Advancing Plant Species Recognition with Cascaded CNNs and Transfer Learning

A. Karnan, *Member, IAENG,* R. Ragupathy, *Member, IAENG*

***Abstract*—Classifying plant species is essential for ecological monitoring and biodiversity preservation. In this paper, a novel method is proposed for species classification that uses ResNet50 for feature extraction, k-means clustering for classifier selection, and cascaded pre-trained Convolutional Neural Networks (CNNs) for classification. ResNet50 is used to extract detailed features from plant images, which are further grouped by k-means clustering and it is used to select appropriate classifier. Subsequently, the extracted features from ResNet50 are processed using transfer learning by the cascaded pre-trained CNNs to produce a more reliable and precise species classification. PlantCLEF 2015 dataset which contains 1,13,205 images of different organs of 1000 species of trees, herbs and ferns living in Western European regions, is used to asses the performance of the proposed model. When compared to conventional pre-trained deep learning models namely ResNet50, Inception, MobileNet, Xception, and EfficientNet, the proposed pre-trained CNN model has superior performance with respect to standard performance metrics such as accuracy, precision, recall, F1 score, training time, parameter size, etc. A notable improvement is observed in training time and parameter size for classification using pre-trained deep learning models.**

***Index Terms*—Transfer Learning, Machine Learning Algorithms, Pre-trained Models, PlantCLEF 2015, Cascaded Convolutional Neural Networks, Plant Species Classification.**

I. Introduction

In the realm of deep learning, transfer learning has become a potent method, especially for a complex problem domain or little labeled data. Transfer learning provides substantial benefit in the classification of plant species by utilizing pre-trained models that have been created on sizable, varied datasets, like ImageNet. Numerous feature hierarchies and patterns are captured by these pre-trained models, including ResNet50, Inception, MobileNet, Xception, and EfficientNet, which can be adjusted to the particular job of plant species identification. Transfer learning's capacity to manage the varied and heterogeneous character of plant species, which frequently necessitates the extraction of intricate features, is one of its main advantages in the classification of plants. By utilizing pre-trained models, the need for extensive training from scratch is eliminated, allowing models to generalize better even with smaller datasets. This drastically reduces computational time and resources while still delivering high accuracy. Moreover, transfer learning can improve the robustness of the classification models. Pre-trained networks have already learned to recognize various textures, shapes, and structures from vast image datasets, making them more resilient to environmental variations, such as lighting or background changes, which are common in plant imagery. This adaptability is crucial for tasks like ecological monitoring, where data comes from uncontrolled and dynamic environments. As many challenges persist with PlantCLEF 2015 dataset, it is considered for this work. In the proposed methodology, transfer learning plays a vital role in the classification of plant species. These models are not only used for feature extraction, but also provide a solid foundation that can be fine-tuned for the specific plant species classification task. When combined with techniques like ResNet50 for feature extraction and k-means clustering for classifier selection, transfer learning helps to create a highly efficient and accurate classification system [1] [2]. In summary, transfer learning is a key enabler for advanced plant species classification. It accelerates model development, enhances performance, and improves the generalization of deep learning models in complex and diverse environments, making it an essential tool for ecological monitoring and biodiversity conservation initiatives. Hence, this paper uses transfer learning along with ResNet50 and k-means clustering for classification of more complicated PlantCLEF 2015 dataset [3]. The rest of the paper is organized as follows: Section 2 covers the literature review, Section 3 describes the materials and methods, Section 4 showcases and analyzes the experimental results, and Section 5 concludes with insights and suggestions for future research directions.

Manuscript received November 9, 2024; revised May 13, 2025.

A. Karnan is a Ph.D candidate of Department of Computer Science and Engineering, Annamalai University, Tamil Nadu - 608002, India. (corresponding author to provide phone: +91-8248600965; e-mail: mekarnan@gmail.com).

R. Ragupathy is a Professor of Computer Science and Engineering, Annamalai University, Tamil Nadu - 608002, India. (e-mail:cse_ragu@yahoo.com).

II. Literature Review

Importance of research need for classifying plant species is covered in this section. Conventional methods for classifying plant species usually include feature extraction, image preprocessing, and application of traditional machine learning techniques. Despite their widespread use, Support Vector Machines (SVMs), which rely on global form and local texture features, have limitations because of inconsistent feature extraction techniques and dataset variability [4]. Similarly, traditional neural networks paired with morphological feature extraction techniques can achieve moderate accuracy. However, they often struggle with small datasets and variations in plant structure, such as differences in leaf shape and texture [5] [6]. These traditional methods, while foundational, frequently encounter challenges related to accuracy and adaptability in diverse environmental conditions. Recent developments in deep learning have significantly improved plant species classification. Convolutional Neural Networks (CNNs), particularly architectures like ResNet and VGG16, have shown remarkable accuracy. Studies have demonstrated that

