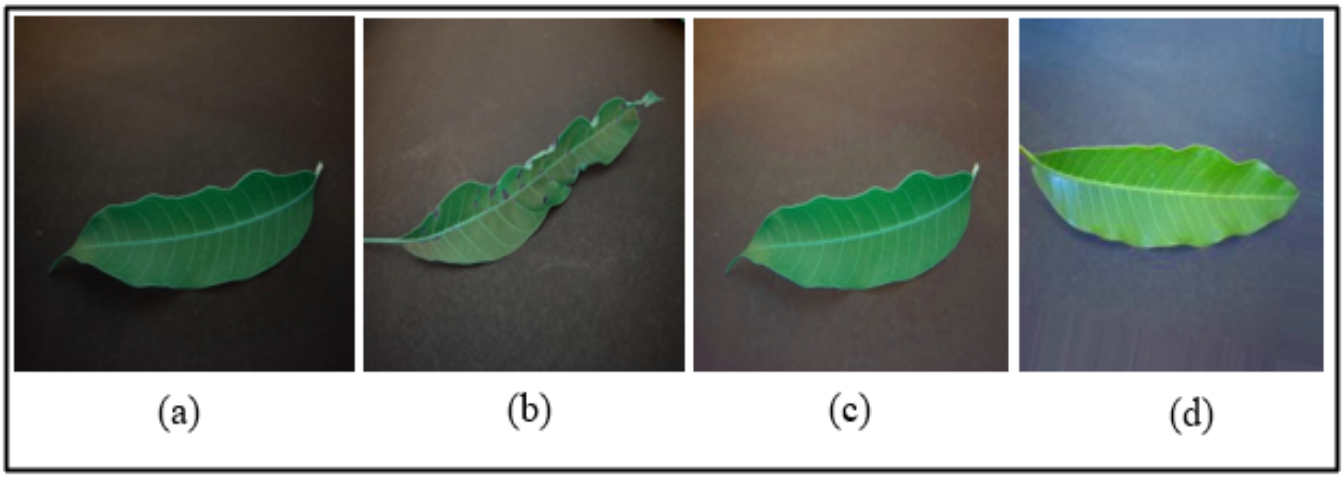

Figure 4. Sample of images present in the dataset (**a**) Healthy mango leaf, (**b**) Infected mango leaf, (**c**) Image after enhancing the brightness of 4 (**a**) image, (**d**) artificially generated mango leaf.

3.3. Organization of the Dataset for Training

The dataset was organised as two test cases, as shown in Figure 5. Each of the healthy and infected sets of data of different plant species were considered a class in this experiment. Infected lemon had a minimum of 77 image samples, and there was an average of 205 image samples in all the classes. For the first test case, the dataset was organised with 77 image samples from every class. The classes with more than 77 image samples were under-sampled through a random sampling method. A dataset was prepared with 205 image samples from every class in the second case. The classes with more than 205 image samples were under-sampled, and classes with fewer than 205 images were provided with artificially generated images using the image augmentation method. Original images were horizontally or vertically flipped, rotated, shifted, and changed in their brightness level to create artificial variations, as shown in Figure 4d.

3.4. Classification

AlexNet, GoogLeNet, ResNet, Inception V3, Inception V4, VGG-16, VGG-19, etc., were some of the popular DL architectures used for classification [27]. There are many criteria on which a DL model can be selected for a particular application. Canziani et al. (2016) have done extensive research and analysis comparing the performance of state-of-the-art DL models on which a model could be selected for practical applications [28]. Inference time for input data sample on DL architecture and its changes across different batch size is significant in this context. It has been observed that the number of operations and inference time have a linear relationship. DL models with low inference time, limited operations count, and low power consumption are suitable for real-time and resource-constrained applications. AlexNet, which is also considered the first modern CNN architecture, has the lowest inference time with increasing batch size, limited operations count, and low power consumption. Accuracy and utilisation of parameters are criteria that are significant in determining the performance of the model. ResNet is one such DL network that has

reported good classification accuracy with standard datasets. It also has a high capacity to use the parametric space.

