Journal of Engineering and Technology for Industrial Applications

ISSN ONLINE: 2447-0228

ITEGAM-JETIA

Manaus, v.11 n.51, p. 206-212. January/February., 2025.
DOI: https://doi.org/10.5935/jetia.v11i51.1464

RESEARCH ARTICLE OPEN ACCESS

DEEP TRANSFER LEARNING FOR AUTOMATIC PLANT SPECIES RECOGNITION

Abdelwhab OUAHAB[1], Lazreg Taibaoui[2] and Boubakeur Zegnini[3]

[1,2,3] Department of Electrical Engineering, Laboratoire d'Etudes et Développement des Matériaux Semi-Conducteurs et Diélectriques, Amar Telidji University of Laghouat, BP 37 G, Route de Ghardaïa, Laghouat03000, Algeria.

[1]http://orcid.org/0000-0003-0648-2947, [2]https://orcid.org/0000-0002-1598-4546, [3]https://orcid.org/0000-0003-0937-188X

Email: ouahab.abdelwhab@univ-adrar.edu.dz, l.taibaoui@lagh-univ.dz, b.zegnini@lagh-univ.dz.

ARTICLE INFO

Article History
Received: December 11, 2024
Revised: January 20, 2025
Accepted: January 25, 2025
Published: February 28, 2025

Keywords:
Deep learning,
transfer learning,
plant species recognition,
convolutional neural networks,

ABSTRACT

Image processing has emerged as a promising tool for plant species recognition, allowing individuals to capture images with their mobile phones in the field and identify plant species or a list of closely related plants. Deep learning, particularly Convolutional Neural Networks (CNNs), has become the leading approach in image recognition tasks. This study explores the use of transfer learning, a deep learning technique, for automatic plant species recognition. Transfer learning involves using pre-trained CNN models, originally trained on large datasets like ImageNet, and fine-tuning them for specific tasks with smaller datasets. In this research, six pre-trained CNN models—VGG16, VGG19, DenseNet121, InceptionResNetV2, MobileNet, and MobileNetV2—were evaluated on a dataset comprising 30 plant species. The goal is to determine which transfer learning model performs best for plant species recognition.

I.INTRODUCTION

Image processing is considered a promising tool for plant species recognition, enabling individuals to take pictures with their mobile phone cameras in the field and identify the plant species or a list of closely related plants [1]. When a computer application assists people in accurately identifying plants, it not only helps in recognizing various species but also raises awareness among the public about the importance of protecting them [2]. In manual recognition system, scientists use different characteristics of the plant such as seeds, fruits, flower, stem and leaf [3].

A key aspect of plant identification presents a significant scientific and technical challenges. These challenges arise not only because of the vast diversity of plant species but also because of their highly varied taxonomic characteristics [4]. For this reason, using manual approaches for plant recognition is a time consuming and demanding [5].

Therefore, it became necessary to develop an automated system for plant identification.This system involves capturing images with a smartphone, which can then be analyzed using image processing software or applications to identify the specific plant species. The analysis includes several steps such as preprocessing to enhance image quality, feature extraction to isolate important parts of the image like leaf shape and texture, and classification using machine learning or deep learning algorithms to match the extracted features with a database of known plant species [1]. Currently, deep learning is widely used in various artificial intelligence applications, especially in image recognition and classification tasks [6],[7]. Different models of Convolutional neural networks (CNNs) are generally used in this tasks [8]. Transfer learning is a technique of deep learning that uses a pre-trained CNNs model on various problems. Transfer learning is a valuable technique when there is a shortage of datasets or limited computational resources. It allows models pre-trained on large datasets like ImageNet to be fine-tuned for specific tasks with smaller datasets. In this paper, we present an approach that uses transfer learning for plant species recognition. Six pre-trained models of CNNs such as VGG16, VGG19, Mobile Net and DenseNet have been tested on a dataset of 30 classes. Our goal is to decide which transfer leaning model is more appropriate for plant species recognition.

II.RELATED WORKS

This section discusses different methods that have been used for species plant recognition using image processing and deep learning.In [9], the authors proposed research that uses deep learning for recognize local fruits. They used transfer learning models such VGG19, Inception-V3, ResNet-50, and MobileNet on a dataset of eight classes and 3240 samples. The best results are obtained by MobileNet with an accuracy of 99.21%. In [10], the authors used AlexNet model for fruits freshness classification. This model gives an accuracy of 99,3%, 98.2% and 99.8% on three datasets. In [11], the authors suggested a CNN model using data augmentation for plant classification to overcome the problem of insufficient dataset. This work used four dataset which are Fruits-360, PlantVillage, PlantDoc and Plants.