VGG16 can achieve classification accuracies as high as 95.0% for plant image datasets [7]. Additionally, more complex approaches, such as CNNs combined with Bidirectional Long Short-Term Memory (Bi-LSTM) networks for deep feature fusion, have further enhanced precision and classification performance [8]. Other innovative techniques, such as the VI-CNN model, which integrates hyper-spectral LiDAR data with CNNs, have achieved considerable improvements by analyzing both spectral and biological features [9]. The D-Leaf model, leveraging CNNs, has also outperformed traditional methods, emphasizing the potential of deep learning for automated plant species identification [10]. Numerous comprehensive studies and reviews of plant classification using machine learning and deep learning highlight the significant strides made in this area [11] [27]. The ongoing development of deep feature fusion techniques and advanced CNN architectures continues to enhance classification accuracy and robustness [12] [15]. Most recently, transfer learning has emerged as a valuable approach in plant species classification, particularly when limited labeled data are available. Pre-trained models, such as ResNet, Inception, and EfficientNet, which are initially trained on large datasets like ImageNet, enable the extraction of rich, hierarchical features that improve classification performance in target tasks. Fine-tuning of these models for specific plant classification challenges enhances both accuracy and robustness, especially in complex environments with varying conditions, such as changes in lighting or background [23] [26]. This approach minimizes the dependency on large labeled datasets and reduces the need for manual feature extraction by enabling deep learning models to automatically adapt to the unique characteristics of plant imagery. As transfer learning has become instrumental in improving the precision and stability of plant species classification systems [13] [14], this research work combines transfer learning with CNNs and k-means clustering, to outperform traditional deep learning methods in terms of standard performance metrics. Hence, the proposed methodology is more effective in handling limited labeled data, making it suitable for real-world applications.

III. Materials and Methods

A. Experimental Setup

The tests were carried out on a high-performance computing platform with an NVIDIA T4 GPU, which is more suitable for deep learning model training as it has 2,560 CUDA cores, 320 Tensor cores, 16 GB of GDDR6 memory, and a memory bandwidth of 320 GB/s. The system ran on Ubuntu OS with CUDA 11.x support and was powered by an Intel Xeon CPU with 51 GB of RAM. TensorFlow 2.x and PyTorch 1.x/2.x frameworks were used for model construction, training, and testing which made it possible to handle complex CNN architectures effectively and to conduct extensive model evaluation and experimentation quickly.

B. Datasets Utilized

The PlantCLEF 2015 dataset [30] was employed for training and testing, which consists of 1,13,205 images of different organs of 1000 species of trees, herbs and ferns living in Western European regions. This dataset is part of a citizen science initiative launched in collaboration with Tela Botanica, which has involved a community of botanists and plant enthusiasts over the past five years [24]. The dataset's images reflect real-world variations, having been captured by different users in diverse locations and under various conditions throughout the year. The dataset also includes metadata such as the author, date, location, and EXIF data. Recent updates to the dataset feature vote annotations contributed by the Tela Botanica community

Fig. 1: Sample images from the PlantCLEF 2015 dataset, showing different views and quality ratings

through the collaborative tool Pictoflora. The images encompass various views include photographs of ”leafscan”, ”leaf”, ”flower”, ”fruit”, ”stem”, ”entire”, and the recently added ”branch” view of a plant species. Image quality is rated by users on a scale of 1 to 5, where a higher rating (*****) indicates a clear and well-focused image suitable for identification [20] [25] as shown in Fig. 1.

C. System Architecture

The system architecture is divided into three primary stages: feature extraction, grouping of features for selecting classifier, and classification. Initially, augmentation is performed for each class of plant species to increase the number of samples for each class. Then, augmented dataset is pre-processed in sequence one by one from quality evaluation to normalization and resizing, further divided into two parts namely training and testing in the ratio of 80% and 20%. 20% of training data actually used for validation. which results in 60:20:20 ratio for training, validation and testing. As in every deep learning system, there are training and testing phases. In the training phase, initially features are extracted using ResNet50. Based on these extracted features, The clusters are formed using k-means clustering technique with Silhouette algorithm to determine the optimal value of k. Now each cluster is assigned to a deep learning model which is trained using transfer learning with pre-trained weights from the ImageNet dataset. These models consist of non-trainable layers (frozen layers) which have weights from the ImageNet dataset and trainable layers, allowing them to leverage existing knowledge while learning new patterns. Likewise, 15 CNN classifiers are trained and validated, since Silhouette algorithm determined the optimal value of k as 15. In the testing phase, feature extraction is performed on each testing sample. Then, k-means algorithm is employed on extracted features to select the corresponding pre-trained CNN model to predict the given multi-model image data. Majority voting scheme is applied on these predicted labels to take the final decision on plant species identification. Fig. 2 offers a visual depiction of the entire process from the input images to the final classification of plant species. This modular design enables efficient processing and classification of plant images by utilizing deep transfer learning and clustering techniques.

D. Feature Extraction using ResNet50

ResNet50, a deep convolutional neural network, is used in the system’s first step for feature extraction. ResNet50 is well-known for its capacity to extract intricate and significant features, which make it appropriate for image classification in difficult datasets [29]. To guarantee high-quality feature extraction, pre-trained weights were used from sizable ImageNet dataset through transfer learning. These traits enhance the classification process by highlighting distinctive traits of the plant species.

