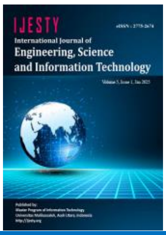
International Journal of Engineering, Science and Information Technology
Volume 5 No. 1 (2025) pp. 313-328
ISSN 2775-2674 (online)
Website: http://ijesty.org/index.php/ijesty
DOI: https://doi.org/10.52088/ijesty.v5i1.723
Research Paper, Short Communication, Review, Technical Paper



Herbal Plant Leaves Classification Using Convolutional Neural Network Models: A Literature Review

Alfharizky Fauzi[1], Emy Haryatmi[1]*, Tri Agus Riyadi[2], Murniyati[3]

[1]Department of Master of Electrical Engineering, Universitas Gunadarma, Indonesia
[2]Department of Informatics,Universitas Gunadarma, Indonesia
[3]Department of Informatics Management, Universitas Gunadarma, Indonesia

** Corresponding author Email: emy_h@staff.gunadarma.ac.id*

The manuscript was received on 10 May 2024, revised on 18 August 2024, and accepted on 3 January 2025, date of publication 14 January 2025

Abstract

Plants are essential to human beings because plants are considered most as foods. Plants can be used for food ingredients, medical purposes, and industrial applications. People inspect plants using traditional methods, such as using the naked eye, which can be time-consuming and expensive. Therefore, the effectiveness and high quality of automated crop identification classification systems are needed for adequate crop protection. This study aims to identify and classify nine plant species using different datasets, focusing on transfer learning from models trained on plant leaf datasets. Most research has shown that increasing the dataset size would significantly improve classification accuracy. The accuracy of the first test using the modified N1 classification model was 99.45%. In the second experiment, the accuracy of the N2 model was 99.65%. The accuracy of the N3 model, despite being slightly less accurate than AlexNet, was 99.55%, and it performed better, while the accuracy of AlexNet was 99.73%. Compared to the AlexNet model, the proposed model performed better and required less training time. The N1 model reduced the training time by 34.58%, the N2 model by 18.25%, and the N3 model by 20.23%. The N1 and N3 resulted in the same size, namely 14.8MB, and the compactness was 92.67%. The size of the N2 model was 29.7MB, and the compactness was 85.29% compactness. The proposed models provide more accuracy and efficiency in classifying plant leaves and can be used as a standalone mobile application that benefits farmers.

Keywords: *Plant Leaves, Classification, Detection, Transfer Learning, Convolutional Neural Network*.

1. Introduction

Plants are undeniably a vital food source for all living beings, thus earning the title of the backbone of ecosystems [1]. Plant species hold immense value as they can be utilized for medicinal purposes, food ingredients, and industrial applications. Some plant species are now endangered, underscoring the need for a comprehensive database for plant protection. The traditional method of manually inspecting plants with the naked eye is time-consuming and expensive [2], [3]. Continuous monitoring by experts in various agricultural fields follows this procedure. To optimize plant protection, accurate classification of plants is crucial [4]. Unlike seasonal flowers, plant leaves are accessible year-round and provide prominent features, making them ideal for automated plant classification. Leaves are instrumental in exploring genetic relationships and understanding plant development. However, given the many species, even botanists find plant identification challenging [5], [6]. Leaf recognition technology assists botanists in classifying specific plant species. Plants generally exhibit distinct characteristics such as texture, shape, color, and size, which differ among species [7]. In recent years, various "Computer-Aided Detection" (CAD) methods have been employed for leaf-based plant recognition due to their high classification accuracy [8], [9].
The interdisciplinary approach to plant classification integrates botanical data and species concepts with computational solutions [10]. Recent advancements in science and technology have enabled computer vision to assist botanists in identifying plants. Researchers in Computer vision have utilized leaves as a comparative tool for classifying plants [11]. From a machine-learning perspective, the classification problem can be solved by adopting innovative and rapid solutions and bringing together experts, decision-makers, farmers, and strategists [12].

Evolutionary neural networks have garnered significant attention from researchers in recent years due to their ability to provide superior image classification accuracy. These networks combine neural networks and computation to address a variety of problems. Krizhevsky et al. [13] set a record in the 2012 ImageNet Large Scale Visual Recognition Challenge by achieving 10.9% higher classification accuracy than the second-best entry. Advances in image processing have introduced various preprocessing techniques for image feature extraction. Feature extraction is the process of identifying discriminative characteristics that form the basis of classification. The classification task can be performed using several machine learning technologies, such as Support Vector Machines (SVM), Naïve Bayes, K-Nearest Neighbor (KNN), and Convolutional Neural Networks (CNN) [14].

Deep Learning is a subset of machine learning. It is a set of algorithms to model high-level data abstractions through deep graphs and multiple processing layers, including non-linear and linear transformations [15]. Convolutional Neural Networks (CNNs) are best used for image classification tasks due to the close relationship between layers and spatial information, which accounts for their recent popularity in plant classification.

Image-based assessment methods give more accurate and consistent results than human visual assessment from several studies [16]. Extensive work has been done to classify various items using different techniques. Lecun et al. [17] introduced the foundational deep learning tool CNN as an entry point to deep learning models in classification and detection. Deep learning models have been increasingly applied in agriculture in recent years. CNNs are dynamic models that significantly aid classification applications. Various CNN models used for classification include AlexNet, GoogLeNet [18], ResNet50, ResNet18, ResNet101[19], VGG16, VGG19 [20], DenseNet [21], and SqueezeNet [22], among others.

In [23], Mohanty et al. utilized AlexNet and GoogLeNet to classify 14 plant leaves. The accuracy was 99.27% and 99.34%, respectively. They experimented with different input data types, including color images, segmented images, and grayscale images separately. In [24], Dyrmann et al. implemented a CNN model to classify plant leaf data, and the accuracy was 86.2%. Barré et al. [25] classified plant leaves using the LeafSnap, Foliage, and Flavia datasets for different class classifications implemented in their proposed LeafNet model. The results of 184 LeafSnap classes gave an accuracy of 86.3%. The result of 60 foliage dataset classes gave an accuracy of 95.8%. The result of 32 Flavia dataset classes gave an accuracy of 97.9%.

A deep CNN model combined with a Multilayer Perceptron (MLP) classifier achieved 97.7% accuracy, and with an SVM classifier, the accuracy improved to 98.1% for the MalayaKew dataset, which contains 44 classes. Haque et al. [26] presented work on plant classification using geometric features during preprocessing, reaching 90% accuracy for classifying 10 plant species from the Flavia dataset. Previous studies achieved 84.2% accuracy in the LifeCLEF Plant Identification Task using their proposed Siam 3SN network, which learns spatial and structural features for leaf classification tasks. Further recognition of plant families and identification of plant classes for the four datasets were conducted using a two-way attention CNN model [27].

Identifying and classifying medicinal plants are crucial for the preparation of Ayurvedic medicine. Accurate classification is also essential for farmers, botanists, practitioners, forestry department officials, and those preparing Ayurvedic medicines. Using an AlexNet model, [28] the accuracy was 94.87% in classifying medicinal plants, while the accuracy of the Ayurleaf CNN model was 95.06%. The accuracy of previous studies using 20 self-collected medical plant species in the MobileNet model was 98.5%. In [29], a ten-layer CNN model was proposed by Liu et al. for classifying plant leaves into 32 categories. The accuracy of that model was 87.92%. The accuracy of the ResNet model for plant identification using the LeafSnap dataset was 93.09%. Using an Apple iPad, plant leaf classification was performed on images taken by Silva et al. [30]. The Deep Neural Network (DNN) model showed an accuracy of 91.17%, which improved to 95.58% using a CNN model. The classification accuracies were 91.5%, 92.4%, and 89.6% for the VGG16, VGG19, and Inception ResNetV2 models. For berry plant identification, the accuracy of the AlexNet model was 97.80% for three self-collected berry plant classes. A comparative analysis of work related to plant classification is presented in Table 1.

Table 1. Comparative analysis of research related to plant classification

Ref.	Obj	Data	Cls	Model	Accuracy
Dyrmann et al.	Plant leaf classification	Six different datasets	22	CNN	86.20%
Mohanty et al.	Identify 14 crop species.	Plant Village	38	AlexNet	99.27%
			38	Google Net	99.34%
Barré et al.	plant identification system	LeafSnap	184	LeafNet	86.30%
			60	LeafNet	95.80%
			32	LeafNet	97.90%
Lee et al.	Plant leaf classification	Malaya Kew	44	CNN MLP	97.70%
			44	Deep CNN SVM	98.10%
Gao et al.	Leaf Identification	Life CLEF 2015	30	3SN	84.20%
Dileep and Pournami.	Medicinal plant classification	AyurLeaf	40	AlexNet	94.87%
			40	Ayurleaf CNN	95.06%
Duong-Trung et al.	Medicinal plant classification	Own data	20	MobileNet	98.50%
Liu et al.	Classification of 32 different plant leaves	Flavia	32	Ten-layer CNN model	87.92%
Bodhwani et al.	Plant Identification	LeafSnap	180	ResNet	93.09%
Tiwari.	Plant leaf classification	Dataset collected by	30	DNN	91.17%
			30	CNN	95.58%
Yang et al.	Classification of plant leaf	Own data	15	VGG16	91.50%
			15	VGG19	92.40%
			15	Inception- ResNet V2	89.60%
Villaruz.	Identification of berry plants	Own data	3	AlexNet	97.80%

After completing plant classification, the work can be extended to disease classification. The accuracy of the VGG16 model, trained with transfer learning for apple leaf diseases, was 90.4%. Using an augmented dataset of 14,828 tomato leaf images, AlexNet reached an accuracy of 98.66%, while the VGG16 model achieved 98.18% [31]. The accuracy of a small CNN proposed by [32] to classify plants as healthy or diseased was 96.6%. The classification accuracy for tomato plant diseases using laboratory data was 98.50% for the VGG16 model, 98.30% for the VGG19 model, 99.40% for the ResNet model, and 99.60% for the Inception V3 model [33]. The model proposed by [34] outperformed both using AlexNet and VGG16. The accuracy in tomato plant leaf classification was 99.45% and 90.1%, respectively.