This method showed higher performance comparted to other methods when the experiments were tested on the same datasets. In [12], the authors used VGG16 CNN model for fruits classification. Six classes of the most known fruits were used for the experiments. The results showed the classification accuracy of 94.16%. In [13], the authors used a plant dataset that have 30000 images and contains 100 ornamental species. These images were collected from Beijing Forestry University campus. The proposed ResNet-based model suggested in this work achieved a classification accuracy of 91.78. In [5], the authors proposed a hybrid approach that uses the histogram of oriented gradients vector to extract features. Then, they used those features to make classification using SVM. Secondly, they used CNN for plant species recognition. They achieved an accuracy of 98.22 on Swidish dataset when data is augmented.

III. THE PROPOSED APROACH

Transfer learning is one of the powerful techniques that has been extensively utilized for image recognition applications because of their hierarchical structure and their features extraction capabilities[14]. Transfer learning is a machine learning technique where a pre-trained model, is developed for one task and is repurposed for a different related task. In the context of Convolutional Neural Networks (CNNs), it involves using the learned features from a model trained on a large dataset to improve performance and reduce training time on a smaller, target dataset. It is proved that CNNs can achieve better performance than the classical methods [15].The proposed approach uses transfer leaning for automatic species plant identification. To achieve this goal, four CNN models were applied on a dataset of 30 classes. The flowchart of the proposed approach is shown in Figure 1.

IV.1. PRE-TRAINED CNN MODELS

Six CNN models were utilized in this work which are DenseNet, MobileNet, MobileNetV2, InceptionResNetV2. VGG16 is a CNN model developed by the visual Geometry Group at the University of Oxford. VGG16 modified AlexNet by using 3x3 kernels with 1 stride instead of 1x1 and 5x5 which allows for obtaining complicated features with short time computation. VGG16 is composed of 5 convolutional blocks. Each block have 2 to 3 convolutional layers [16]. All convolutional layers have Relu activation.

VGG19 is a convolutional neural network architecture that was proposed by the Visual Geometry Group (VGG) at the University of Oxford. It is similar to VGG16 but differ in the depth of the layers. It has 6 convolutional layers, 3 fully connected layers, 5 max-pooling layers and total of 19 weight layers. It gave a classification accuracy rate of 88% on the ImageNet dataset [17].

MobileNet is a deep CNN network was proposed by Howard to overcame the problem of using high computational resources. It is suitable for devices with limited resource such as IoT and smartphones devices [18]. MobileNet uses a single filter in the input layer which reduces the computation and uses a 1x1 convolution to join the outputs of the depthwise convolution [19].

Residual Network or ResNet was developed by [20]. ResNet is composed of the residual blocks. Each block a small number of convolutional layers. ResNet have shortcut connection that join directly the input of block by its output. RetNet50 that was used in this work is a specific type of the ResNet. It has 50 layers and is one of the most widely and known model of the ResNet types because of its tradeoff between computational efficiency, performance and the depth [21].

InceptionResNetV2 was introduced by Christian Szegedy [22]. It integrates the power of the residual networks and InceptionNetworks. It is composed of 164 layers and it can classify 1000 objects. It have a good balance between performance and resource requirements [23]. DenseNet is a deep Neural network in which the input of each layer is the concatenation of the outputs of all preceding layers within the same block in a feed-forward fashion to guarantee the maximum information stream between layers [24].

IV.2. DATASET:

To develop our automatic plan species recognition method, we downloaded various plant images from Kaggle. The used dataset is composed of 26970 plant images of 30 classes. Each class have 790 images for training and 100 images for test of different sizes. The 30 plant species are Aloevera, banana, bilimbi, cantaloupe, cassava, coconut, corn, cucumber, curcuma, eggplant, galangal, ginger, guava, kale, longbeans, mango, melon, orange, paddy, papaya, peper chili, pineapple, pomelo, shallot, soybeans, spinach, sweet potatoes, tobacco, waterapple and watermelon.