Clustering for Feature Grouping: The features extracted from ResNet50 are grouped using k-means clustering algorithm. The iterative process of k-means organizes the features into groups according to their principal similarities, improving the representation of the input dataset. By dividing the feature space into meaningful groups, k-means clustering helps to eliminate redundant information in the dataset. Furthermore, k-means clustering helps in the discovery of hidden patterns in the features extracted, which may not be easily discernible in their raw form. Such patterns help in enhancing the discriminative ability of the classification model by providing a feature representation that is semantically richer. Moreover, by clustering similar features, k-means reduces noise and outliers, thereby providing a more refined and robust feature set. The application of k-means clustering [16] [17] enhances the robustness and reliability of the output classification results while aligning feature space with data intrinsic structure and thus focusing on the most important and informative information for the CNNs. In this compact format of data presentation, the features extracted by ResNet50 has been further processed through the application of k-means clustering, arranging the data into clusters as conferred in [16]. The Silhouette Score is 15 for the PlantCLEF 2015 dataset. Therefore, the optimum number of clusters is 15 that are separated well and can be defined precisely. Finally, the dataset has been classified into 15 clusters with different kinds of plants belonging to different classes based on the visual features of the plants.

E. Cascaded Pre-trained CNNs for Species Classification

In the final stage, cascaded pre-trained CNNs are employed to classify the plant species. The cascading of pre-trained CNNs allows for a multi-layer analysis of the features [28], progressively refining the classification as the features pass through each layer. This method minimizes the risk of misclassification by relying on deeper representations of the plant characteristics [18]. Fig. 2 illustrates the process of applying cascaded pre-trained CNNs for final classification. This feature-based clustering groups the image into one of several categories, ranging from cluster-1 to cluster-15, by determining its similarity to other data points. After image assigned to a specific cluster, such as cluster-4, the system selects the corresponding classifier for that cluster, for example, classifier-4. By processing the image through this specialized classifier, the system increases the likelihood of accurate classification. Once the classifier has processed the image, the final classification result is produced as output class. This method effectively combines the use of unsupervised learning through k-means clustering with selected classifiers, aiming to enhance both efficiency and accuracy [19]. Likewise, all the input images which represent different views or modality (leaf, flower, fruit, stem, entire and branch) of a single plant are processed and the output class for each modality has been obtained. These obtained classes are taken into account by majority voting scheme to finalize the class of the plant species out of 1000 classes.

F. Proposed Transfer Learning Model Architecture

Fig. 3 illustrates the concept of transfer learning, a technique in deep learning where a model pre-trained on a large dataset is adapted for a new, related task. The left side of the diagram represents a CNN trained on the ImageNet dataset, which contains 1000 classes. This model consists of several layers, including an input layer, three convolutional layers, corresponding three max-pooling layers, one flatten layer, two fully connected layers and one softmax layer. The