The accuracy of the Bagged Tree classifier, leveraging RGB, HSV, and LBP features for guava fruit disease classification, was 99% [35]. The accuracy of the detection of cassava plant diseases using a deep residual neural network was 96.75% [36]. Alli et al. [37] applied data augmentation techniques, achieving 99.7% accuracy in cassava plant disease classification using MobileNetV2. Data was autonomously collected from farms and processed using a Custom-Net model for pearl millet disease classification. The accuracy was 98.78% [38]. Table 2 provides a comparative overview of research efforts in plant disease classification.

Table 2. Comparative Analysis of Research Related to Plant Disease Classification

Ref.	Obj	Data	Model	Future Scope
Wang et al.	Apple black rot disease severity	PV	VGG16	More data at various disease stages can be used to improve accuracy.
Mohanty et al.	Identify 14 crop species and 26 diseases	PV	AlexNet, GoogLeNet	Image data from a smartphone can be supplemented with location and time information to improve accuracy.
Brahimi et al.	Tomato plant disease classification	PV	AlexNet, GoogLeNet	the size of deep models and the computation should be reduced.
Bharali et al.	Classifying into healthy and disease classes for different plant	Google images	CNN model	Larger datasets and more complex networks can be created to assess performance and improve accuracy.
Ahmad et al.	Classification of Tomato plant disease	Own data	VGG16, VGG19, ResNet Inception-V3	Optimize these models for better performance on real-world field-based data.
Anadhakrishnan et al.	Classification of Tomato plant disease	PV	AlexNet, VGG16, LeNet, ResNet, CNN model	computational time can be improved
Oyewola et al.	Cassava mosaic disease classification	Cassava Disease Dataset from Kaggle	Deep residual neural network	To improve accuracy, novel image augmentation methods are combined with other deep neural networks.
Alli et al.	Cassava disease recognition	Own data	MobileNetV2	Identifying a variety of plant diseases using multi-class detection
Almadhor et al.	Guava fruit disease detection	Own data	Bagged Tree classifier	To extract features automatically, employing deep learning methods
Kundu et al.	Pearl millet disease classification	Own data	Custom-Net	scope of making the predictions based on the parametric dataset collected by the data collector part

In this study, the CNN model proposed is utilized to classify plant species using the PlantVillage (PV) and Flavia datasets. A comparison is drawn with AlexNet through transfer learning. Despite its depth, the CNN model exhibits compact dimensions, requires less training time, and maintains high accuracy. This research introduces significant contributions:

1. Three exact and efficient models (N1, N2, and N3) are introduced for plant leaf classification. These models gave the best accuracy in classification while demanding reduced training durations.
2. The models' efficacy is validated by classifying outputs from the challenging PV and Flavia datasets, showcasing their high accuracy.
3. To demonstrate the versatility of the proposed models, the model was implemented to classify tomato leaf diseases using images captured from mobile phones. The accuracy achieved in disease classification underscores their suitability for plant and disease classification tasks.

2. Methods

This research uses a newly developed compact CNN model and AlexNet with transfer learning to explore plant classification. Nine classes, each representing distinct plant species images sourced from the PlantVillage (PV) database, were employed for classification. Additionally, 32 courses from the Flavia dataset underwent classification. Figure 1 depicts the workflow for the classification and validation procedures. The PV and Flavia datasets underwent separate augmentation, resizing images as necessary. The input image dimensions for the proposed model were 256 × 256 × 3, while for AlexNet, they were 227 × 227 × 3. The datasets were split into 80% for training and 20% for testing. The proposed model and AlexNet were trained using the training dataset to classify plant species. Subsequently, the trained models were validated using the testing data to predict classes for new data instances.

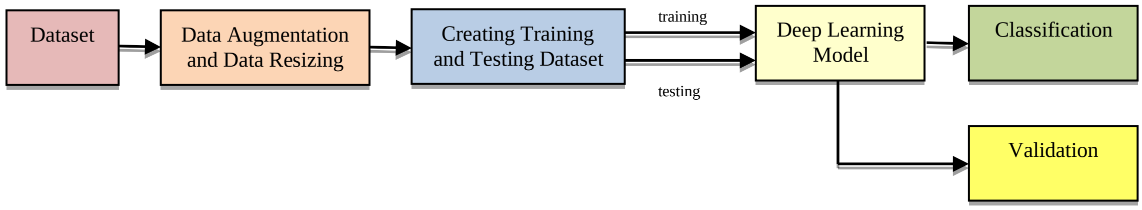


Fig 1. Workflow used in classification and validation

2.1. Plant Leaf Dataset

The PV dataset comprises images of healthy and diseased leaves across 38 distinct classes. This dataset includes healthy and diseased leaves from nine plant species: apple, cherry, corn, grape, peach, pepper, potato, strawberry, and tomato. The author considers these plant species for classification purposes. The Flavia dataset, consisting of 32 classes, was utilized for classification with the proposed model. The Flavia dataset encompasses leaf variant classes from various plant species such as "Anhui Barberry", "Beale's barberry", "Big-fruited Holly", "Camphortree", "Canadian Poplar", "Castor Aralia", "Cina Kayu Manis", "Berangan Kuda Cina", "Bunga Merah Cina", "Toon Cina", "Pohon Tulip Cina", "Crape Myrtle" atau "Crepe Myrtle", "Deodar", "Ford Woodlotus", "Ginkgo Maidenhair Tree", "Privet Mengkilap", "Pohon Hujan Emas", "Kayu Panah Jepang", "Kayu Keju Jepang", "Ceri Berbunga Jepang", "Maple Jepang", "Nanmu", "Oleander", "Persik", "Bambu Puber", "Selatan Magnolia", "Sweet Osmanthus", "Tangerine", "Trident Maple", "True Indigo", "Wintersweet", "Yew Plum Pine".

2.2. Preprocessing

Data preprocessing plays a critical role in ensuring algorithmic consistency and effectiveness. Deep learning models benefit from a diverse input dataset that avoids overfitting. Subtle adjustments, imperceptible to the human eye—like introducing noise and blur to input images—assist CNNs in capturing more complex features [39], [40]. In this study, the dataset undergoes augmentation, including Gaussian blur, salt and pepper noise with a random scale between 0.95 to 1.05 in horizontal and vertical directions, and random rotations ranging from -30° to 30° from the original image orientation. Detailed augmentation strategies are outlined in Table 3, encompassing rotations, flipping, color adjustments such as saturation, hue (representing color shades like blue, red, green, etc., ranging from 0 to 360 degrees), and contrast. Histogram equalization assesses and enhances contrast values in color augmentation, improving overall accuracy.

Table 3. Augmentation Data

No.	Augmentation	Methods
Augmentation 1	Noise	Salt and pepper noise
	Blur	Gaussian blur
	Position augmentation	Random scaling
		Random rotation
Augmentation 2	Position augmentation	45° rotation, 135° rotation, 225° rotation
		315° rotation, horizontal flip, vertical flip
	Color augmentation	Hue
		Saturation
		Contrast

The plant leaf data classification was conducted on a dataset comprising 38,400 images and an augmented dataset totaling 336,000 images. The deep learning framework utilized in this study includes proposed CNN model 1 (N1 model), CNN model 2 (N2 model), CNN model 3 (N3 model), and AlexNet model with transfer learning. Input images were resized to 256 × 256 × 3 for the developed and proposed models and resized to 227 × 227 × 3 for the AlexNet model.

2.3. Deep Learning Modelling

This analysis aims to create a computationally efficient and accurate classification of plant leaf learning models. The developed CNN model with three convolutional layers, as illustrated in Figure 2, is employed in this study. The model consists of three sets of 2D convolutional layers (Conv2D) followed by batch normalization and ReLU layers.

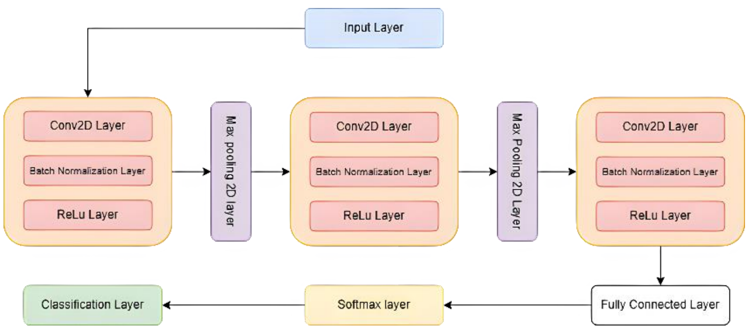


Fig 2. CNN models for classification and validation

Each model, N1, N2, and N3, incorporates three sets of Conv2D, batch normalization, and ReLU layers. The first two sets of Conv2D layers are followed by max-pooling layers, while the third set of Conv2D layers includes fully connected layers, softmax classifiers, and classification layers. The convolutional layers are customized with varying filter sizes and numbers across the CNN models N1, N2, and N3, as depicted in Table 4.