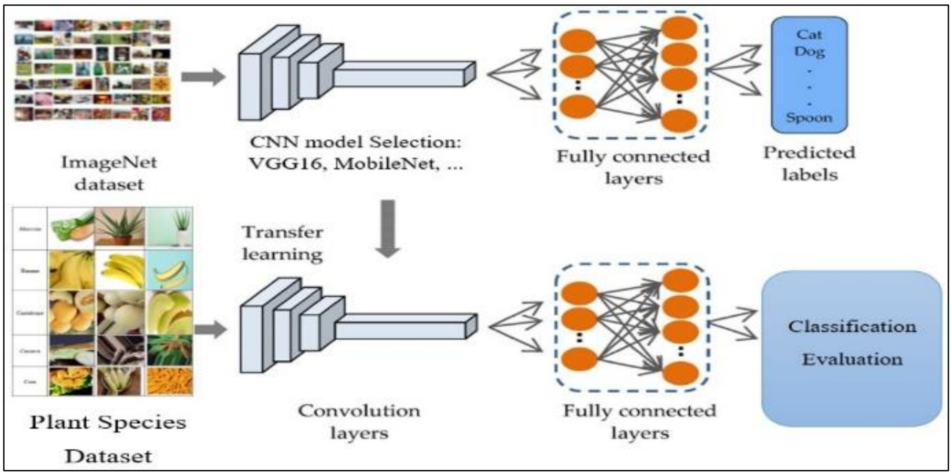

Figure 1: Flowchart of the proposed transfer learning methodology.
Source: Authors, (2025).

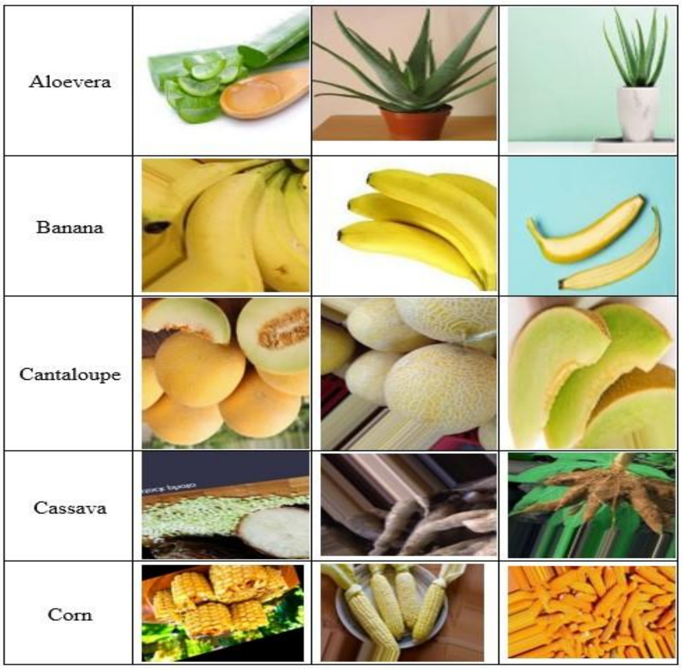

Figure 2: Samples of the different plant species of the used dataset.
Source: Authors, (2025).

IV.EXPERIMENT AND DISCUSSION

IV.1.EXPERIMENT SETTINGS

The six deep learning models were applied on machine using python 2.6 with Keras and Tensorflow. Each model is run on Windows 10 (64 bits) with Intel® Core™ i5-7200U CPU @ 2.50GHz 2.71GHz and 16 Go of RAM. The Adam optimizer with 50 epochs, 32 batch and a learning rate of 0.001 is used to train each model. The sparse categorical cross-entropy was utilized as a loss function. The models used weights pre-trained on ImageNet dataset for transfer learning. The evaluation of deep learning models for plant species recognition is done using the following measurements criteria: The Accuracy is ratio of the number of samples correctly predicted to the overall data. The accuracy is calculated using the following expression [16]:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN++FN+FP} \quad (1)$$

Where TP and FP represent the number of positive samples classified as true and false respectively and TN and FN represent the number of negative samples classified as true and false respectively. The Precision is the ratio of the number of positive samples correctly classified to the overall of samples positive classified. The precision is computed as:

$$\text{Precison} = \frac{TP}{TP+FP} \quad (2)$$

The Recall is the ratio of the number of positive samples correctly classified to the overall of positive samples.

The recall is calculated as:

$$\text{Recall} = \frac{TP}{TP+FN} \quad (3)$$

Where FN represents the count of false negatives

The F1Score combine between precision and recall into a single metric. It is calculated using the following equation:

$$F_1\text{Score} = \frac{2*Precision*Recall}{Precision+Recall} \quad (4)$$

The values of accuracy, precision, recall and F1-score of each model are shown in Table .1.