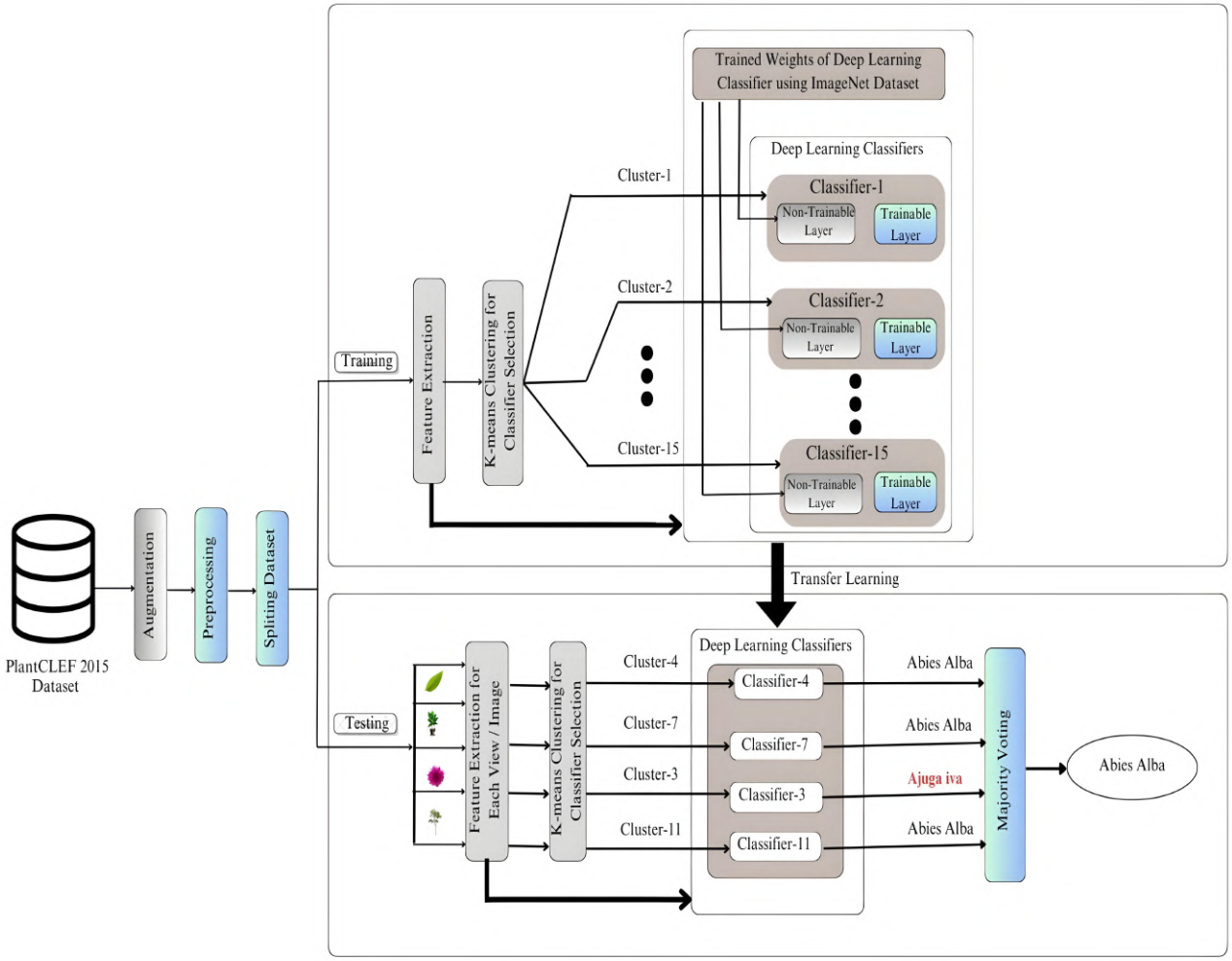

Fig. 2: System Architecture for Plant Species Classification

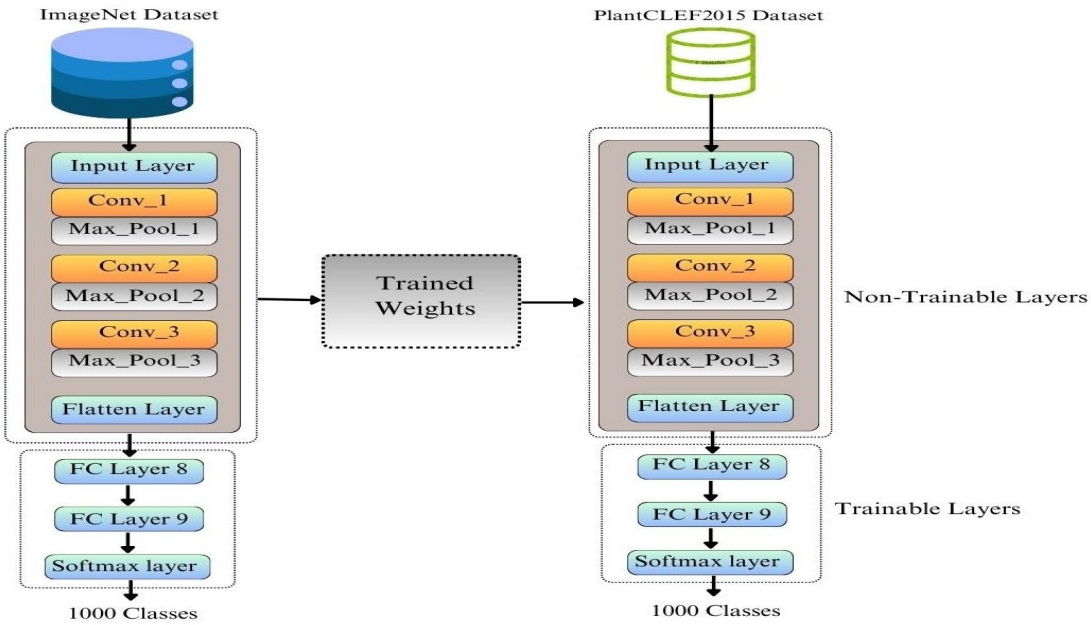

Fig. 3: Transfer Learning Framework

model starts with an input layer that handles images with 224 x 224 pixels size in RGB channels. Then, the convolutional layer, labelled as conv_1, employs 32 filters in order to extract edges and corners and followed up by a ReLU activation function to handle non-linearity. Next, a max-pooling layer is used to reduce the spatial dimensions to 111 × 111 × 32. Then the second convolutional layer, conv_2, uses 64 filters to detect more complex features and a max-pooling layer is employed to reduce the spatial dimensions to 54 × 54 × 64. The third convolutional layer, conv_3, uses 128 filters to focus on higher-level features like shapes or structures and a max-pooling layer is applied to reduce the spatial dimensions to 26 × 26 × 128. The convolution filter of size 3 x 3 and 2 x 2 max-pooling are used in this work. After each convolutional layer, max pooling is applied to down sample the feature maps, retaining only the most important features while reducing computational complexity and increasing robustness to variations [31]. The output of last max-pooling layer is fed into a flattening layer, which converts the multi-dimensional feature maps into a one-dimensional vector with a size of 1 × 1 × 86528. This feature vector is then forwarded to fully connected layers for high-level reasoning. The first fully connected layer (FC Layer 8) consists of 128 neurons, which combine learned features to identify the relationship relevant to the target classes. After this, the fully connected terminal layer (FC Layer 9) produces a vector whose size is equivalent to the number of classes i.e. 1000, where each element is a raw score or logit, associated with each class. The softmax layer then converts these logits into probabilities, thus allowing the model to assign a confidence score to each class. The class