Table 4. Conv layers N1 model, N2 model, and N3 model

CNN Layer	N1 Model	N2 Model	N3 Model
1st Conv2D	3 × 3, 8	3 × 3, 16	7 × 7, 8
Max pooling	2	2	2
2nd Conv2D	3 × 3, 16	3 × 3, 32	5 × 5, 16
Max pooling	2	2	2
3rd Conv2D	3 × 3, 32	3 × 3, 64	3 × 3, 32

The convolutional layers employ a set of filters that convolve across the entire image. Each convolutional layer is tasked with learning various attributes that capture discriminative patterns to differentiate between plant leaf types in this architecture. Deep Neural Networks analyze different feature information from the preceding layers after each gradient update on the dataset. On the one hand, due to the parameters being updated in the previous layers during training, the distribution of input feature maps varies significantly. This condition has a significant effect on training speed and requires the use of various heuristics for parameter initialization. CNNs utilize Rectified Linear Unit (ReLU), an activation function designed specifically for neural networks. It operates as an identity function, f(x) = x, for all positive input 'x,' and zero for negative values. ReLU is rarely activated, mimicking biological neuron inactivity in response to certain stimuli. Max-pooling layers selectively activate a subset of neurons in the feature map. These are applied uniformly across all blocks in a '2x2' window with a stride factor of '2'. This effectively reduces the width and height of feature maps while maintaining a constant number of channels. Predicting multinomial probability distributions in CNN models, the softmax function is the output layer's activation function. In other words, softmax is employed as the classifier for multi-class classification.

Minimizing computational costs and shared weight propagation while producing lower backpropagation weights is one of the advantages of using smaller filter sizes compared to fully connected networks. Experts say the optimal choice remains 3×3 [41] [42]. The CNN N1 model has a consistent filter size 3×3 across all three Conv layers. The first Conv2D layer has eight filters, while the second and third Conv2D layers have 16 and 32 filters, respectively. In the CNN N2 model, the filter size remains the same as N1, but the number of filters is doubled compared to N1. For the CNN N3 model, the filter size for the first Conv2D layer is 7×7 with eight filters. The second Conv2D layer has a filter size 5×5 with 16 filters, and the third Conv2D layer is 3×3 with 32 filters. The first and second Conv2D layers are followed by max-pooling layers with a stride of 2 in a 2×2 window. The dataset is split into training and testing sets of 80-20% from 38,400 and 336,000 images. This data combination trains all CNN models for plant leaf classification. AlexNet, a pre-trained model capable of classifying up to 1000 classes, is also utilized. This study classifies plant leaves from the PV and Flavia datasets with 9 and 32 classes, respectively. For this purpose, AlexNet with transfer learning is implemented. Transfer learning aims to optimize Learning by leveraging the transferability of knowledge from its source.

2.4. CNN Model Performance Parameters

Deep learning model classification depends on their performance and accuracy. Performance parameters are assessed using the confusion matrix derived from the test dataset. Diagonal elements denote correct classifications, while non-diagonal elements indicate misclassifications within the confusion matrix. The components of the confusion matrix are detailed as follows [43]

1. True Positive (TP) refers to positive samples correctly labeled by the classifier.
2. True Negative (TN) refers to negative samples correctly labeled by the classifier.
3. False Positive (FP) refers to negative samples incorrectly labeled as positive.
4. False Negative (FN) refers to positive samples incorrectly labeled as unfavorable.

This evaluation focuses on key performance metrics such as macro recall, macro precision, macro F1 score, and average accuracy [31]. Sensitivity, or recall, measures the model's ability to correctly identify the positive class, also known as the actual positive rate. Precision indicates how accurately the model predicts positive instances. The F1 score, the harmonic mean of recall and precision, offers a balanced view of the model's performance. "Macro recall measures the classifier's ability to identify labels correctly across all classes." "Macro precision calculates the average consistency between actual data labels and the classifier's predictions for each class." "Macro F1 score represents the overall balance between precision and recall, averaged across all classes." "Accuracy is the proportion of correct predictions out of all predictions made."

$$Sensitivity/\,Recall = \frac{TP}{TP+FN} \tag{1}$$

Where C is the total of the class

$$Macro\ Recall = \frac{\sum_{n=1}^{C} Sensitivity}{C} \tag{2}$$

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$Macro\ Precision = \frac{\sum_{n=1}^{C} Precision}{C} \tag{4}$$

$$F1\ Score = \frac{2\ x\ Precision\ x\ Recall}{Precision+Recall} \tag{5}$$

$$Macro\ F1\ Score = \frac{\sum_{n=1}^{C} F1\ Score}{C} \tag{6}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{7}$$

The PV dataset comprises nine classes, resulting in a 9 × 9 confusion matrix. The confusion matrix for the Flavia dataset with 32 classes is 32 × 32. The accuracy of each class was evaluated for the N1, N2, N3, and AlexNet models. Additionally, the simulation time for each deep-learning model was recorded and measured in seconds.

2.5. Validation of The Trained CNN Model

Model validation was performed for the trained CNN models using images from the PlantVillage and Flavia datasets, which were not included in the training or testing sets. This validation utilized 33,600 images to classify unknown leaf images and evaluate the model's accuracy.

3. Result and Discussion

The PV dataset, which includes nine plant species, is depicted in Figure 3. The classes are abbreviated as follows: Apple with four varieties (A); Cherry with two varieties (Ch); Corn with four varieties (Co); Grape with four varieties (G); Peach with two varieties (Pch); Pepper with two varieties (Pep); Potato with three varieties (Po); Strawberry with two varieties (S); and Tomato with nine varieties (To).

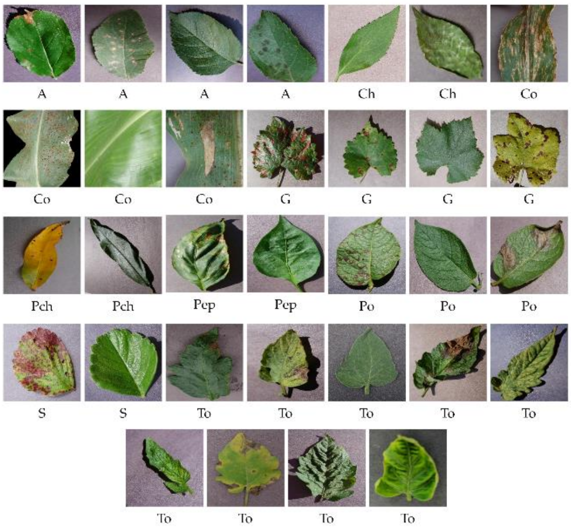


Fig 3. Dataset Train PlantVillage [29]

The Flavia dataset has 32 classes or types, as shown in Figure 4. These include "Anhui Barberry" (AB), "Beale's Barberry" (BB), "Big-Fruited Holly" (BFH), "Castor Aralia" (CA), "Camphortree" (Cam), "Chinese Cinnamon" (CC), "Chinese Horse Chestnut" (CHC), "Crape Myrtle" (CM), "Canadian Poplar" (CP), "Chinese Redbud" (CR), "Chinese Toon" (CT), "Chinese Tulip Tree" (CTT), "Deodar" (D), "Ford Woodlotus" (FW), "Ginkgo Maidenhair Tree" (GMT), "Glossy Privet" (GP), "Goldenrain Tree" (GT), "Japan Arrowwood" (JA), "Japanese Cheesewood" (JC), "Japanese Flowering Cherry" (JFC), "Japanese Maple" (JM), "Nanmu" (N), "Oleander" (O), "Peach" (P), "Pubescent Bamboo" (PB), "Southern Magnolia"(SM), "Sweet Osmanthus" (SO), "Tangerine" (T), "Trident Maple" (TM), "True Indigo" (TI), "Wintersweet" (W), and "Yew Plum Pine" (YPP). All 32 classes represent different species.

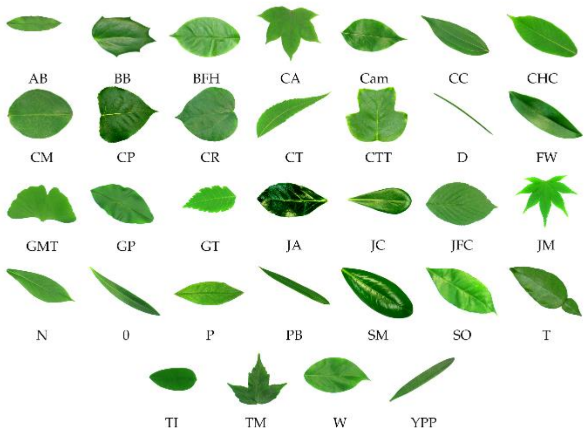


Fig 4. Dataset train Flavia [29]

The dataset preprocessing steps are detailed in Section 2.2. The dataset was enhanced with two augmented data types, augmented data 1 (ad1) and augmented data 2 (ad2). The images were subsequently resized to 256 × 256 × 3 for the proposed model and to 227 × 227 × 3 for AlexNet. Figure 5 illustrates examples of these augmented images.