IV.2.DISCUSSION:

It can be seen from table 1 that Cucumber, Kale, Longbeans, and Sweet Potatoes have high metrics across all four parameters, indicating strong VGG16 performance in identifying these classes. Cantaloupe and Melon show significantly lower scores, particularly in Recall and F1-score, suggesting that VGG16 struggles with these classes. It can be seen from table 2 that Papaya, Peper Chili, Sweet Potatoes, and Spinach demonstrate high performance across all metrics, indicating that the VGG19 model effectively recognizes these classes. Cantaloupe has the lowest performance metrics, particularly in Recall and F1-score, indicating significant difficulty for the model in correctly identifying this class. Curcuma and Galangal also show relatively lower performance, suggesting the need for improvement. The VGG16 shows higher mean values across all metrics compared to VGG19, indicating generally better performance with the former model. From table 3, it can be seen that Longbeans, Corn, Paddy, and Aloevera demonstrate exceptionally high performance across all metrics, indicating MobileNet effectively recognizes these classes. Melon and Cantaloupe show the lowest performance metrics, particularly in Precision, Recall, and F1-score, suggesting significant difficulty for the model in correctly identifying these classes. MobileNet shows the highest mean values across all metrics compared to VGG19 and VGG16, indicating superior overall performance.

Table 1: Different evaluation metrics obtained by using VGG16.

Class	Accuracy	Precision	Recall	F1-score
Aloevera	0.829	0.843	0.860	0.851
banana	0.885	0.855	0.812	0.833
bilimbi	0.872	0.818	0.775	0.796
cantaloupe	0.694	0.585	0.475	0.523
cassava	0.888	0.856	0.885	0.870
coconut	0.818	0.779	0.779	0.779
corn	0.855	0.859	0.875	0.867
cucumber	0.903	0.895	0.949	0.921
curcuma	0.779	0.759	0.737	0.748
eggplant	0.852	0.842	0.800	0.821
galangal	0.797	0.812	0.675	0.737
ginger	0.828	0.829	0.831	0.830
guava	0.803	0.789	0.833	0.810
kale	0.907	0.892	0.923	0.907
longbeans	0.928	0.901	0.912	0.906
mango	0.806	0.723	0.747	0.735
melon	0.721	0.742	0.659	0.698
orange	0.858	0.875	0.820	0.846
paddy	0.833	0.839	0.791	0.814
papaya	0.834	0.857	0.939	0.896
peper chili	0.906	0.851	0.957	0.901
pineapple	0.900	0.861	0.882	0.872
pomelo	0.886	0.855	0.812	0.833
shallot	0.840	0.844	0.852	0.848
soybeans	0.868	0.844	0.802	0.822
spinach	0.862	0.874	0.889	0.881
sweet potatoes	0.905	0.884	0.889	0.886
tobacco	0.853	0.829	0.778	0.803
waterapple	0.871	0.830	0.843	0.836
watermelon	0.846	0.802	0.766	0.784
Mean	**0.847**	**0.829**	**0.823**	**0.825**

Source: Authors, (2025).

Table 4 shows that Waterapple, Sweet Potatoes, Melon, Cucumber, and Peper Chili demonstrate exceptionally high performance across all metrics, indicating MobileNetV2 effectively recognizes these classes. Spinach, Coconut, Ginger, and Tobacco show the lowest performance metrics, particularly in Precision, Recall, and F1-score, suggesting significant difficulty for the model in correctly identifying these classes. MobileNetV2 shows slightly higher mean values for Precision and Recall compared to MobileNet. However, the F1-score is almost identical, showing consistent performance between the two models.

Table 2: Different evaluation metrics obtained by using VGG19.