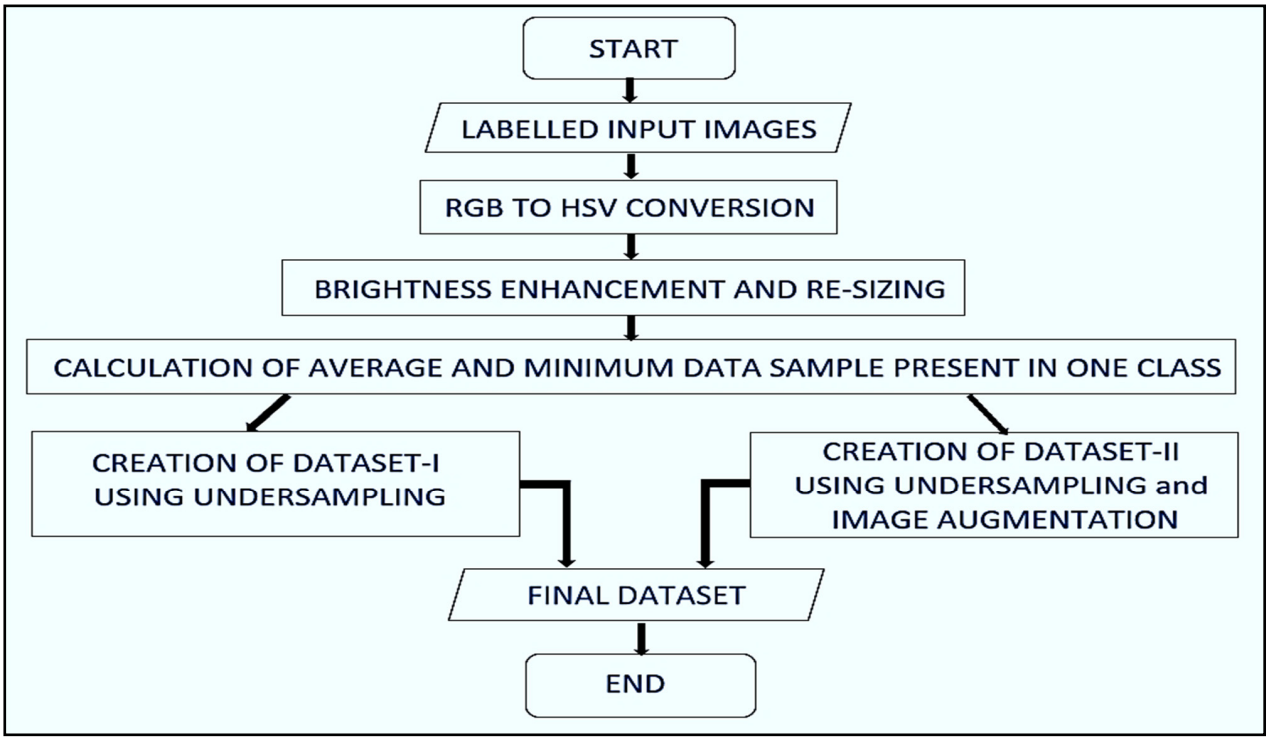

Figure 5. Steps for pre-processing and organisation of the original dataset for generation of Dataset I and Dataset II.

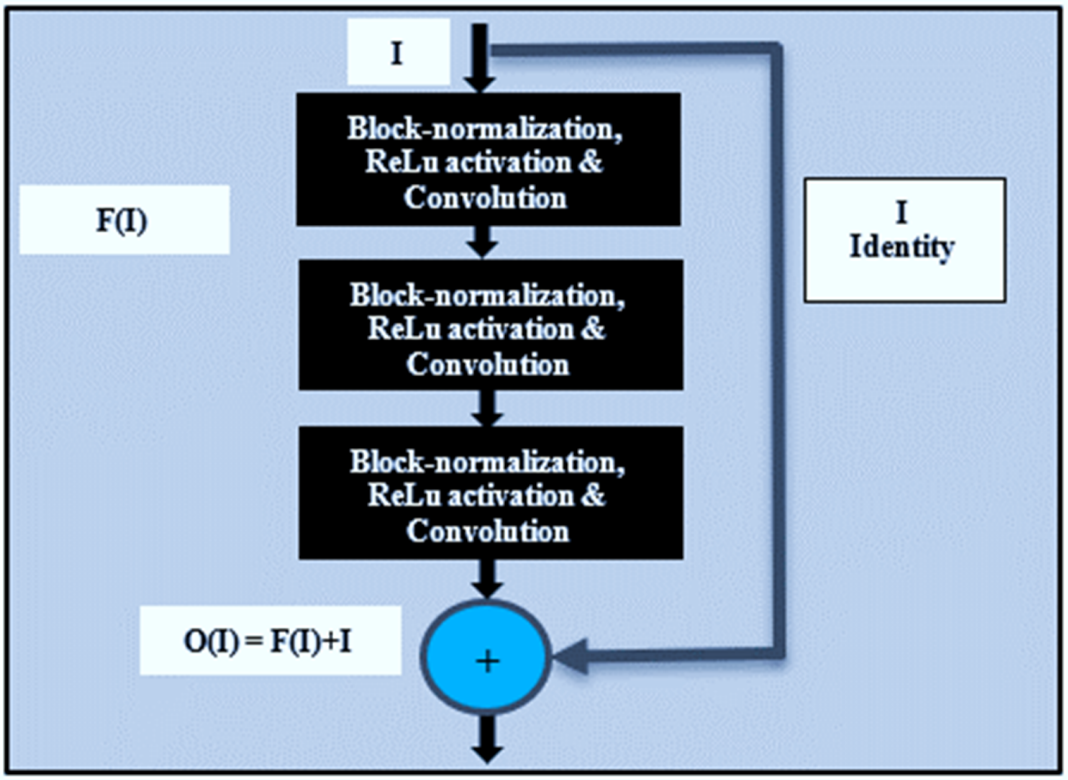

Figure 6. The basic building blocks of residual network version 2: residual learning.

In this article, we have used the residual network (ResNet) (Version 2) based convolutional neural network (CNN) architecture for classification and compared the species recognition results with AlexNet so that the outcome can be used in a wide variety of platforms. The generic CNN architecture contains a series of convolutional layers and filters, pooling layers, fully connected (FC) layers, and a softmax classifier. Convolutional layers with filters extract features from the input images. Padding is used to fit the filter into the image. An activation layer is applied after the convolution layer. Generally, non-linear functions, such as the hyperbolic tangent function, sigmoid, and rectified linear unit (ReLu), are used to introduce nonlinearity in CNN. In ResNet, the ReLu activation function is preferred. The pooling layer reduces the number of parameters while retaining the required information. Fully connected layers convert the output of the previous layers into a single vector before feeding it to the classifier.