that has the highest probability is then considered as the output class. The complete transformation, starting from input image to feature maps and culminating the final classification of plant species is illustrated in Fig. 4. With this model architecture, training is performed with ImageNet dataset. After training on ImageNet, this model has learned meaningful feature representations that can be transferred to a different task.

The right side of the diagram shows how the pre-trained model is fine-tuned for a new dataset, the PlantCLEF 2015 dataset, which focuses on plant classification. Instead of training the entire network from scratch, the convolutional and max-pooling layers are frozen (i.e., their weights remain unchanged) and labeled as non-trainable layers. These layers continue to function as feature extractors, leveraging their learned representations. However, the remaining layers are trainable, meaning its weights will be updated to learn task-specific features for classifying 1000 plant classes instead of the original 1000 natural images. The transfer learning approach also enhances the model's adaptability by allowing it to generalize well across different plant species with minimal modifications. Therefore, by using this transfer learning approach, the model benefits from pre-existing knowledge, reducing computational costs, training time, and the need for large amounts of labeled data. This technique is especially useful in scenarios where collecting and annotating large datasets are challenging, allowing researchers to achieve high performance even with limited data.

G. Training and Testing Procedure

Initially, performance measurements were used to fine-tune hyper-parameters such learning rate, batch size, and number of epochs. To enhance model generalization, data augmentation methods such as flips, random rotations, and

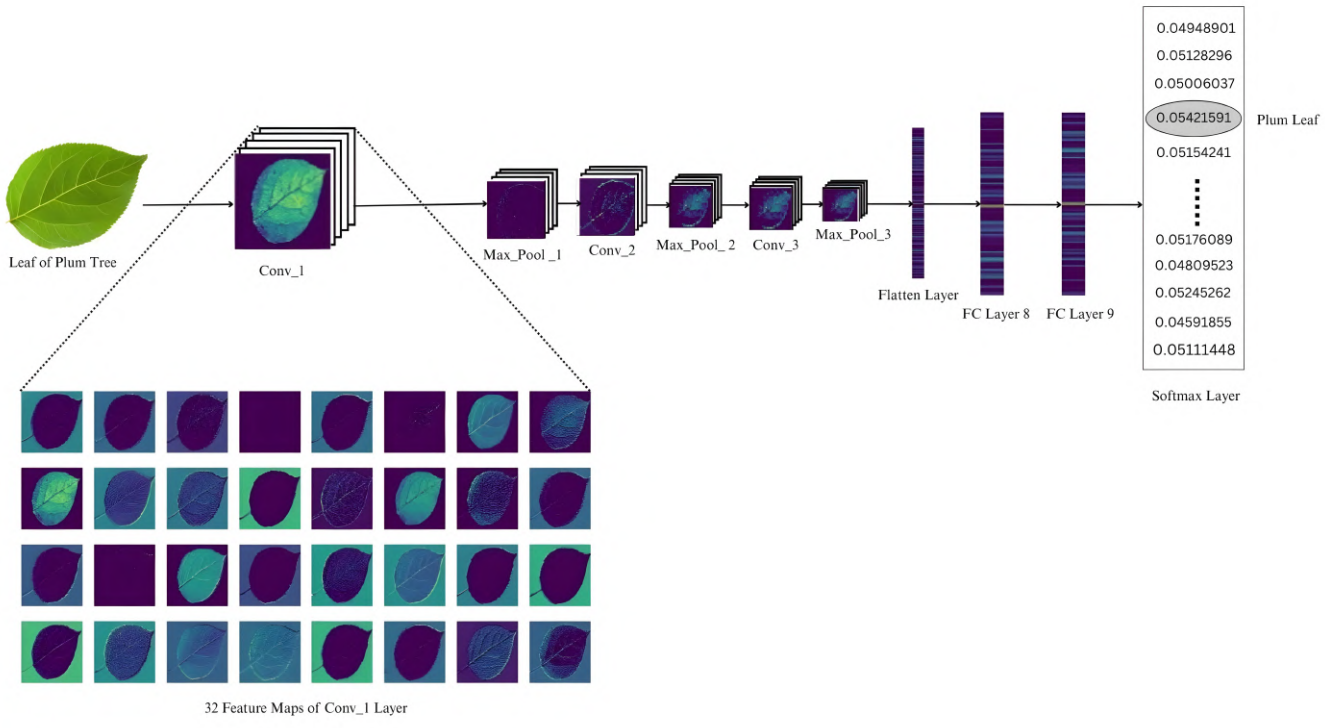

Fig. 4: Illustration of Feature Map Transformation on Each Layer of Proposed CNN Model