Class	Accuracy	Precision	Recall	F1-score
Aloevera	0.770	0.845	0.770	0.794
banana	0.853	0.870	0.800	0.833
bilimbi	0.724	0.707	0.820	0.759
cantaloupe	0.533	0.595	0.440	0.506
cassava	0.828	0.763	0.780	0.771
coconut	0.790	0.790	0.820	0.805
corn	0.777	0.724	0.875	0.792
cucumber	0.824	0.752	0.760	0.756
curcuma	0.686	0.666	0.700	0.682
eggplant	0.724	0.779	0.760	0.765
galangal	0.704	0.670	0.690	0.678
ginger	0.805	0.821	0.790	0.800
guava	0.795	0.780	0.790	0.785
kale	0.821	0.831	0.840	0.835
longbeans	0.851	0.844	0.860	0.851
mango	0.763	0.787	0.760	0.770
melon	0.698	0.708	0.650	0.675
orange	0.834	0.839	0.820	0.829
paddy	0.770	0.740	0.820	0.778
papaya	0.876	0.868	0.890	0.878
peper chili	0.899	0.875	0.945	0.909
pineapple	0.845	0.828	0.820	0.823
pomelo	0.865	0.818	0.820	0.819
shallot	0.826	0.810	0.840	0.823
soybeans	0.774	0.740	0.815	0.773
spinach	0.845	0.862	0.860	0.861
sweet potatoes	0.858	0.834	0.890	0.861
tobacco	0.804	0.781	0.770	0.775
waterapple	0.841	0.812	0.850	0.829
watermelon	0.841	0.860	0.780	0.807
Mean	**0.796**	**0.783**	**0.791**	**0.785**

Source: Authors, (2025).

Table 5 shows that Banana, Watermelon, Tobacco, Paddy, Eggplant, Galangal, Orange, Pepper Chili, and Waterapple demonstrate exceptionally high performance across all metrics, indicating that DenseNet effectively recognizes these classes. Cantaloupe shows the lowest performance metrics, particularly in Accuracy, Recall, and F1-score, suggesting significant difficulty for the model in correctly identifying this class. Coconut, Curcuma, Shallot, and Bilimbi also show relatively lower performance compared to other classes, though better than Cantaloupe.

DenseNet shows strong overall performance, with mean Precision and F1-score similar to MobileNetV2 and slightly better than MobileNet. However, its mean Accuracy is slightly lower than MobileNetV2 and MobileNet.

Table 3: Different evaluation metrics obtained by using MobileNet

Class	Accuracy	Precision	Recall	F1-score
Aloevera	0.925	0.902	0.950	0.925
banana	0.906	0.880	0.880	0.880
bilimbi	0.900	0.900	0.900	0.900
cantaloupe	0.850	0.745	0.850	0.794
cassava	0.905	0.950	0.950	0.927
coconut	0.870	0.825	0.870	0.847
corn	0.945	0.935	0.945	0.940
cucumber	0.870	0.780	0.840	0.809
curcuma	0.900	0.850	0.850	0.850
eggplant	0.885	0.850	0.850	0.850
galangal	0.890	0.860	0.860	0.860
ginger	0.820	0.820	0.820	0.820
guava	0.885	0.825	0.870	0.847
kale	0.890	0.890	0.980	0.933
longbeans	0.980	0.980	0.980	0.980
mango	0.900	0.870	0.870	0.870
melon	0.860	0.370	0.370	0.370
orange	0.935	0.830	0.830	0.830
paddy	0.925	0.950	0.950	0.950
papaya	0.900	0.845	0.900	0.872
peper chili	0.890	0.890	0.890	0.890
pineapple	0.900	0.830	0.900	0.864
pomelo	0.905	0.905	0.905	0.905
shallot	0.855	0.820	0.855	0.837
soybeans	0.880	0.840	0.840	0.840
spinach	0.895	0.845	0.845	0.845
sweet potatoes	0.900	0.850	0.850	0.850
tobacco	0.900	0.900	0.900	0.900
waterapple	0.870	0.865	0.865	0.865
watermelon	0.910	0.905	0.910	0.907
Mean	0.893	0.859	0.887	0.863

Source: Authors, (2025).

Table 6 shows that Banana, Cantaloupe, Corn, Melon, and Watermelon demonstrate relatively high performance across all metrics, indicating InceptionResNetV2 effectively recognizes these classes. Coconut shows the lowest performance metrics, particularly in Recall and F1-score, suggesting significant difficulty for the model in correctly identifying this class. Mango and Curcuma also show relatively lower performance compared to other classes, with Mango showing particularly poor Precision and F1-score. InceptionResNetV2 shows the lowest overall performance metrics compared to DenseNet, MobileNetV2, and MobileNet. Its mean Accuracy, Precision, Recall, and F1-score are all lower than the other models, indicating that InceptionResNetV2 is less effective for this particular classification task.

Table 4: Different evaluation metrics obtained by using MobileNetV2.