ResNet was first introduced at the ImageNet classification challenge in 2015 [29]. In the process of classification, deeper networks have been used to improve the classification performance. He et al. (2016) reported that adding more layers can cause training errors, resulting in accuracy degradation. The problem is addressed in ResNet by using deep residual learning. Fitting the stacked layer into residual mapping is easier to optimise than unreferenced mapping. The building blocks of the residual leaning of improved ResNet (or ResNet Version 2) is shown in Figure 6 [30]. ResNet introduced the identity path I through which the input of the block is added to the output of the block, i.e., $O(I) = F(I) + I$. The abstractions modelled in the previous layer are forwarded to the next layer through the identity path. Hence the incremental abstractions are easily built on top of the existing one. Each building block models only the incremental abstraction $F(I) = O(I) - I$ thus eliminating the degradation error. In ResNet V2, block normalisation and ReLu activation are performed before the convolution operation. Figure 7 elaborates the architecture of ResNet with the convolutional layers having filters and the identity shortcuts. The ResNet (V2) model follows the form of (6a + 2) number of layers which define the depth of the network [30]. We have compared for a = (1, 2, 3, 4, and 5) which give a 11, 20-, 29-, 38-, and 47-layer networks.

AlexNet architecture was first introduced in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 by Krizhevsky et al. (2016). It has eight learned layers, i.e., five convolutional layers and three fully connected layers [31]. The input images of $224 \times 224 \times 3$ pixels are filtered with a kernel of dimension 11X11X3 by the first convolutional layer. A total of 96 kernels are used, and the stride of filtering is maintained at four. The response is normalised, and overlapping pooling is used to create a summarised kernel map. The second convolutional layer uses 256 kernels of dimension $5 \times 5 \times 48$ for filtering. The response of the second layer is normalised and max-pooled. The third, fourth, and fifth convolutional layers use 384 kernels of dimension $3 \times 3 \times 256$, 384 kernels of dimension $3 \times 3 \times 192$, and 256 kernels of dimension $3 \times 3 \times 192$, respectively, for filtering. Normalisation or pooling is not applied to the response of the final three convolutional layers. fully connected layers have 4096, 4096, and 1000 connected neurons, respectively.

We have used five convolutional layers and five fully connected layers (Figure 8). The fourth fully connected layer has 100 connected neurons. The final layer uses the softmax function and generates the classification output based on the number of the target class. AlexNet reported better accuracy in ImageNet classification compared to the existing algorithm. A rectified linear unit (ReLu) is used as an activation function. It is more time-efficient hence takes less training time compared to other activation functions such as sigmoid functions. Overfitting reduces the classification efficiency of deep neural networks. The concept of dropout is used in AlexNet where the output of an individual neuron is dropped out based on a certain probability to avoid overfitting.