color changes were used. Strategies like balanced batch sampling and oversampling minority classes were used to overcome the possible class imbalance. Eighty percent of the images is utilized for training(20% for validation), while the remaining twenty percent is used for testing. This is performed by utilizing 60%-20%-20% split. For each sample, n number of images of same plant is feed into ResNet50 to extract features from each image. k-means clustering is then used to aggregate the features which help to select classifier appropriately. Next, cascaded pre-trained CNNs are used to learn classification task on PlantCLEF 2015 dataset with the help of knowledge gained from ImageNet dataset. In the testing phase, one among 15 pre-trained CNNs models is employed to classify the feature extracted by ResNet50 from plant images based on likelihood of k-means clustering. Likewise, model predicts classes for all the view (image) of the same plant which is considered as a single sample. the final class is obtained by making use of majority voting scheme from predicted classes of plant views.

H. Evaluation Metrics

A number of important criteria are used to assess the effectiveness of the suggested plant species classification system. By dividing the number of correctly categorized cases (both positive and negative) by the total number of instances, accuracy quantifies the model's overall correctness. While recall, sometimes referred to as sensitivity or true positive rate, indicates the model's capacity to locate all pertinent occurrences in the dataset, precision measures the model's capacity to accurately identify positive examples. The F1 score serves as a harmonic mean between precision and recall, providing a single metric that balances both. This metric is particularly valuable in scenarios where class distribution is uneven or when both precision and recall are critical. By employing these metrics, a comprehensive evaluation of the model's performance was achieved, highlighting its effectiveness in classifying different plant species.

IV. Results and Analysis

The proposed plant species classification system, combining ResNet50, k-means clustering, and transfer learning, is extensively tested using the PlantCLEF 2015 dataset. This section provides a detailed analysis of the experimental results, covering classification performance.

A. Classification Accuracy

The transfer learning based CNN architecture demonstrated remarkable performance in classifying 1000 plant species. By leveraging ResNet50 for feature extraction and transfer learning, the model achieved an overall accuracy of 97.8% on the validation set. This accuracy is significantly higher than traditional deep learning approaches such as ResNet50, Inception, MobileNet, Xception, and EfficientNet. Even it outperforms several standalone CNN-based architectures. To further validate the model's effectiveness, the F1-score, precision, and recall metrics were computed. These metrics are particularly important for ensuring that the model performs well across all classes, especially when dealing with imbalanced datasets. To understand the performance metrics calculation, 20 classes have been considered for better representation. From the testing process, predicted classes have been obtained for every classifier. Similarly, confusion matrix is calculated using predicted and actual classes for every classifier. The two confusion matrices of proposed model shown in Fig. 5 and Fig. 6 illustrate the impact of transfer learning on a plant species classification model. Fig. 5 represents the confusion matrix for without pre-trained model. Some classes, such as Abies alba and Achillea millefolium, exhibit high true positive values with 8 and 9 correct predictions, respectively. However, several other species show lower accuracy, indicating that the model struggles to differentiate between them effectively. To enhance the model's performance transfer learning is applied, its confusion matrix is shown in Fig. 5. In Fig. 6, the red annotations highlight specific cases where true positive values have increased as compared to the baseline model. For instance, Allium triquetrum plant species has a significant improvement, with correctly classified samples increasing from 2 to 6. Similarly, other species show better classification results due to the refined feature extraction capabilities of transfer learning. From confusion matrix, TP, TN, FP, and FN were calculated for each classes. Accuracy, precision, recall and F1 score were computed using TP, TN, FP and FN. The result of computation is tabulated and presented in Table I. The results demonstrate the robustness and reliability of the system, with a high F1-score indicating balanced performance across all species.

Table II presents a comparative analysis of traditional deep learning models and the proposed model without pre-trained weights. The comparison is based on key performance metrics, including total parameters, trainable parameters, average accuracy, average loss, and average training time per epoch (in seconds). Among the traditional models, MobileNet achieves the highest accuracy (0.8839) with the lowest number of parameters (35.85 million), making it a relatively efficient model. Inception (0.8824) and Xception (0.882) also perform well, but have significantly higher parameter counts, leading to longer training times. EfficientNet achieves a slightly lower accuracy (0.8773), but has a significantly reduced training time (50.5 seconds per epoch), indicating better computational efficiency. The proposed model outperforms all traditional models, achieving the highest accuracy (0.98) and the lowest average loss (0.0615). Notably, it does so with only 2,40,719 total parameters, which is significantly smaller than all the traditional models. This drastic reduction in parameter count leads to improved efficiency, requiring only 61.5 seconds per epoch, which is much faster than most other models except for EfficientNet. Overall, the results highlight that the proposed model achieves superior accuracy while being significantly lighter and computationally efficient, making it a promising alternative to traditional deep learning architectures, especially in resource-constrained environments.