Class	Precision	Recall	F1-Score
Aloevera	0,91	0,91	0,91
Banana	0,97	0,88	0,92
Bilimbi	0,77	0,91	0,84
Cantaloupe	0,89	0,85	0,87
Cassava	0,9	0,94	0,92
Coconut	0,86	0,82	0,84
Corn	0,93	0,92	0,92
Cucumber	0,95	0,91	0,93
Curcuma	0,94	0,88	0,91
Eggplant	0,94	0,88	0,91
Galangal	0,85	0,94	0,89
Ginger	0,84	0,88	0,86
Guava	0,89	0,89	0,89
Kale	0,95	0,91	0,93
Longbeans	0,85	0,93	0,89
Mango	0,89	0,87	0,88
Melon	0,91	0,96	0,93
Orange	0,86	0,9	0,88
Paddy	0,87	0,96	0,91
Papaya	0,91	0,88	0,89
Peper Chili	0,95	0,92	0,93
Pineapple	0,85	0,85	0,85
Pomelo	0,9	0,85	0,87
Shallot	0,88	0,85	0,86
Soybeans	0,91	0,86	0,88
Spinach	0,84	0,83	0,83
Sweet Potatoes	0,93	0,95	0,94
Tobacco	0,89	0,84	0,86
Waterapple	0,97	0,95	0,96
Watermelon	0,94	0,95	0,94
Mean	0,90	0,90	0,89

Source: Authors, (2025).

Overalll from table 7, it can be see.

is the first choice for this classification task given its top performance across all metrics. MobileNet and DenseNet121 are also alternative Options with robust performance.

The confusion matrix of each model is shown in Fig. 5. Considering the confusion matrix of MobleNetV2, it can be seen that there are clear confusions between Melon and Cantaloupe, Pomelo and Coconut, Curcuma and Ginger and Orange and Pomelo.

Table 5: Different evaluation metrics obtained by using DenseNet.

Class	Accuracy	Precision	Recall	F1-Score
Aloevera	0.87	0.89	0.88	0.88
Banana	0.97	0.98	0.97	0.97
Bilimbi	0.84	0.86	0.85	0.85
Cantaloupe	0.75	0.78	0.75	0.76
Cassava	0.84	0.87	0.84	0.85
Coconut	0.81	0.83	0.81	0.81
Corn	0.91	0.92	0.91	0.91
Cucumber	0.90	0.91	0.90	0.90
Curcuma	0.83	0.86	0.83	0.84
Eggplant	0.92	0.94	0.92	0.92
Galangal	0.92	0.94	0.92	0.92
Ginger	0.87	0.89	0.87	0.87
Guava	0.88	0.89	0.88	0.88
Kale	0.86	0.89	0.86	0.87
Longbeans	0.87	0.89	0.87	0.88
Mango	0.88	0.90	0.88	0.88
Melon	0.84	0.86	0.84	0.84
Orange	0.92	0.93	0.92	0.92
Paddy	0.95	0.95	0.95	0.95
Papaya	0.89	0.90	0.89	0.89
Pepper Chili	0.92	0.94	0.92	0.92
Pineapple	0.90	0.91	0.90	0.90
Pomelo	0.87	0.89	0.87	0.87
Shallot	0.84	0.86	0.84	0.84
Soybeans	0.90	0.91	0.90	0.90
Spinach	0.88	0.90	0.88	0.88
Sweet Potatoes	0.88	0.90	0.88	0.88
Tobacco	0.95	0.96	0.95	0.95
Waterapple	0.92	0.94	0.92	0.92
Watermelon	0.96	0.97	0.96	0.96
Mean	**0.88**	**0.90**	**0.88**	**0.89**

Source: Authors, (2025).

Table 6: Different evaluation metrics obtained by using InceptionResNetV2.

Class	Accuracy	Precision	Recall	F1-Score
Aloevera	0.75	0.75	0.75	0.75
banana	0.89	0.98	0.83	0.90
bilimbi	0.83	0.68	0.82	0.74
cantaloupe	0.85	0.87	0.83	0.85
cassava	0.70	0.91	0.68	0.78