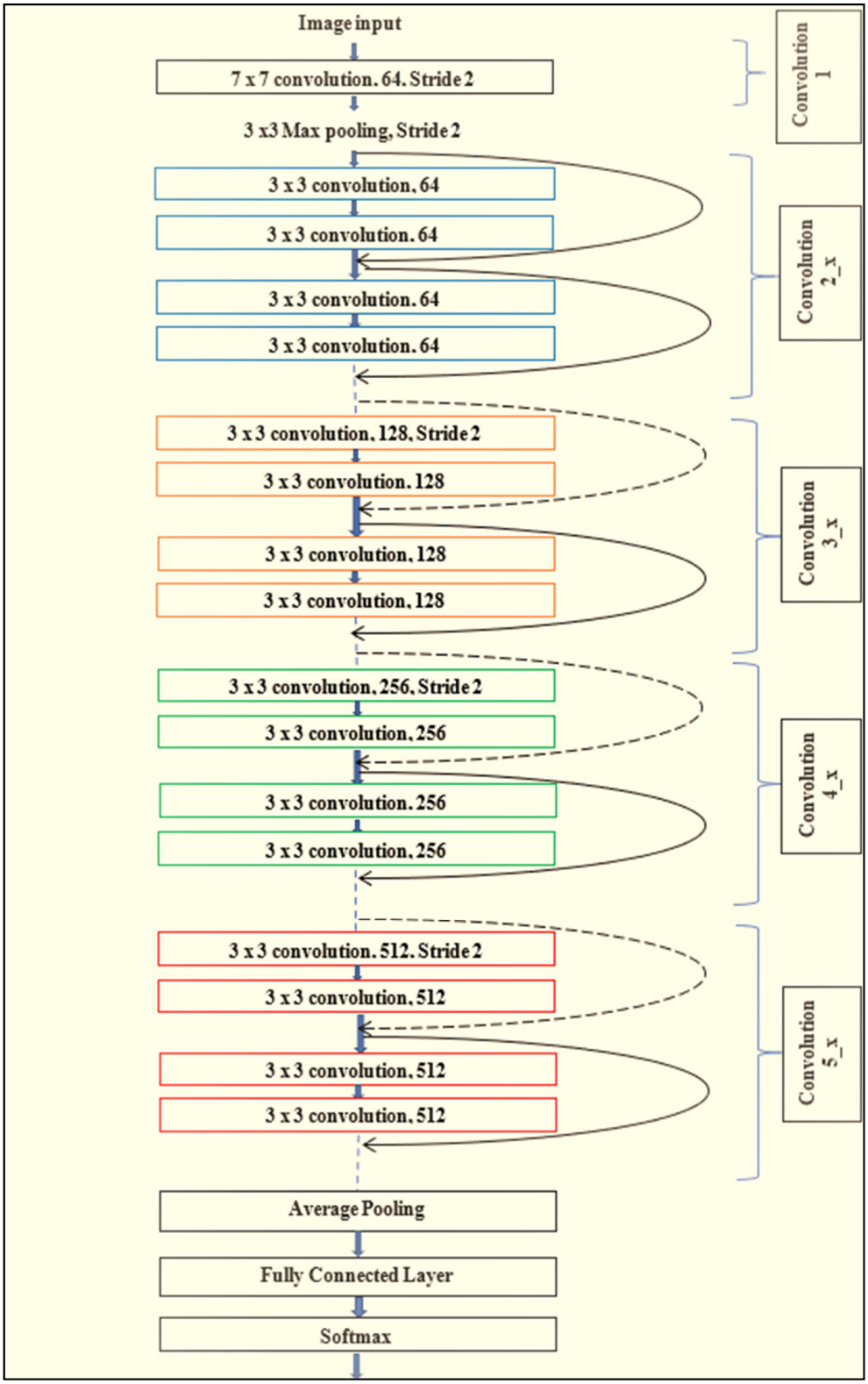

Figure 7. The architecture of ResNet; where x refers to an individual building block and the number of building blocks depends on the total number of layers present in the architecture.

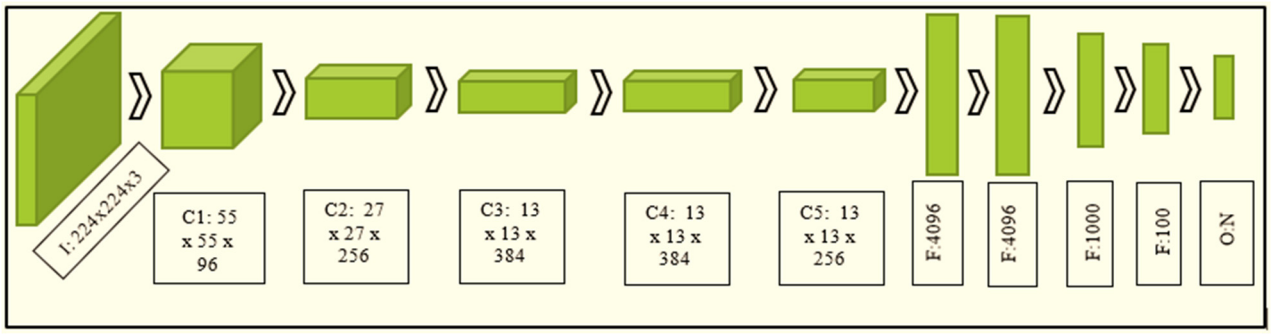

Figure 8. AlexNet-based architecture used in the experiment: I: input, C: convolutional layer, F: fully connected later, O: output layer with softmax classifier, N: the total number of the class.

3.5. Implementation

The proposed methodology has been implemented using Python, which is an interpreter based high-level programming language. Keras [32], an open-source neural network library, has been used to implement ResNet. Libraries like Open Cv [33] and Scikit-Image [34] have been used for other image processing tasks. The entire dataset is passed to the neural network in several epochs (a single epoch is a single cycle of learning) to complete the learning process. The required number of epochs to achieve the learning process varies in different training circumstances. If the model is trained through a greater number of epochs, it can result in overfitting, and for fewer epochs, it may go into underfitting. Here we have used the Early Stopping method from the Keras library. It stops training whenever the validation process indicates a saturation in the performance of the model. The ModelCheckpoint callback has also been incorporated into the DL based classifier to save the best performing model after every epoch.

4. Results, Analysis, and Comparison

Nine different test cases have been designed to perform the following tasks on (1) species recognition (SR) and (2) identification of healthy and infected leaves (IHIL). For species recognition, the input datasets are classified into 12 different classes that correspond to 12 separate species (Table 2). Each of the plant species except Bael and Basil has a set of healthy and infected images. Hence for the second task, the datasets are classified into 22 different classes. The terminology of the test cases is as follows: UN1_RN2. UA can be used alternatively with U, and Alex can be used instead of R. The details are elaborated in Table 3.

Table 3. The terminology of the test cases.

Character/Number	Details
U	Dataset generated by under-sampling (i.e., Dataset-I)
UA	Dataset generated by under-sampling and augmentation (i.e., Dataset-II, Figure 3)
N1	12 for SR and 22 for IHIL
R	ResNet version 2 based DL classifier has been used.
Alex	Alexnet based DL classifier has been used.
N2	Indicates depth of Residual Network based classifier.

The input images are classified into one of the n different classes, i.e., C_i where $1 \leq i \leq n$. The total number of classes is n which is 12 for SR and 22 for IHIL. Figure 9 represents the confusion matrix of a multiclass classification with n classes [35]. The efficiency of classifiers has been evaluated using confusion matrix-based performance metrics. The relevant parameters such as tn, tp, fn, and fp are elaborated in Table 4. The performance metrics relevant for multiclass classification are elaborated in Table 5 [36]. The parameters such as tp_i, tn_i, fp_i, and fn_i are the counts of true positive, true negative, false

positive and false negative, respectively, for class C_i. Table 6 lists the value of performance metrics (macro-average) for each of the test cases. Macro-averaging assigns equal weightage to all classes, hence, avoids disfavouring Bael and Basil classes for the task of SR. Accuracy is a better performance metric for symmetric class distribution (i.e., false positives and false negatives have almost the same cost), for asymmetric class distribution precision, recall and F1 score reflect the performance better. F1 score is the harmonic mean of precision and recall. It gives a balanced measure and is more suitable to reflect the performance of DL model. Hence, F1 score has been given priority in our analysis. ResNet 20 (V2) has reported the best Accuracy and F1 Score among all test cases.