A comparative analysis of different deep learning models with pre-trained weights is presented in Table III. This comparison highlights the efficiency and effectiveness of the proposed model in contrast to widely used architectures. Among the traditional models, MobileNet demonstrates the

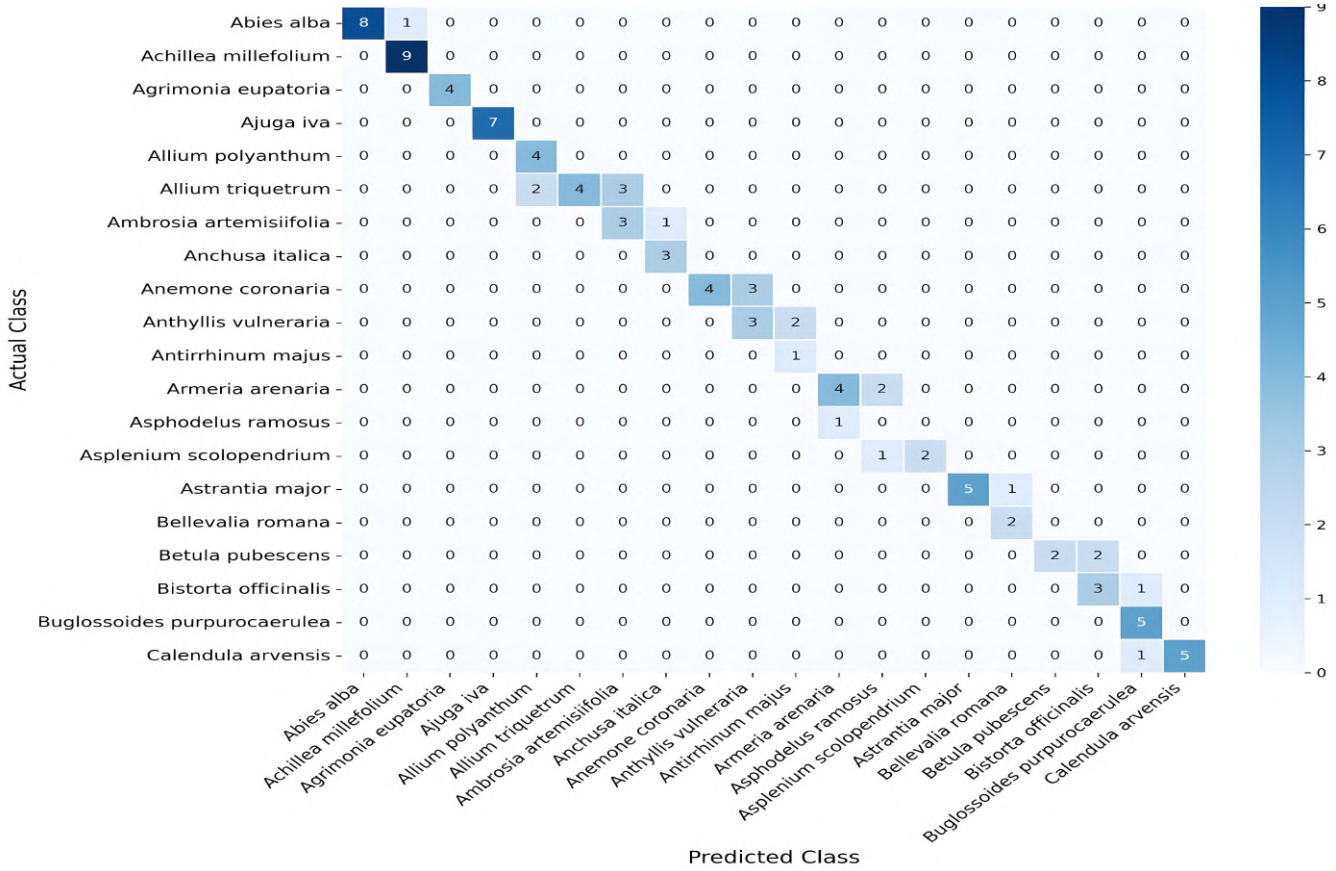

Fig. 5: Confusion Matrix of Proposed CNN Model without Pre-training

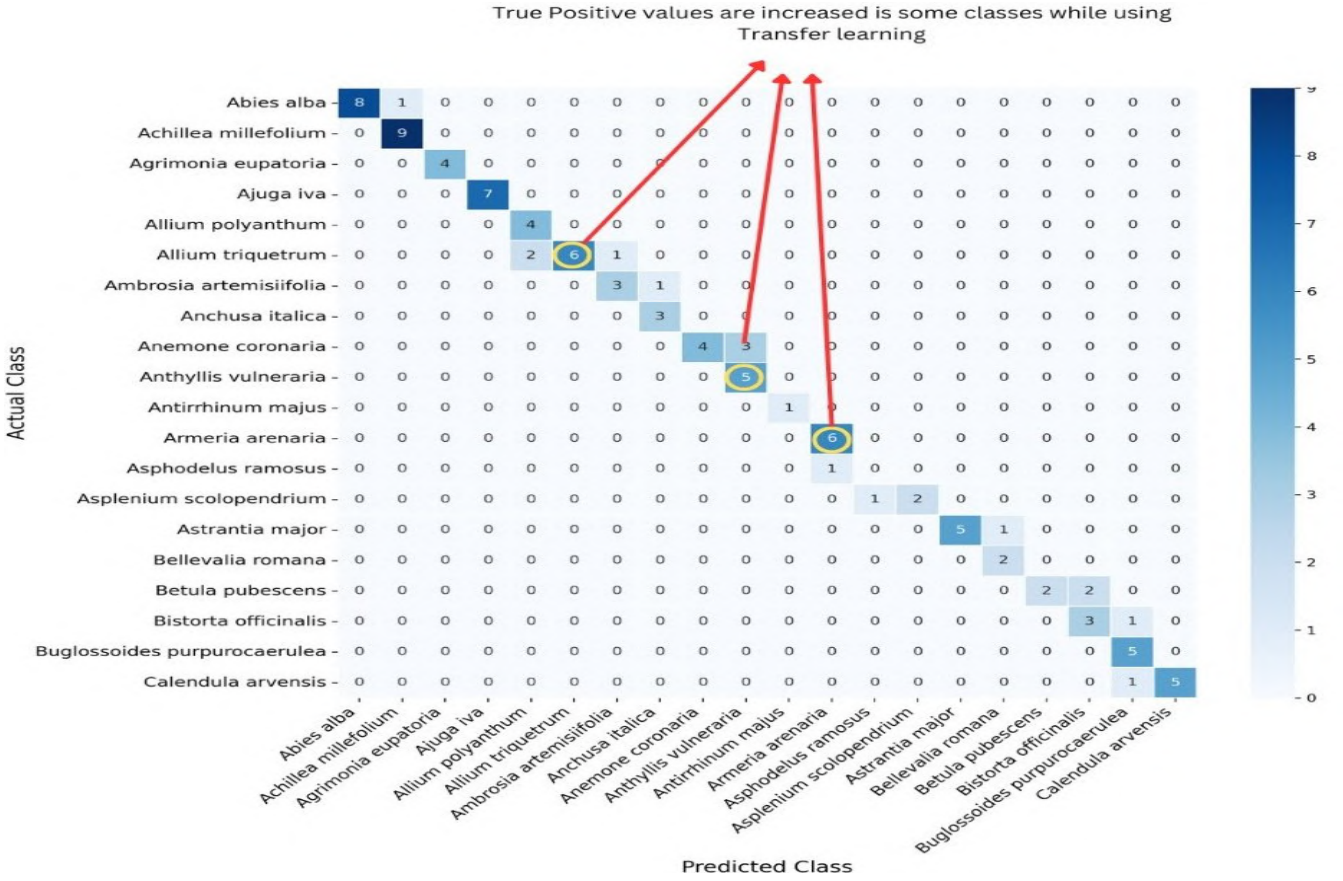

Fig. 6: Confusion Matrix of Proposed Pre-trained CNN Model

TABLE I: Classification Report of Proposed Model with Pre-Trained Weights

S.No	Class	TP	FP	TN	FN	Accuracy	Precision	Recall	F1-Score
1	Abies alba	8	1	520	0	1.00	0.89	1.00	0.94
2	Achillea millefolium	9	0	520	0	1.00	1.00	1.00	1.00
3	Agrimonia eupatoria	4	0	525	0	1.00	1.00	1.00	1.00