coconut	0.55	0.64	0.46	0.53
corn	0.85	0.87	0.82	0.85
cucumber	0.71	0.63	0.71	0.67
curcuma	0.59	0.85	0.58	0.69
eggplant	0.72	0.86	0.69	0.77
galangal	0.73	0.63	0.62	0.63
ginger	0.73	0.69	0.57	0.62
guava	0.75	0.77	0.61	0.68
kale	0.76	0.78	0.65	0.71
longbeans	0.84	0.82	0.86	0.84
mango	0.55	0.33	0.57	0.42
melon	0.80	0.77	0.80	0.79
orange	0.80	0.64	0.83	0.72
paddy	0.71	0.72	0.70	0.71
papaya	0.75	0.89	0.70	0.78
peper chili	0.75	0.65	0.79	0.71
pineapple	0.77	0.69	0.65	0.67
pomelo	0.71	0.55	0.65	0.60
shallot	0.76	0.67	0.69	0.68
soybeans	0.76	0.59	0.82	0.69
spinach	0.83	0.69	0.82	0.75
sweet potatoes	0.81	0.70	0.83	0.76
tobacco	0.80	0.58	0.68	0.63
waterapple	0.76	0.79	0.68	0.73
watermelon	0.82	0.88	0.79	0.83
Mean	**0.75**	**0.73**	**0.71**	**0.71**

Source: Authors, (2025).

Table 7: Mean values of each metric for each model.

Class	Accuracy	Precision	Recall	F1-Score
VGG16	0.84	0.829	0.823	0.82
VGG19	0.79	0.783	0.791	0.78
MobileNet	0.8	0.859	0.887	0.86
MobileNetV2	**0,90**	**0,90**	**0,89**	**0,90**
DenseNet121	0.88	**0.90**	0.88	0.89
InceptionResNetV2	0.75	0.73	0.71	0.71

Source: Authors, (2025).

V.CONCLUSION

The study presented an approach for automatic plant species recognition using transfer learning with six pre-trained CNN models: VGG16, VGG19, DenseNet121, InceptionResNetV2 MobileNet, and MobileNetV2. These models were tested on a dataset of 30 plant species. The experimental results demonstrated that transfer learning is highly effective for plant species recognition, with MobileNetV2 showing the best overall performance across all evaluation metrics. The MobileNetV2 model achieved the highest accuracy, precision, recall, and F1-score, making it the most suitable model for this task. MobileNet and DenseNet also showed strong performance and can be considered as alternativ .

VI. AUTHOR'S CONTRIBUTION

Conceptualization: OUAHAB Abdelwhab.
Methodology: OUAHAB Abdelwhab.
Investigation: OUAHAB Abdelwhab.
Discussion of results: OUAHAB Abdelwhab.
Writing – Original Draft: OUAHAB Abdelwhab.
Writing – Review and Editing: OUAHAB Abdelwhab.
Resources: OUAHAB Abdelwhab.
Supervision: OUAHAB Abdelwhab..
Approval of the final text: OUAHAB Abdelwhab.

VII.REFERENCES

[1] J. Banzi and T. Abayo, "Plant Species Identification from Leaf Images Using Deep Learning Models (CNN-LSTM Architecture)," 90(3), pp. 1–X, 2021.

[2] D. Zhou, X. Ma, and S. Feng, "An Effective Plant Recognition Method with Feature Recalibration of Multiple Pretrained CNN and Layers," Appl. Sci., vol. 13, no. 7, pp. 4531, Apr. 2023. doi: 10.3390/app13074531.

[3] N. Goyal, K. Gupta, and N. Kumar, "Multiclass Twin Support Vector Machine for plant species identification," Multimed. Tools Appl., vol. 78, no. 19, pp. 27785–27808, 2019. doi: 10.1007/s11042-019-7588-2.

[4] B. Yanikoglu, E. Aptoula, and C. Tirkaz, "Automatic plant identification from photographs," Mach. Vis. Appl., vol. 25, no. 6, pp. 1369–1383, 2014. doi: 10.1007/s00138-014-0612-7.

[5] T. Quoc Bao, N. T. Tan Kiet, T. Quoc Dinh, and H. X. Hiep, "Plant species identification from leaf patterns using histogram of oriented gradients feature space and convolution neural networks," J. Inf. Telecommun., vol. 4, no. 2, pp. 140–150, 2020. doi: 10.1080/24751839.2019.1666625.

[6] O. V. Grinchuk and V. I. Tsurkov, "Cyclic Generative Neural Networks for Improved Face Recognition in Nonstandard Domains," J. Comput. Syst. Sci. Int., vol. 57, no. 4, pp. 620–625, 2018. doi: 10.1134/S1064230718040093.