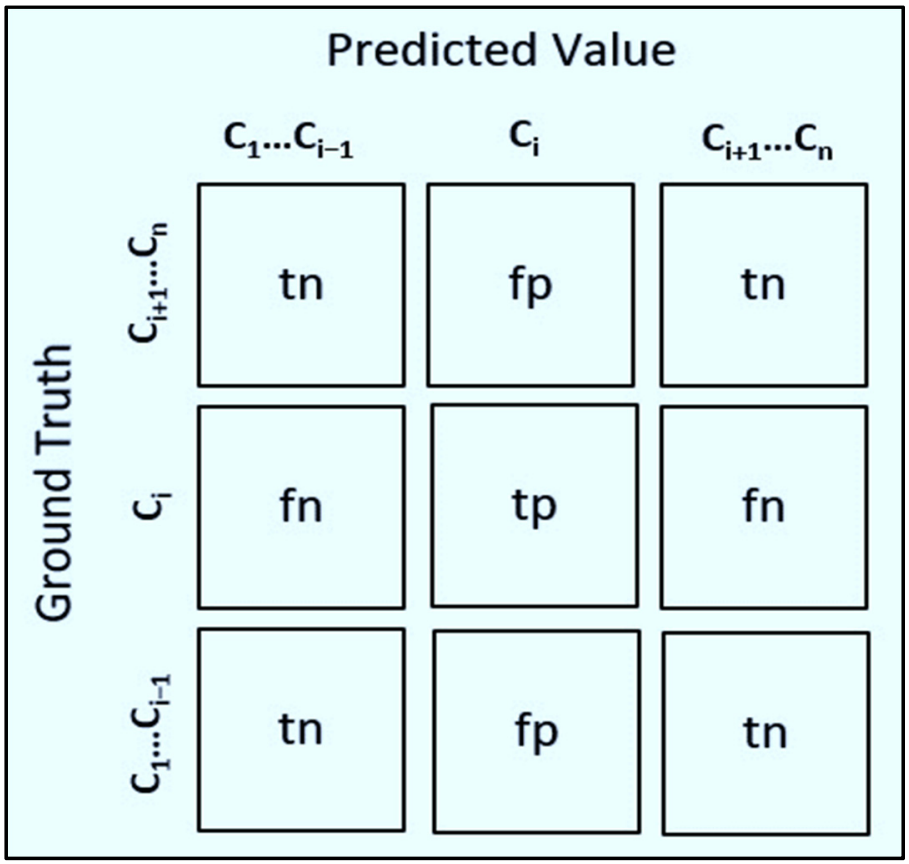

Figure 9. Confusion matrix of n classes (n = 12 for SR and 22 for IHIL) considering the class C_i, where $1 \leq i \leq n$.

Table 4. Relevant parameters used in confusion matrix-based performance metrics.

	Relevant Parameters
True positive (tp)	The number of class examples that are correctly predicted.
True negative (tn)	The number of correctly recognised examples that do not belong to the class
False positive (fp)	The number of predicted class examples that do not truly belong to the class.
False negative (fn)	The number of class examples which the classifier fails to recognise.

Table 5. Performance metrics used for analysing the performance of the classifiers.

Metrics	Mathematical Expression	Remarks
Average accuracy	$\frac{\sum_{i=1}^{n}(tp_i+tn_i)}{\sum_{i=1}^{n}(tp_i+tn_i+fp_i+fn_i)}$ / n	Average of per class ratio of correct prediction to total test samples
Precision	$\frac{\sum_{i=1}^{n} tp_i}{\sum_{i=1}^{n}(tp_i+fp_i)}$ / n	Indicates how accurate the classifier is among those predicted to be class examples
Recall	$\frac{\sum_{i=1}^{n} tp_i}{\sum_{i=1}^{n}(tp_i+fn_i)}$ / n	Indicates how accurate the classifier is for predicting the true class examples
F1 Score	$\frac{2 \times (Precision \times Recall)}{Precision + Recall}$	Indicates the balanced average of both precision and recall

Table 6. Results of all test cases.

Test Cases	Task	Average Accuracy (in %)	Precision_M (in %)	Recall_M (in %)	F1 Score_M (in %)
U12_R11	SR	85.98	85.99	86.46	85.59
U12_R20	SR	90.53	92.13	90.62	90.89
U12_R29	SR	87.12	87.63	86.81	86.49
U12_R38	SR	89.39	89.38	89.24	89.11
U12_R47	SR	86.36	86.58	86.11	85.53
U22_R20	IHIL	81.44	83.74	81.44	81.13
UA12_R20	SR	91.94	91.84	91.67	91.49
UA22_R20	IHIL	83.14	84.00	83.14	83.19
U12_Alex	SR	81.06	76.85	75.32	74.87

An analysis and comparison of the performances of the DL models are reflected in Figure 10. With increasing the depth of CNN up to a specific limit, the optimisation capabilities increase. As shown in Figure 10a, the F1 Score gets better up to 20 layers in this case, beyond which the classifier requires more training data to perform better. Increased depth also multiplies the time taken per epoch, as shown in Figure 10b. Furthermore, the time consumption, as shown in Figure 10b is system dependent, and it will vary if experiments are performed in systems with different specifications. These experiments have been performed on a system with an i5 CPU @1.60 GHz, 8 Gb RAM.