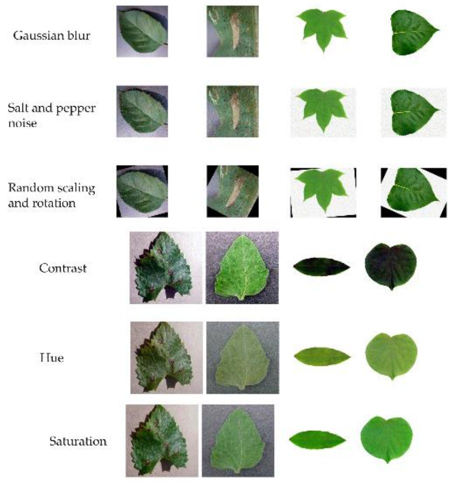


Fig 5. Preprocess dari dataset PV dan Flavia [29]

With transfer learning, plant leaf classification was performed using multiple models such as N1, N2, N3, and AlexNet. The classification accuracy of these models on the dataset, ad1 and ad2, is shown in Figure 6. Classification accuracy improved with dataset augmentation. The results show that more features were learned in ad2 because of the increased number of images used to train the models, which aided in achieving better prediction accuracy. The accuracy of the proposed N1 model was 86.58% with the dataset, which increased to 89.31% with ad1 and 99.45% with ad2. The recommended N2 model achieved an accuracy of 92.09% with the dataset, improving to 99.65% with ad2. The proposed N3 model showed an accuracy of 89.61%, which increased to 89.8% with ad1 and 99.55% with ad2. AlexNet demonstrated an accuracy of 98.53% with the dataset, rising to 99.73% with ad2. The accuracy of the N1, N2, N3 models and AlexNet was nearly identical for ad2. Training time increased with the number of images. The training times for models N1, N2, N3, and AlexNet are shown in Figure 7. The proposed models N1, N2, and N3 required less training time than AlexNet. The number of layers and filter sizes used in the proposed CNN models N1, N2, and N3 were fewer than those in the traditional AlexNet model. The developed models had three CNN layers, whereas AlexNet had five convolutional layers. Additionally, AlexNet used larger filter sizes than the proposed models. The compact design of our developed models, with fewer CNN layers and smaller filter sizes, reduced model complexity, resulting in shorter training times.

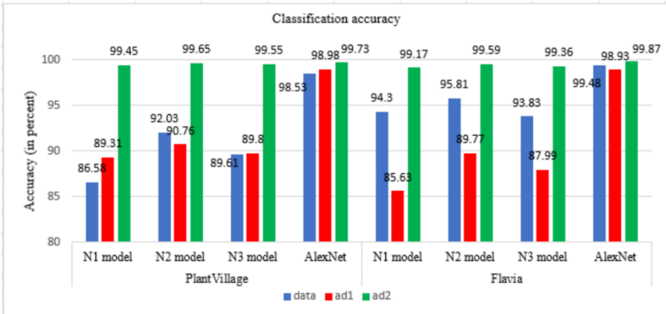


Fig 6. Model classification accuracy for datasets, ad1 and ad2 for PV and Flavia datasets [29]

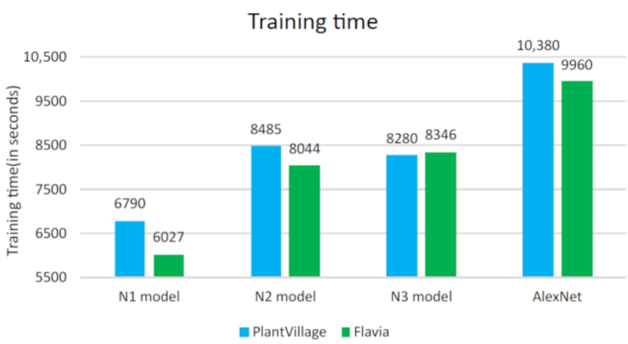


Fig 7. Model train time for PV and Flavia datasets [29]

Overfitting arises when a model performs well on training data but struggles to generalize to new, unseen data. To mitigate overfitting, strategies such as data augmentation, model simplification, dropout, regularization, and early stopping are implemented [44], [45]. In this study, the models were trained over two epochs with a learning rate set at 0.0001. The training accuracy and loss, as well as the validation accuracy and loss, are illustrated in Figure 8. Effective prevention of overfitting is indicated by models that show increasing training and validation accuracy alongside decreasing training and validation loss. Figure 8 displays the training accuracy and loss for (a) model N1, (c) model N2, (e) model N3, and (g) AlexNet. Validation accuracy and loss are depicted in Figure 8 for (b) model N1, (d) model N2, (f) model N3, and (h) AlexNet.

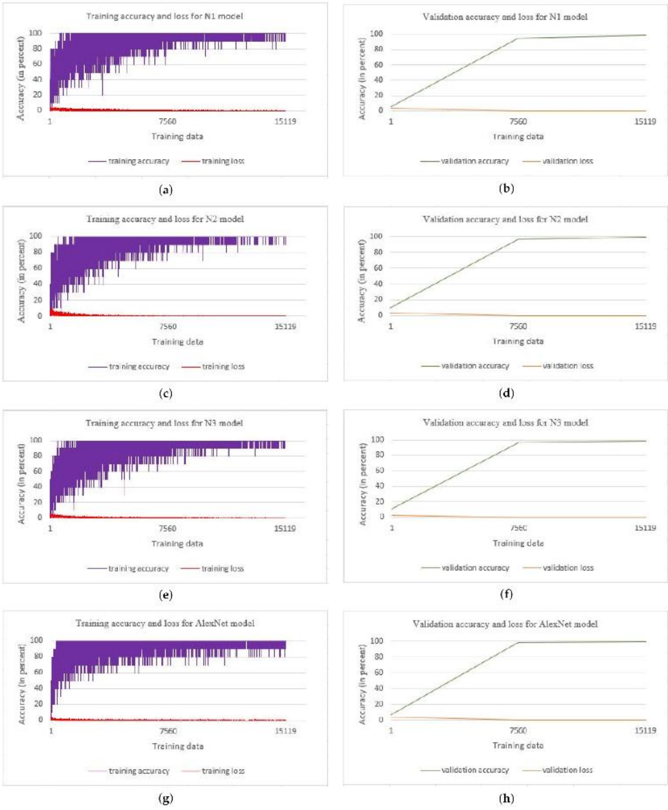


Fig 8. The accuracy of training and the loss of training along with validation accuracy and validation loss for model N1, model N2, model N3, and AlexNet [29]

A comparison of models in terms of accuracy and model size with existing benchmarks is presented in Table 5. Jeon and Rhee [46] achieved a notable 99.60% accuracy using Google. In their study on plant classification, Kaya et al. [47] utilized the PV and Flavia datasets employing AlexNet and VGG16 models. Wang and Wang [48] achieved 84.47% accuracy with VGG16 and improved to 92.64% with ResNet50 for plant classification. VGG16 and VGG19 achieved 81.3% and 96.25% accuracy, respectively [49]. Pruning and post-quantization techniques were applied to VGG16, AlexNet, and LeNet models [50] to reduce their size, resulting in performances of 91.49%, 96.59%, and 95.2%, respectively. A ten-layer CNN model by [29] achieved 87.92% accuracy with the Flavia dataset and 84.02% with the PV dataset. In our proposed models, N1 achieved 99.45% accuracy, N2 achieved 99.65%, N3 achieved 99.55%, and AlexNet with transfer learning reached 99.73% using the ad2 training set. The proposed model sizes for N1, N2, and N3 were 14.8 MB, 29.7 MB, and 14.8 MB, respectively, compared to AlexNet's 202 MB. Models N1 and N3 were 92.67% more compact than AlexNet, and N2 was 85.29% more compact while achieving similar accuracy ranges. Furthermore, training times for N1 and N2 were shorter compared to AlexNet. N1 required approximately 34.58% less training time, N2 about 18.25% less, and N3 20.23% less than AlexNet, demonstrating efficient training capabilities alongside superior performance metrics.

Table 5. Comparison of the performance of the proposed job with other existing jobs in the classification

Ref.	Data	Method	Accuracy	Size
Mohanty et al.	PV	AlexNet	99%	227 MB
		GoogLeNet	99%	27 MB
Lee et al.	Flavia	AlexNet	99.40%	202 MB
Jeon and Rhee	PV	GoogLeNet	99.60%	27 MB
Kaya et al.	Flavia	Alexnet	97.89%	202 MB
		VGG16	98.16%	515 MB
	PV	Alexnet	98.6%	202 MB
		VGG16	99.8%	515 MB
Wang	Flavia	VGG16	84.47%	515 MB
		ResNet50	92.24%	96 MB
Anubha Pearline et al.	Flavia	VGG16	95%	515 MB
		VGG19	96.25%	535 MB
Venkatesh et al.	PV	VGG16	81.3%	515 MB
		VGG16 +Inception	92.2%	-
Fountsop et al.	Flavia	VGG16 Pruning + post-quantization	91.49%	36.76 MB

		AlexNet Pruning + post-quantization	96.59%	32.37 MB
		LeNet Pruning + post-quantization	95.02%	9.91 MB
Liu et al.	Flavia	ten-layer CNN	87.92%	7 MB
Own work	PV	N1 model	99.45%	14.8 MB
		N2 model	99.65%	29.7 MB
		N3 model	99.55%	14.8 MB
		AlexNet	99.73%	202 MB
	Flavia	N1 model	99.17%	14.8 MB
		N1 model	99.59%	29.7 MB
		N1 model	99.36%	14.8 MB
		AlexNet	99.87%	202 MB

The classified output images for models N1, N2, N3, and AlexNet using transfer learning with 80% of the PV image dataset training data are depicted in Figure 9. These models underwent training using the original dataset, ad1 and ad2, and their respective outputs are presented here. Figure 9 illustrates the classified outputs for both the proposed models and AlexNet: (a) N1 model with the dataset, (b) N2 model with the dataset, (c) N3 model with the dataset, (d) AlexNet model with the dataset, (e) N1 model with ad1, (f) N2 model with ad1, (g) N3 model with ad1, (h) AlexNet model with ad1, (i) N1 model with ad2, (j) N2 model with ad2, (k) N3 model with ad2, and (l) AlexNet with ad2. The introductory part of the Results and Discussion section introduces the acronyms used to denote these classified output images associated with the PV dataset.