[7] Yu. V. Vizil'ter, O. V. Vygolov, D. V. Komarov, and M. A. Lebedev, "Fusion of Images of Different Spectra Based on Generative Adversarial Networks," J. Comput. Syst. Sci. Int., vol. 58, no. 3, pp. 441–453, 2019. doi: 10.1134/S1064230719030201.

[8] Y. S. Efimov, V. Y. Leonov, G. Odinokikh, and I. Solomatin, "Finding the Iris Using Convolutional Neural Networks," J. Comput. Syst. Sci. Int., vol. 60, pp. 108–117, 2021.

[9] Md. M. Rahman, Md. A. Basar, T. S. Shinti, Md. S. I. Khan, H. Md. H. Babu, and K. M. M. Uddin, "A deep CNN approach to detect and classify local fruits through a web interface," Smart Agric. Technol., vol. 5, pp. 100321, 2023. doi: 10.1016/j.atech.2023.100321.

[10] U. Amin, M. I. Shahzad, A. Shahzad, M. Shahzad, U. Khan, and Z. Mahmood, "Automatic Fruits Freshness Classification Using CNN and Transfer Learning," Appl. Sci., vol. 13, no. 14, pp. 8087, 2023. doi: 10.3390/app13148087.

[11] G. Batchuluun, S. H. Nam, and K. R. Park, "Deep Learning-Based Plant-Image Classification Using a Small Training Dataset," Mathematics, vol. 10, no. 17, pp. 3091, 2022. doi: 10.3390/math10173091.

[12] Y. Zhan, Y. Chen, X. Li, M. Chen, and Z. Luo, "Fruit Recognition Based on Convolution Neural Network," J. Phys. Conf. Ser., vol. 1651, no. 1, pp. 012176, 2020. doi: 10.1088/1742-6596/1651/1/012176.

[13] Y. Sun, Y. Liu, G. Wang, and H. Zhang, "Deep Learning for Plant Identification in Natural Environment," Comput. Intell. Neurosci., pp. 1–6, 2017. doi: 10.1155/2017/7361042.

[14] F. Risdin, P. K. Mondal, and K. M. Hassan, "Convolutional Neural Networks (CNN) for Detecting Fruit Information Using Machine Learning Techniques," IOSR J. Comput. Engin., vol. 22, no. 2, pp. 1–13, 2022.

[15] J. Chen, J. Chen, D. Zhang, Y. Sun, and Y. A. Nanehkaran, “Using deep transfer learning for image-based plant disease identification,” Comput. Electron. Agric., vol. 173, pp. 105393, 2020. doi: 10.1016/j.compag.2020.105393.

[16] D. Rosmala, M. R. P. Anggara, and J. P. Sahat, “Transfer learning with vgg16 and inceptionv3 model for classification of potato leaf disease,” J. Theor. Appl. Inform. Technol., vol. 99, no. 2, pp. 1–X, 2021.

[17] F. Özyurt, “Efficient deep feature selection for remote sensing image recognition with fused deep learning architectures,” J. Supercomput., vol. 76, no. 11, pp. 8413–8431, 2020. doi: 10.1007/s11227-019-03106-y.

[18] A. G. Howard et al., “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” arXiv preprint, 2017. http://arxiv.org/abs/1704.04861.

[19] J. Imran and B. Raman, “Deep motion templates and extreme learning machine for sign language recognition,” Vis. Comput., vol. 36, no. 6, pp. 1233–1246, 2020. doi: 10.1007/s00371-019-01725-3.

[20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA: IEEE, pp. 770–778, 2016. doi: 10.1109/CVPR.2016.90.

[21] Q. Wu, K. Zhang, and J. Meng, “Identification of Soybean Leaf Diseases via Deep Learning,” J. Inst. Eng. India Ser. A, vol. 100, no. 4, pp. 659–666, 2019. doi: 10.1007/s40030-019-00390-y.

[22] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. Alemi, “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning,” arXiv preprint, 2016. http://arxiv.org/abs/1602.07261.

[23] Y. Chen et al., “Classification of lungs infected COVID-19 images based on inception-ResNet,” Comput. Methods Programs Biomed., vol. 225, pp. 107053, 2022. doi: 10.1016/j.cmpb.2022.107053.

[24] Z. Tang, W. Jiang, Z. Zhang, M. Zhao, L. Zhang, and M. Wang, “DenseNet with Up-Sampling block for recognizing texts in images,” Neural Comput. Appl., vol. 32, no. 11, pp. 7553–7561, 2020. doi: 10.1007/s00521-019-04285-8.