The task of health condition identification of different plants includes the task of species recognition. Hence a better F1 Score is reported (as shown in Figure 10c) when the training data has 12 classes compared to 22 classes. When similar classes are combined during training, it reduces the number of misclassifications. More training images may enhance the feature discrimination power of the classifier. To test it, Dataset-I and Dataset-II have been fed to the same classifier. The test case with Dataset-II reports the best accuracy and F1 score as shown in Figure 10d, which signifies the importance of the augmentation method adopted here. AlexNet requires fewer computations hence has a lower computation time, and it also reports a lower F1 score for SR than ResNet 20 (V2). The performance comparison of ResNet 20 (V2) and AlexNet are shown in Figure 10e.

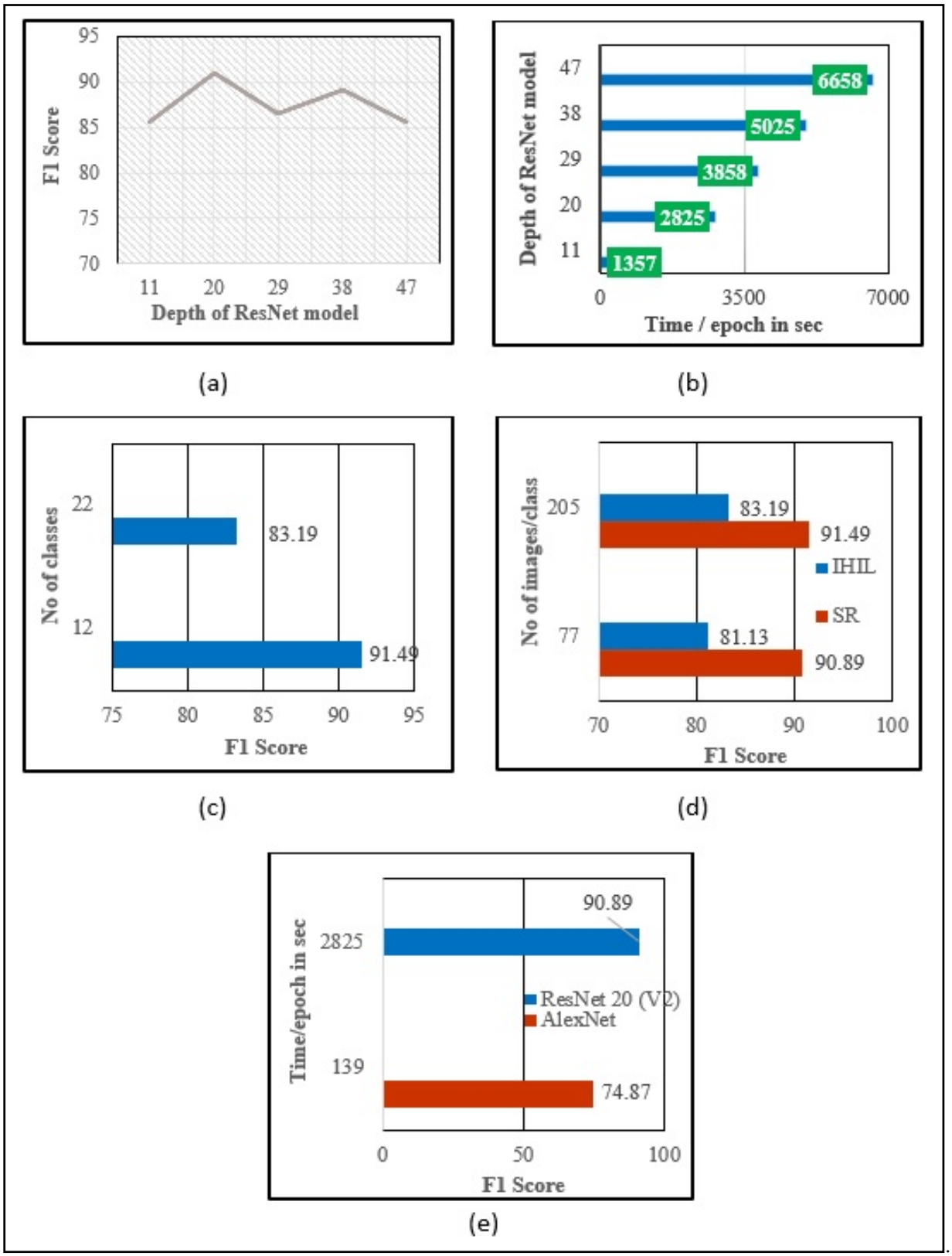

Figure 10. Performance analysis of the classifiers. (**a**) Change in F1 score with the changing depth of ResNet, (**b**) computational time in seconds of ResNet with different depths, (**c**) change in F1 Score with increased number of classes, (**d**) comparison of Dataset I & Dataset II, (**e**) Comparison of ResNet and AlexNet model. ResNet20 V2 was used for (**c**–**e**).

5. Discussion and Conclusions

In this article, species recognition and plant health condition identification have been performed in different experimental scenarios. A comprehensive analysis of the performance of DL models was provided using multiclass classification-based performance metrics. The leaf image dataset had an unequal number of images present in each class, i.e., a minimum of 77 images and a maximum of 345 images. Under-sampling and data augmentation methods have been used to deal with the imbalanced dataset and diversify the training set. The dataset with synthetic images added achieved a higher F1 Score than the dataset with fewer images, i.e., 0.6% higher in species recognition and 2.6% higher in health condition identification. The ResNet classifier provides a solution for degradation of accuracy with deeper networks. ResNet 20 (V2) gave the highest F1 Score of 91.49% for SR and 83.19% for IHIL. State-of-the-art DL models have shown that with increasing depth, over parameterisation can cause overfitting. DL models with higher depths than ResNet 20 have recorded lower F1 Scores. If the number of classes is increased without increasing the training samples, the classifier's performance may degrade. ResNet 20 (V2) reported an 8.3% higher F1 Score with 12 classes than with 22 classes. The computational time of AlexNet was approximately 20 times lower than the ResNet 20 (2) based classifier. On the other hand, ResNet 20 (V2) provided a 16% higher F1 Score than AlexNet. It can be derived from the analysis that AlexNet is more suitable for real-time applications and ResNet is ideal for a high-performance applications. The research and analysis give an insight into how the performance of the deep learning model changes with the number of classes, number of images in the training set, depth of the classifier, and computational time in the context of SR and IHIL. The methodology also provides a suitable solution to deal with an imbalanced dataset. The analysis does not reflect any idea of power consumption, memory utilisation, etc., in this context. Moreover, there is scope to analyse how to increase the detection accuracy further.