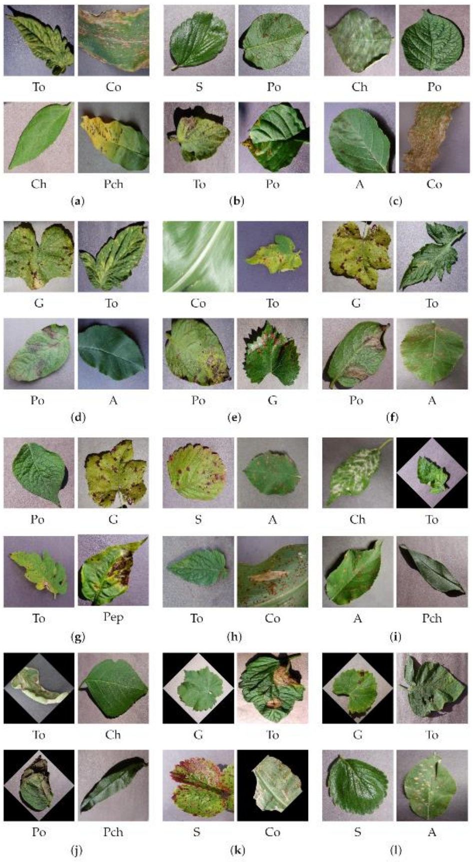


Fig 9. Classified output images for 80% of training data with PV dataset using (a) N1 model with the dataset, (b) N2 model with the dataset, (c) N3 model with the dataset, (d) AlexNet model with the dataset, (e) Model N1 with ad1, (f) Model N2 with ad1, (g) Model N3 with ad1, (h) Model AlexNet with ad1, (i) Model N1 with ad2, (j) Model N2 with ad2, (k) Model N3 with ad2, (l) AlexNet with ad2 [21]

The classified output images for the proposed models N1, N2, N3, and AlexNet using transfer learning with 80% of the Flavia image dataset training data are depicted in Figure 10. These models underwent training with the dataset, ad1 and ad2, and their respective outputs are presented here. Figure 10 illustrates the classified outputs for both the proposed models and AlexNet: (a) N1 model with the dataset, (b) N2 model with the dataset, (c) N3 model with the dataset, (d) AlexNet model with the dataset, (e) N1 model with ad1, (f) N2 model with ad1, (g) N3 model with ad1, (h) AlexNet model with ad1, (i) N1 model with ad2, (j) N2 model with ad2, (k) N3 model with ad2, and (l) AlexNet with ad2. The abbreviations used for the classified output images about the Flavia dataset are introduced at the beginning of the Results and Discussion section.

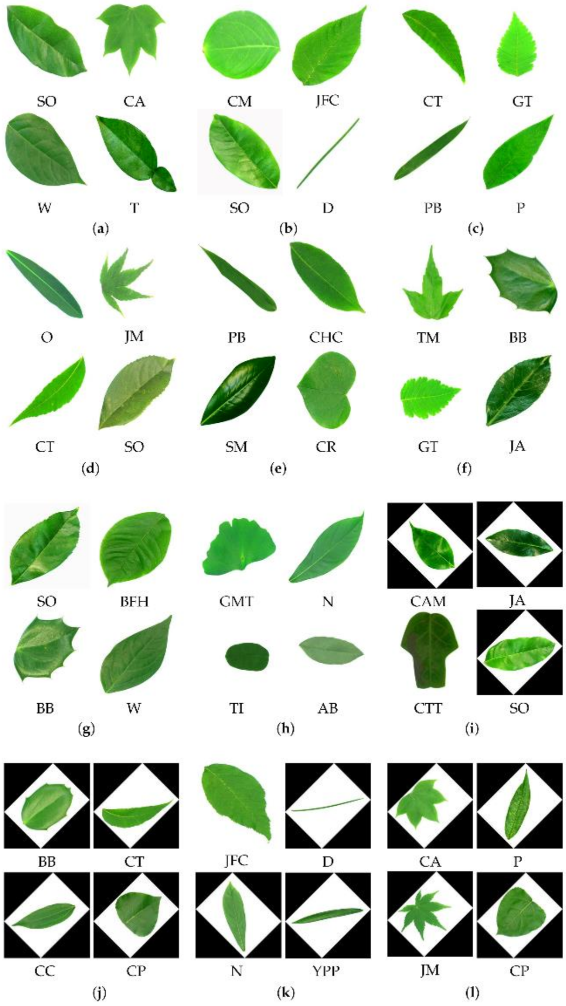


Fig 10. Classified output images for 80% of training data with Flavia dataset using (a) N1 model with the dataset, (b) N2 model with the dataset, (c) N3 model with the dataset, (d) AlexNet model with the dataset, (e) Model N1 with ad1, (f) Model N2 with ad1, (g) Model N3 with ad1, (h) Model AlexNet with ad1, (i) Model N1 with ad2, (j) Model N2 with ad2, (k) Model N3 with ad2, (l) AlexNet with ad2 [21]

Classification performance by models trained with dataset ad1 and ad2 was evaluated on the PV dataset using a confusion matrix as shown in Figure 11 for proposed models N1, N2, N3, and AlexNet. The confusion matrix provides insights into the classification and misclassification errors made by the models. Elements on the diagonal represent correct classifications, while off-diagonal elements represent misclassifications. Figure 11a illustrates the confusion matrix for the proposed N1 model trained on 80% of the dataset and tested on the remaining 20%. Diagonal elements indicate correct classifications for each class, highlighted in yellow-colored cells.

Actual Class (columns) / Predicted Class (rows)

Class	A	Ch	Co	G	Pch	Pep	Po	S	To
A	109	0	0	0	6	0	0	0	0
Ch	0	116	0	0	0	0	0	0	3
Co	0	0	115	2	0	0	3	0	0
G	0	0	0	119	1	0	0	0	0
Pch	1	0	0	0	100	0	0	0	0
Pep	0	0	0	0	0	118	0	0	0
Po	0	0	3	0	0	0	116	0	1
S	0	0	0	0	0	0	0	118	0
To	0	0	4	0	0	0	5	0	105

(a)

Class	N1		N2		N3		AlexNet	
	ad1	ad2	ad1	ad2	ad1	ad2	ad1	ad2
A	4.2	15.1	−10.4	3.7	−5.8	7	0	−0.1
Ch	7.9	14.7	−2.1	3	−4.2	3.6	0.8	0.7
Co	1.7	2.7	0.8	1	1	2	0	0
G	−1.2	2.8	−5	1.2	−0.4	2.8	0	0
Pch	5.8	12.2	2.5	8.2	1	8.9	1.3	1
Pep	10	24.3	11.3	21	3.8	18.1	−0.8	0.4
Po	5.4	11.1	−2	3.5	8.3	11	0	0
S	0	7	2.9	4	−0.8	2	0	−0.5
To	−4.6	4.7	2	6	−0.4	7	0	0.7

(b)

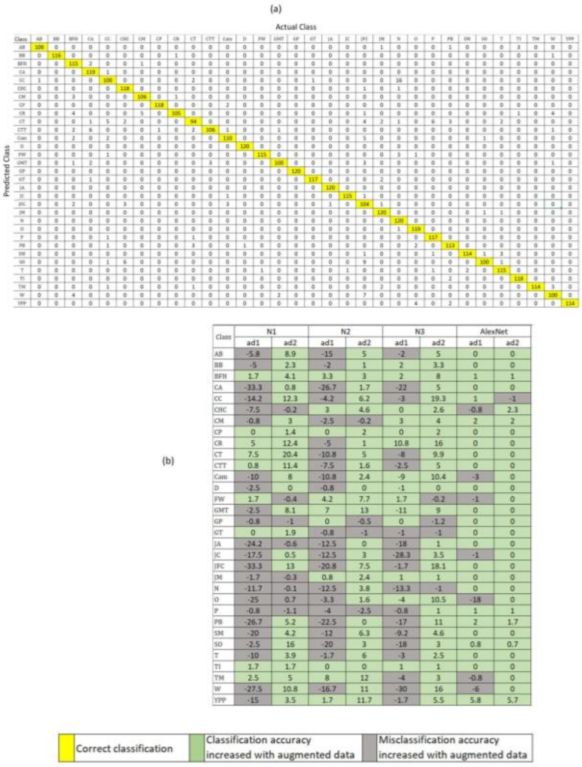
Correct classification | Classification accuracy increased with augmented data | Misclassification increased after augmented data

Fig 11. Confusion Matrix for Models created with PV Dataset, a. Confusion Matrix for N1 with 80% Training Data, b. Effect of Data Augmentation on Confusion Matrix with ad1 and ad2 for PV Dataset.

Figure 11b illustrates the influence of data augmentation on the confusion matrices of models N1, N2, N3, and AlexNet using ad1 and ad2. Negative values indicate a decline in classification performance, while positive values indicate an improvement. Cells shaded in green represent enhanced classification accuracy (percentage), whereas cells shaded in gray indicate the percentage of misclassifications following data augmentation.

It appears that the model performance is enhanced with training using augmented datasets. The proposed model N1's accuracy for the PV dataset is increased by 7.9% and 14.7% for the class "Ch" when trained with ad1 and ad2, respectively. The accuracy for the "Pep" class is improved by 10% and 24.3% for ad1 and ad2-trained models, respectively. Model N2's performance is enhanced by 11.3% and 21% for the "Pep" class with ad1 and ad2 training. The accuracy of model N2 is increased by 2% and 6% with ad1 and ad2 training, respectively, for the "To" class. Proposed model N3's accuracy is improved by 8.3% and 11% for the "Po" class with ad1 and ad2 training. The accuracy of the "Pch" class is enhanced by 8.9% with ad2. AlexNet's performance is increased by 1.3% for the "Pch" class when trained with ad1. In the case of AlexNet trained with ad2, the accuracy of two classes, "A" and "S," is reduced. The accuracy performance for proposed models N1, N2, and N3 is observed to be enhanced for each class with ad2 training.