The limitations of the methodology indicate the future research prospects in this field. Real-time plant species identification and health condition analysis for a large-scale agricultural farm is the need of the hour. The plants are prone to several diseases due to various factors such as environmental, genetic, inappropriate use of insecticides, etc. The scarcity of the necessary infrastructure in place to control such infections is also a constraint. A high-performance DL model with lower computation time, power consumption and memory requirements will meet the need of a real-time and resource-constrained system. Further research can propose a deep learning based model suitable for real-time plant species recognition and health identification.

Deep learning-based classification algorithms have self-learning capabilities, which may further be enhanced by the inclusion of primary datasets on a classifier trained using secondary datasets in the context of smart farming. Future smart farming systems both indoors and outdoors must be trained on a large dataset consisting of both primary and secondary data, collected in various controlled and uncontrolled environments to achieve complete automatization. There is also scope to analyse whether adding more augmented data will enhance the model's performance. Some of the research has reported better accuracy in SR and IHIL with larger and balanced datasets. Computer vision-based health condition detection and analysis of plant diseases and their control in real-time will help in increasing the yield of farmers in the near future.

The ResNet based deep learning algorithm can provide a better performance measure for plant phenotyping in the context of smart farming. The inclusion of multiple classes and evaluating the classifier's performance with multiclass evaluation parameters reduces false alarms to achieve robustness in predictions. The present work has the potential to provide an optimal solution for smart farming systems, which could be achieved by tuning and augmenting the dataset, designing various test-cases, balancing the depth of the classifier with that of training examples, trying various training cycles and employing multiclass evaluation parameters for performance assessment.

Author Contributions: Conceptualisation, A.J.H. and R.R.S.; methodology, A.J.H. and R.R.S.; software, A.J.H.; validation, A.J.H. and R.R.S.; formal analysis, A.J.H. and R.R.S.; investigation, A.J.H. and R.R.S.; writing—original draft preparation, A.J.H.; writing—review and editing, R.R.S.; visualisation, A.J.H. and R.R.S.; supervision, R.R.S.; project administration, R.R.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data repository is available at https://data.mendeley.com/datasets/hb74ynkjcn/1. Data was accessed on 7 July 2020.

Acknowledgments: We would like to thank C. S, Ramesh, Dean Research and Innovation, Presidency University, Itgalpur Raankunte, Yehalanka, Bengaluru, Karnataka 560064, India for all kinds of logistics and technical support.

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

References

1. FAO. *The Future of Food and Agriculture—Trends and Challenges*; Annual Report; FAO: Rome, Italy, 2017.
2. Costa, C.; Schurr, U.; Loreto, F.; Menesatti, P.; Carpentier, S. Plant Phenotyping Research Trends, a Science Mapping Approach. *Front. Plant Sci.* **2019**, *9*, 1933. [CrossRef] [PubMed]
3. Li, L.; Zhang, Q.; Huang, D. A Review of Imaging Techniques for Plant Phenotyping. *Sensors* **2014**, *14*, 20078–20111. [CrossRef]
4. Mahlein, A.-K. Plant Disease Detection by Imaging Sensors—Parallels and Specific Demands for Precision Agriculture and Plant Phenotyping. *Plant Dis.* **2016**, *100*, 241–251. [CrossRef]
5. Jin, X.; Zarco-Tejada, P.J.; Schmidhalter, U.; Reynolds, M.P.; Hawkesford, M.J.; Varshney, R.K.; Yang, T.; Nie, C.; Li, Z.; Ming, B.; et al. High-Throughput Estimation of Crop Traits: A Review of Ground and Aerial Phenotyping Platforms. *IEEE Geosci. Remote Sens. Mag.* **2021**, *9*, 200–231. [CrossRef]
6. Jiang, Y.; Li, C. Convolutional Neural Networks for Image-Based High-Throughput Plant Phenotyping: A Review. *Plant Phenomics* **2020**, *2020*, 1–22. [CrossRef]
7. Mishra, K.B.; Mishra, A.; Klem, K. Govindjee Plant phenotyping: A perspective. *Indian J. Plant Physiol.* **2016**, *21*, 514–527. [CrossRef]
8. Fiorani, F.; Schurr, U. Future Scenarios for Plant Phenotyping. *Annu. Rev. Plant Biol.* **2013**, *64*, 267–291. [CrossRef]
9. Kaur, S.; Pandey, S.; Goel, S. Plants Disease Identification and Classification through Leaf Images: A Survey. *Arch. Comput. Methods Eng.* **2018**, *26*, 507–530. [CrossRef]
10. Barré, P.; Stöver, B.C.; Müller, K.F.; Steinhage, V. LeafNet: A computer vision system for automatic plant species identification. *Ecol. Inform.* **2017**, *40*, 50–56. [CrossRef]
11. Owomugisha, G.; Mwebaze, E. Machine Learning for Plant Disease Incidence and Severity Measurements from Leaf Images. In Proceedings of the 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, USA, 18 December 2016; pp. 158–163.
12. Pawara, P.; Okafor, E.; Surinta, O.; Schomaker, L.; Wiering, M. Comparing Local Descriptors and Bags of Visual Words to Deep Convolutional Neural Networks for Plant Recognition. In Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods, Porto, Portugal, 24 February 2017; SCITEPRESS: Setúbal, Portugal, 2017; Volume 24, pp. 479–486.
13. Xiong, J.; Yu, D.; Liu, S.; Shu, L.; Wang, X.; Liu, Z. A Review of Plant Phenotypic Image Recognition Technology Based on Deep Learning. *Electronics* **2021**, *10*, 81. [CrossRef]
14. Lee, S.H.; Chan, C.S.; Mayo, S.J.; Remagnino, P. How deep learning extracts and learns leaf features for plant classification. *Pattern Recognit.* **2017**, *71*, 1–13. [CrossRef]
15. Zhang, S.; Zhang, C. Plant Species Recognition Based on Deep Convolutional Neural Networks. In Proceedings of the International Conference on Intelligent Computing, Liverpool, UK, 7 August 2017; Springer: Cham, Switzerland, 2017; pp. 282–289.
16. Wang, G.; Sun, Y.; Wang, J. Automatic Image-Based Plant Disease Severity Estimation Using Deep Learning. *Comput. Intell. Neurosci.* **2017**, *2017*, 1–8. [CrossRef]
17. Figueroa-Mata, G.; Mata-Montero, E. Using a convolutional siamese network for image-based plant species identification with small data-sets. *Biomimetics* **2020**, *5*, 8. [CrossRef] [PubMed]
18. Sun, Y.; Liu, Y.; Wang, G.; Zhang, H. Deep Learning for Plant Identification in Natural Environment. *Comput. Intell. Neurosci.* **2017**, *2017*, 1–6. [CrossRef]
19. Wei Tan, J.; Chang, S.W.; Abdul-Kareem, S.; Yap, H.J.; Yong, K.T. Deep learning for plant species classification using leaf vein morphometric. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2018**, *7*, 82–90.
20. Kumar, D.; Verma, C. Automatic Leaf Species Recognition Using Deep Neural Network. In *Evolving Technologies for Computing, Communication and Smart World*; Springer: Singapore, 2021; pp. 13–22.

21. Pereira, C.S.; Morais, R.; Reis, M.J.C.S. Deep Learning Techniques for Grape Plant Species Identification in Natural Images. *Sensors* **2019**, *19*, 4850. [CrossRef] [PubMed]
22. Ferentinos, K. Deep learning models for plant disease detection and diagnosis. *Comput. Electron. Agric.* **2018**, *145*, 311–318. [CrossRef]
23. Huang, S.; Liu, W.; Qi, F.; Yang, K. Development and Validation of a Deep Learning Algorithm for the Recognition of Plant Disease. In Proceedings of the 2019 IEEE 21st International Conference on High Performance Computing and Communications; IEEE 17th International Conference on Smart City; IEEE 5th International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Zhangjiajie, China, 10 August 2019; pp. 1951–1957.
24. Fuentes, A.; Yoon, S.; Kim, S.C.; Park, D.S. A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. *Sensors* **2017**, *17*, 2022. [CrossRef]
25. Too, E.C.; Yujian, L.; Njuki, S.; Yingchun, L. A comparative study of fine-tuning deep learning models for plant disease identifi-cation. *Comput. Electron. Agric.* **2019**, *161*, 272–279. [CrossRef]
26. Chouhan, S.S.; Singh, U.P.; Kaul, A.; Jain, S. A data repository of leaf images: Practice towards plant conservation with plant pathology. In Proceedings of the 2019 IEEE 4th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 21 November 2019; pp. 700–707.
27. Saleem, M.H.; Potgieter, J.; Arif, K.M. Plant Disease Detection and Classification by Deep Learning. *Plants* **2019**, *8*, 468. [CrossRef]
28. Canziani, A.; Paszke, A.; Culurciello, E. An Analysis of Deep Neural Network Models for Practical Applications. *arXiv* **2017**, arXiv:1605.07678.
29. 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.
30. Atienza, R. *Advanced Deep Learning with Keras: Apply Deep Learning Techniques, Autoencoders, GANs, Variational Autoen-Coders, Deep Reinforcement Learning, Policy Gradients, and More*; Packt Publishing Ltd.: Birmingham, UK, 2018.
31. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Commun. ACM* **2012**, *60*, 1097–1105. [CrossRef]
32. Chollet, F. Keras. 2015. Available online: https://github.com/fchollet/keras (accessed on 7 July 2020).
33. Bradski, G. The opencv library. Dr Dobb's J. *Softw. Tools* **2000**, *25*, 120–125.
34. Van der Walt, S.; Schönberger, J.L.; Nunez-Iglesias, J.; Boulogne, F.; Warner, J.D.; Yager, N.; Gouillart, E.; Yu, T. Scikit-image: Image processing in Python. *PeerJ* **2014**, *2*, e453. [CrossRef]
35. Krüger, F. Activity, context, and plan recognition with computational causal behavior models. Ph.D. thesis, Universiät Rostock, Fakultät für Informatik und Elektrotechnik, Rostock, Germany, 2016.
36. Sokolova, M.; Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. Manag.* **2009**, *45*, 427–437. [CrossRef]