The performance of models trained with datasets ad1 and ad2 is assessed using a confusion matrix on the Flavia dataset, presented in Figure 12 for the proposed model N1. Figure 12a displays the confusion matrix for model N1, trained with 80% of the dataset and tested with the remaining 20%. Correct classifications for each class are indicated by diagonal elements highlighted in yellow. Figure 12b illustrates the impact of data augmentation on the confusion matrix of models N1, N2, N3, and AlexNet with ad1 and ad2. Negative values denote a decrease in classification, while positive values indicate an increase. Cells shaded in green signify improved classification accuracy (percentage), whereas gray cells represent misclassification rates (percentage) post-data augmentation.



Class	N1		N2		N3		AlexNet	
	ad1	ad2	ad1	ad2	ad1	ad2	ad1	ad2
AB	-5.8	8.9	-15	5	-2	5	0	0
BB	-5	2.3	-2	1	2	3.3	0	0
BFH	1.7	4.1	3.3	3	2	8	1	1
CA	-33.3	0.8	-26.7	1.7	-22	5	0	0
CC	-14.2	12.3	-4.2	6.2	-3	19.3	1	-1
CHC	-7.5	-0.2	3	4.6	0	2.6	-0.8	2.3
CM	-0.8	3	-2.5	-0.2	3	4	2	2
CP	0	1.4	0	2	0	2	0	0
CR	5	12.4	-5	1	10.8	16	0	0
CT	7.5	20.4	-10.8	5	-8	9.9	0	0
CTT	0.8	11.4	-7.5	1.6	-2.5	5	0	0
Cam	-10	8	-10.8	2.4	-9	10.4	-3	0
D	-2.5	0	-0.8	0	-1	0	0	0
FW	1.7	-0.4	4.2	7.7	1.7	-0.2	-1	0
GMT	-2.5	8.1	7	13	-11	9	0	0
GP	-0.8	-1	0	-0.5	0	-1.2	0	0
GT	0	1.9	-0.8	-1	-1	-1	0	0
JA	-24.2	-0.6	-12.5	0	-18	1	0	0
JC	-17.5	0.5	-12.5	3	-28.3	3.5	-1	0
JFC	-33.3	13	-20.8	7.5	-1.7	18.1	0	0
JM	-1.7	-0.3	0.8	2.4	1	1	0	0
N	-11.7	-0.1	-12.5	3.8	-13.3	-1	0	0
O	-25	0.7	-3.3	1.6	-4	10.5	-18	0
P	-0.8	-1.1	-4	-2.5	-0.8	1	1	1
PB	-26.7	5.2	-22.5	0	-17	11	2	1.7
SM	-20	4.2	-12	6.3	-9.2	4.6	0	0
SO	-2.5	16	-20	3	-18	3	0.8	0.7
T	-10	3.9	-1.7	6	-3	2.5	0	0
TI	1.7	1.7	0	0	1	1	0	0
TM	2.5	5	8	12	-4	3	-0.8	0
W	-27.5	10.8	-16.7	11	-30	16	-6	0
YPP	-15	3.5	1.7	11.7	-1.7	5.5	5.8	5.7

Fig 12. Confusion Matrix for Models with Olivia Dataset, a. Confusion Matrix for N1 with 80% Training Data, b. Effect of Data Augmentation on Confusion Matrix with ad1 and ad2 for Flavia Dataset

The proposed model N1's accuracy for the Flavia dataset improved by 5% and 12.4% for the "CR" class when trained with ad1 and ad2, respectively, and by 10.8% and 16% for proposed model N2. The accuracy for the "CT" class improved by 7.5% with ad1 and 20.4% with ad2 for proposed model N1. Model N2's performance improved by 7% and 13% for the "GMT" class when trained with ad1 and ad2. The proposed model N3's accuracy increased by 19.3% for the "CC" class when trained with ad2. AlexNet's performance improved by 5.8% and 5.7% for the "YPP" class when trained with ad1 and ad2, respectively.

Data augmentation affects the average class precision [51], [52]. Based on the confusion matrix, macro recall, macro precision, macro F1 score, and average accuracy metrics are evaluated for the PV and Flavia datasets. The performance parameters of the recommended model N1, model N2, model N3, and AlexNet are shown in Table 6. Macro recall, macro precision, macro F1 score, and average accuracy metrics for the PV and Flavia datasets are compared here for data, ad1, and ad2. Performance parameters are enhanced with ad2. Developed models N1, N2, and N3 show comparable performance ranges to AlexNet. These models are significantly more compact than AlexNet and deliver good results.

Table 6. Performance parameters of N1 model, N2 model, N3 model, and AlexNet trained with data, ad1, and ad2

Data	Model	Data	Macro Recall	Macro Precision	Macro F1_score	Mean Accuracy
PV	N1	data	87.76%	86.58%	86.57%	99.16%
		ad1	89.50%	89.31%	89.26%	99.33%
		ad2	99.45%	99.45%	99.45%	99.97%
	N2	data	92.35%	92.03%	91.95%	99.50%
		ad1	91.54%	90.83%	90.83%	99.43%
		ad2	99.65%	99.65%	99.65%	99.98%
	N3	data	90.20%	89.61%	89.57%	99.35%
		ad1	90.32%	89.80%	89.71%	99.36%
		ad2	99.55%	99.55%	99.55%	99.97%
	AlexNet	data	98.60%	98.53%	98.52%	99.91%

		ad1	98.99%	98.98%	98.98%	99.94%
		ad2	99.74%	99.73%	99.73%	99.98%
Flavia	N1	data	94.54%	94.30%	94.30%	99.64%
		ad1	86.43%	85.63%	85.45%	99.10%
		ad2	99.18%	99.17%	99.17%	99.95%
	N2	data	96.05%	95.91%	95.91%	99.74%
		ad1	90.49%	89.77%	89.76%	99.36%
		ad2	99.59%	99.59%	99.59%	99.97%
	N3	data	93.99%	93.83%	93.80%	99.61%
		ad1	88.49%	87.99%	87.76%	99.25%
		ad2	99.37%	99.36%	99.36%	99.96%
	AlexNet	data	99.50%	99.48%	99.48%	99.97%
		ad1	99.02%	98.93%	98.93%	99.93%
		ad2	99.87%	99.87%	99.87%	99.99%

Analysis of variance (ANOVA) was developed by [53] to analyze the experimental design results using statistical tests. ANOVA was conducted on performance parameters for the recommended models, and AlexNet trained with datasets ad1 and ad2 from both datasets, as shown in Tables 7, 8, and 9. Evaluated parameters include the Sum of Squares (SS), degrees of freedom (df), Mean Square (MS), p-value, F-value, and critical F-value. The significance of statistical conditions is assessed based on the p-value, where a smaller F-value compared to the critical F-value indicates statistical significance. If the p-value ranges from 0.0001 to 0.001, it is highly statistically significant; from 0.001 to 0.01, it is very statistically substantial; from 0.01 to 0.05, it is statistically significant; and if the p-value is greater than 0.05, there is no statistical significance. Across the three ANOVA tables, we can observe the statistical significance of the evaluated models across both datasets.

Table 7. ANOVA analysis of performance parameters evaluated for dataset [53]

Source	Ss	Df	Ms	F	P-Value	F Critical	Sign in
Dataset	77.688	1	77.688	6.659	0.0164	4.259	**
Models	191.983	3	63.994	5.485	0.0051	3.009	***
Dataset × Models	24.687	3	8.229	0.705	0.5582	3.009	NS
Within	279.985	24	11.666				
Total	574.343	31					

Table 8. ANOVA of performance parameters evaluated for ad1 [53]

Source	Ss	Df	Ms	F	P-Value	F Critical	Sign in
Dataset	12.276	1	12.276	0.604	0.4446	4.259	NS
Models	366.819	3	122.273	6.015	0.0033	3.009	***
Dataset × Models	7.659	3	2.553	0.125	0.944	3.009	NS
Within	487.848	24	20.327				
Total	874.602	31					

Table 9. ANOVA of performance parameters evaluated for ad2 [53]

Source	Ss	Df	Ms	F	P-Value	F Critical	Sign in
Dataset	4.43×10^{-6}	1	4.43×10^{-6}	0.816	0.3752	4.259	NS
Models	6.23×10^{-5}	3	2.08×10^{-5}	3.829	0.0225	3.009	*
Dataset × Models	1.13×10^{-5}	3	3.77×10^{-6}	0.696	0.5634	3.009	NS
Within	0.0001	24	5.42×10^{-6}				
Total	0.0002	31					

The ability of trained models to classify new data is crucial in decision-making. The PV dataset comprises nine classes representing nine plant species. Species classification for PV dataset images not included in the training and testing datasets has been completed. Validation accuracies of models N1, N2, N3, and AlexNet trained on the PV dataset are presented in Table 10. Images not part of the training and testing datasets were used for validation across each type. Pepper validation performance was lower compared to other species. The proposed model N2 classified pepper species with 92.5% accuracy, while model N2 and AlexNet classified apple species with 95% accuracy. Validation accuracy for corn was 97.5% for both model N2 and AlexNet. Grape, peach, and strawberry validation were excellent across all models. Model N3 achieved 91.11% validation accuracy for tomatoes. Overall, model N2 outperformed model N1 and N3 in performance.

Table 10. Validation accuracy of the proposed N1 model, N2 model, N3 model, and AlexNet model trained with PV dataset [31]

Species	N1	N2	N3	AlexNet
Apple	82.5%	92.5%	90%	100%
Cherry	75%	95%	85%	95%
Corn	95%	97.5%	90%	97.5%
Grape	90%	97.5%	95%	100%
Peach	80%	85%	90%	100%

Pepper	40%	60%	70%	100%
Potato	73.33%	90%	86.67%	100%
Strawberry	100%	90%	100%	100%
Tomato	88.89%	84.44%	91.11%	95.56%

Species classification for images in the Flavia dataset that were not part of the training and testing datasets has been completed. The Flavia dataset includes 32 classes representing different plant species. Validation accuracies of the proposed model N1, model N2, model N3, and AlexNet trained on the Flavia dataset are shown in Table 11. Nearly all species showed good classification except for the Cam class. Overall, model N2 performed equally well as AlexNet. Model N2 achieved superior classification performance and validation for both PV and Flavia datasets with compact model sizes.

Table 11. Validation accuracy of the N1 model, N2 model, N3 model, and AlexNet model trained with the Flavia dataset [31]

Species	N1	N2	N3	AlexNet
AB	100%	60%	80%	100%
BB	100%	100%	100%	100%
BFH	100%	100%	100%	100%
CA	100%	100%	100%	100%
Cam	10%	10%	10%	50%
CC	60%	50%	50%	100%
CHC	80%	90%	90%	100%
CM	100%	100%	100%	100%
CP	100%	100%	100%	100%
CR	90%	80%	90%	100%
CT	30%	30%	20%	80%
CTT	100%	100%	100%	100%
D	90%	100%	90%	100%
FW	90%	90%	90%	100%
GMT	100%	100%	100%	100%
GP	100%	100%	100%	100%
GT	80%	50%	70%	100%
JA	100%	100%	100%	100%
JC	50%	50%	60%	80%
JFC	50%	60%	60%	100%
JM	100%	100%	100%	100%
N	90%	90%	90%	100%
O	100%	100%	100%	100%
P	100%	100%	100%	100%
PB	50%	70%	60%	90%
SM	40%	70%	40%	100%
SO	70%	60%	70%	90%
T	70%	90%	90%	100%
TI	90%	100%	90%	100%
TM	90%	80%	90%	100%
W	90%	90%	90%	100%
YPP	30%	90%	20%	70%

The proposed model demonstrates strong performance in classifying plant leaves. The subsequent model will focus on diagnosing diseases affecting tomato plant leaves. Image data was sourced from tomato farms in Lavale, Pune, India, captured using a Super Speed Dual Pixel 12MP AF sensor smartphone camera, with the background around tomato plant leaves. In practical scenarios, input images may vary in quality. Each of the four classes comprised 300 images, totaling 126,000 images. Augmented images were utilized to train both the proposed model and AlexNet. The dataset includes various tomato plant diseases, including tomato leaf blight, leaf miner, yellow leaf curl virus (YLCV), and healthy leaves. Figure 13 displays images of these diseases affecting tomato plants and the healthy class dataset from Lavale Agriculture.

Fig 13. Image of Plant Leaves from Lavale [5]

The classification accuracy of the recommended models and AlexNet is shown in Figure 14. The proposed N1 model achieved an accuracy of 99.864%, N2 achieved 99.59%, N3 achieved 99.63%, and AlexNet achieved an accuracy of 99.35%. Model N1 required 22.82% training time, N2 required 56.18%, and N3 required 25.73% less training time than AlexNet. N1 and N3 models were 99.06% compact, while N2 was 98.11% compact compared to AlexNet. The performance of the recommended models demonstrates their capability to classify images of tomato plant diseases with complex backgrounds with very high accuracy, shorter training times, and compact model sizes. These recommended models are highly compact and require less training than advanced models. They can be deployed as standalone mobile applications that benefit farmers with their results.

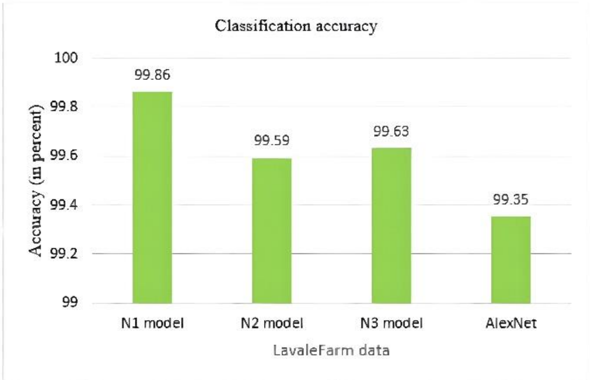


Fig 14. Model classification accuracy graph for the Lavale agricultural dataset [5]

4. Conclusion

Classification of plant leaf images from the nine-class PV database was performed using the recommended CNN models: N1, N2, N3, and AlexNet with transfer learning in this study. The developed models showed improved performance after data augmentation was applied. The achieved accuracies for the developed models were 99.45% for N1, 99.65% for N2, 99.55% for N3, and 99.73% for AlexNet on the PV dataset. The accuracy of the Flavia dataset with 32 classes was 99.17% for N1, 99.59% for N2, 99.36% for N3, and 99.87% for AlexNet. The developed models achieved accuracies comparable to AlexNet. Proposed models N1 and N3 were 92.67% more compact than AlexNet, and N2 was 85.3% more compact. Training time for the developed models was reduced by 34.58% for N1, 18.25% for N2, and 20.23% for N3 compared to AlexNet. N2 showed a compact size compared to AlexNet while maintaining similar range accuracy. These trained models classified species on images from the PV and Flavia datasets that were not included in model training and testing. Overall, the N2 model's performance surpassed that of the N1 and N3 models. Experiments on these challenging datasets, PV and Flavia, confirmed the effectiveness of our method. The uniqueness of the recommended models lies in classifying diseased plant leaves in images taken with mobile phones. The proposed model can be a standalone mobile application benefiting farmers due to its compact size and excellent classification results. Automated plant classification will aid in plant management, benefiting society.

References

[1] Z.-Y. Gao, H.-X. Xie, J.-F. Li, and S.-L. Liu, "Spatial-Structure Siamese Network for Plant Identification," *Intern J Pattern Recognit Artif Intell*, vol. 32, no. 11, p. 1850035, Nov. 2018, doi: 10.1142/S0218001418500350.

[2] S. Kaur, S. Pandey, and S. Goel, "Plants Disease Identification and Classification Through Leaf Images: A Survey," *Archives of Computational Methods in Engineering*, vol. 26, no. 2, pp. 507–530, Apr. 2019, doi: 10.1007/s11831-018-9255-6.

[3] A. Rehman, S. Naz, M. I. Razzak, F. Akram, and M. Imran, "A Deep Learning-Based Framework for Automatic Brain Tumors Classification Using Transfer Learning," *Circuits Syst Signal Process*, vol. 39, no. 2, pp. 757–775, Feb. 2020, doi: 10.1007/s00034-019-01246-3.

[4] S. G. Wu, F. S. Bao, E. Y. Xu, Y.-X. Wang, Y.-F. Chang, and Q.-L. Xiang, "A Leaf Recognition Algorithm for Plant Classification Using Probabilistic Neural Network," in *2019 IEEE International Symposium on Signal Processing and Information Technology*, Giza, Egypt: IEEE, Dec. 2019, pp. 11–16. doi: 10.1109/ISSPIT.2007.4458016.

[5] S. Tiwari, "A Comparative Study of Deep Learning Models With Handcraft Features and Non-Handcraft Features for Automatic Plant Species Identification," *International Journal of Agricultural and Environmental Information Systems*, vol. 11, no. 2, pp. 44–57, Apr. 2020, doi: 10.4018/IJAEIS.2020040104.

[6] K. Yang, W. Zhong, and F. Li, "Leaf Segmentation and Classification with a Complicated Background Using Deep Learning," *Agronomy*, vol. 10, no. 11, p. 1721, Nov. 2020, doi: 10.3390/agronomy10111721.

[7] C. A. Priya, T. Balasaravanan, and A. S. Thanamani, "An efficient leaf recognition algorithm for plant classification using support vector machine," in *International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME-2012)*, Salem, Tamilnadu, India: IEEE, Mar. 2012, pp. 428–432. doi: 10.1109/ICPRIME.2012.6208384.

[8] J. Du, C.-M. Zhai, and Q.-P. Wang, "Recognition of plant leaf image based on fractal dimension features," *Neurocomputing*, vol. 116, pp. 150–156, Sep. 2019, doi: 10.1016/j.neucom.2012.03.028.

[9] V. Bodhwani, D. P. Acharjya, and U. Bodhwani, “Deep Residual Networks for Plant Identification,” *Procedia Comput Sci*, vol. 152, pp. 186–194, 2019, doi: 10.1016/j.procs.2019.05.042.

[10] S. H. Lee, C. S. Chan, S. J. Mayo, and P. Remagnino, "How deep learning extracts and learns leaf features for plant classification," *Pattern Recognit*, vol. 71, pp. 1–13, Nov. 2017, doi: 10.1016/j.patcog.2017.05.015.

[11] N. Kumar *et al.*, "Leafsnap: A Computer Vision System for Automatic Plant Species Identification," in *Computer Vision – ECCV 2018*, vol. 7573, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2018, pp. 502–516. doi: 10.1007/978-3-642-33709-3_36.

[12] N. Duong-Trung, L.-D. Quach, and C.-N. Nguyen, "Learning Deep Transferability for Several Agricultural Classification Problems," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 1, 2019, doi: 10.14569/IJACSA.2019.0100107.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun ACM*, vol. 60, no. 6, pp. 84–90, May 2017, doi: 10.1145/3065386.

[14] J. A. Villaruz, "Deep Convolutional Neural Network Feature Extraction for Berry Trees Classification," *Journal of Advances in Information Technology*, vol. 12, no. 3, 2021, doi: 10.12720/jait.12.3.226-233.

[15] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep Learning for Computer Vision: A Brief Review," *Comput Intell Neurosci*, vol. 2018, pp. 1–13, 2018, doi: 10.1155/2018/7068349.

[16] G. Wang, Y. Sun, and J. Wang, "Automatic Image-Based Plant Disease Severity Estimation Using Deep Learning," *Comput Intell Neurosci*, vol. 2017, pp. 1–8, 2017, doi: 10.1155/2017/2917536.

[17] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 2020, doi: 10.1109/5.726791.

[18] C. Szegedy *et al.*, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, MA, USA: IEEE, Jun. 2018, pp. 1–9. doi: 10.1109/CVPR.2015.7298594.

[19] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2019, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[20] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," 2018, doi: 10.48550/ARXIV.1409.1556.

[21] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," 2019, doi: 10.48550/ARXIV.1608.06993.

[22] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2019, doi: 10.48550/ARXIV.1602.07360.

[23] S. P. Mohanty, D. P. Hughes, and M. Salathé, "Using Deep Learning for Image-Based Plant Disease Detection," *Front Plant Sci*, vol. 7, p. 1419, Sep. 2019, doi: 10.3389/fpls.2016.01419.

[24] M. Dyrmann, H. Karstoft, and H. S. Midtiby, "Plant species classification using deep convolutional neural network," *Biosyst Eng*, vol. 151, pp. 72–80, Nov. 2018, doi: 10.1016/j.biosystemseng.2016.08.024.

[25] P. Barré, B. C. Stöver, K. F. Müller, and V. Steinhage, "LeafNet: A computer vision system for automatic plant species identification," *Ecol Inform*, vol. 40, pp. 50–56, Jul. 2017, doi: 10.1016/j.ecoinf.2017.05.005.

[26] F. Haque and S. Haque, "PLANT RECOGNITION SYSTEM USING LEAF SHAPE FEATURES AND MINIMUM EUCLIDEAN DISTANCE," *ICTACT Journal on Image and Video Processing*, vol. 9, no. 2, pp. 1919–1925, Nov. 2018, doi: 10.21917/ijivp.2018.0272.

[27] Y. Zhu *et al.*, "TA-CNN: Two-way attention models in deep convolutional neural network for plant recognition," *Neurocomputing*, vol. 365, pp. 191–200, Nov. 2019, doi: 10.1016/j.neucom.2019.07.016.

[28] M. R. Dileep and P. N. Pournami, "AyurLeaf: A Deep Learning Approach for Classification of Medicinal Plants," in *TENCON 2019 - 2019 IEEE Region 10 Conference (TENCON)*, Kochi, India: IEEE, Oct. 2019, pp. 321–325. doi: 10.1109/TENCON.2019.8929394.

[29] J. Liu, S. Yang, Y. Cheng, and Z. Song, "Plant Leaf Classification Based on Deep Learning," in *2018 Chinese Automation Congress (CAC)*, Xi'an, China: IEEE, Nov. 2018, pp. 3165–3169. doi: 10.1109/CAC.2018.8623427.

[30] P. F. B. Silva, A. R. S. Marçal, and R. M. A. da Silva, “Evaluation of Features for Leaf Discrimination,” 2018, pp. 197–204. doi: 10.1007/978-3-642-39094-4_23.

[31] M. Brahimi, K. Boukhalfa, and A. Moussaoui, "Deep Learning for Tomato Diseases: Classification and Symptoms Visualization," *Applied Artificial Intelligence*, vol. 31, no. 4, pp. 299–315, Apr. 2017, doi: 10.1080/08839514.2017.1315516.

[32] P. Bharali, C. Bhuyan, and A. Boruah, "Plant Disease Detection by Leaf Image Classification Using Convolutional Neural Network," in *Information, Communication and Computing Technology*, vol. 1025, A. Bin Gani, P. K. Das, L. Kharb, and D. Chahal, Eds., Singapore: Springer Singapore, 2019, pp. 194–205. doi: 10.1007/978-981-15-1384-8_16.

[33] I. Ahmad, M. Hamid, S. Yousaf, S. T. Shah, and M. O. Ahmad, "Optimizing Pretrained Convolutional Neural Networks for Tomato Leaf Disease Detection," *Complexity*, vol. 2020, pp. 1–6, Sep. 2020, doi: 10.1155/2020/8812019.

[34] A. T and J. S. M. Murugaiyan, "Identification of Tomato Leaf Disease Detection using Pretrained Deep Convolutional Neural Network Models," *Scalable Computing: Practice and Experience*, vol. 21, no. 4, pp. 625–635, Dec. 2020, doi: 10.12694/scpe.v21i4.1780.

[35] A. Almadhor, H. T. Rauf, M. I. U. Lali, R. Damaševičius, B. Alouffi, and A. Alharbi, "AI-Driven Framework for Recognition of Guava Plant Diseases through Machine Learning from DSLR Camera Sensor Based High Resolution Imagery," *Sensors*, vol. 21, no. 11, p. 3830, Jun. 2021, doi: 10.3390/s21113830.

[36] D. O. Oyewola, E. G. Dada, S. Misra, and R. Damaševičius, "Detecting cassava mosaic disease using a deep residual convolutional neural network with distinct block processing," *PeerJ Comput Sci*, vol. 7, p. e352, Mar. 2021, doi: 10.7717/peerj-cs.352.

[37] O. O. Abayomi-Alli, R. Damaševičius, S. Misra, and R. Maskeliūnas, "Cassava disease recognition from <scp>low-quality</scp> images using enhanced data augmentation model and deep learning," *Expert Syst*, vol. 38, no. 7, Nov. 2021, doi: 10.1111/exsy.12746.

[38] N. Kundu *et al.*, "IoT and Interpretable Machine Learning Based Framework for Disease Prediction in Pearl Millet," *Sensors*, vol. 21, no. 16, p. 5386, Aug. 2021, doi: 10.3390/s21165386.

[39] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *J Big Data*, vol. 6, no. 1, p. 60, Dec. 2019, doi: 10.1186/s40537-019-0197-0.

[40] S. Cygert and A. Czyzewski, "Toward Robust Pedestrian Detection With Data Augmentation," *IEEE Access*, vol. 8, pp. 136674–136683, 2020, doi: 10.1109/ACCESS.2020.3011356.

[41] M. Längkvist, J. Jendeberg, P. Thunberg, A. Loutfi, and M. Lidén, "Computer aided detection of ureteral stones in thin slice computed tomography volumes using Convolutional Neural Networks," *Comput Biol Med*, vol. 97, pp. 153–160, Jun. 2018, doi: 10.1016/j.compbiomed.2018.04.021.
[42] S. Jaiswal and G. C. Nandi, "Robust real-time emotion detection system using CNN architecture," *Neural Comput Appl*, vol. 32, no. 15, pp. 11253–11262, Aug. 2020, doi: 10.1007/s00521-019-04564-4.
[43] H. F. Nweke, Y. W. Teh, M. A. Al-garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," *Expert Syst Appl*, vol. 105, pp. 233–261, Sep. 2019, doi: 10.1016/j.eswa.2018.03.056.
[44] R. Moradi, R. Berangi, and B. Minaei, "A survey of regularization strategies for deep models," *Artif Intell Rev*, vol. 53, no. 6, pp. 3947–3986, Aug. 2020, doi: 10.1007/s10462-019-09784-7.
[45] Srivastava, N. and Hinton, G. and Krizhevsky, A. and Sutskever, I. and Salakhutdinov, and Ruslan, "a simple way to prevent neural networks from overfitting," vol. 15, pp. 1929–1958, 2019, [Online]. Available: JMLR.
[46] W.-S. Jeon and S.-Y. Rhee, "Plant Leaf Recognition Using a Convolution Neural Network," *The International Journal of Fuzzy Logic and Intelligent Systems*, vol. 17, no. 1, pp. 26–34, Mar. 2017, doi: 10.5391/IJFIS.2017.17.1.26.
[47] A. Kaya, A. S. Keceli, C. Catal, H. Y. Yalic, H. Temucin, and B. Tekinerdogan, "Analysis of transfer learning for deep neural network based plant classification models," *Comput Electron Agric*, vol. 158, pp. 20–29, Mar. 2019, doi: 10.1016/j.compag.2019.01.041.
[48] B. Wang and D. Wang, "Plant Leaves Classification: A Few-Shot Learning Method Based on Siamese Network," *IEEE Access*, vol. 7, pp. 151754–151763, 2019, doi: 10.1109/ACCESS.2019.2947510.
[49] S. Anubha Pearline, V. Sathiesh Kumar, and S. Harini, "A study on plant recognition using conventional image processing and deep learning approaches," *Journal of Intelligent & Fuzzy Systems*, vol. 36, no. 3, pp. 1997–2004, Mar. 2019, doi: 10.3233/JIFS-169911.
[50] A. N. Fountsop, J. L. Ebongue Kedieng Fendji, and M. Atemkeng, "Deep Learning Models Compression for Agricultural Plants," *Applied Sciences*, vol. 10, no. 19, p. 6866, Sep. 2020, doi: 10.3390/app10196866.
[51] A. Fuentes, S. Yoon, S. Kim, and D. Park, "A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition," *Sensors*, vol. 17, no. 9, p. 2022, Sep. 2017, doi: 10.3390/s17092022.
[52] A. F. Fuentes, S. Yoon, J. Lee, and D. S. Park, "High-Performance Deep Neural Network-Based Tomato Plant Diseases and Pests Diagnosis System With Refinement Filter Bank," *Front Plant Sci*, vol. 9, p. 1162, Aug. 2018, doi: 10.3389/fpls.2018.01162.
[53] B. De Agustina, E. Rubio, and M. Sebastián, "Surface Roughness Model Based on Force Sensors for the Prediction of the Tool Wear," *Sensors*, vol. 14, no. 4, pp. 6393–6408, Apr. 2018, doi: 10.3390/s140406393.