4	Ajuga iva	7	0	522	0	1.00	1.00	1.00	1.00
5	Allium polyanthum	4	0	523	2	1.00	1.00	0.67	0.80
6	Allium triquetrum	6	1	522	0	1.00	0.86	1.00	0.92
7	Ambrosia artemisiifolia	3	1	525	0	1.00	0.75	1.00	0.86
8	Anchusa italica	3	0	526	0	1.00	1.00	1.00	1.00
9	Anemone coronaria	4	3	522	0	0.99	0.57	1.00	0.73
10	Anthyllis vulneraria	5	0	524	0	1.00	1.00	1.00	1.00
11	Antirrhinum majus	1	0	528	0	1.00	1.00	1.00	1.00
12	Armeria arenaria	6	0	522	1	1.00	1.00	0.86	0.92
13	Asphodelus ramosus	0	0	528	1	1.00	0.00	0.00	0.00
14	Asplenium scolopendrium	2	0	527	0	1.00	1.00	1.00	1.00
15	Astrantia major	5	1	523	0	1.00	0.83	1.00	0.91
16	Bellevalia romana	2	0	527	0	1.0	1.0	1.0	1.0
17	Betula pubescens	2	2	525	0	1.00	0.50	1.00	0.67
18	Bistorta officinalis	3	1	525	0	1.00	0.75	1.00	0.86
19	Buglossoides purpurocaerulea	5	0	523	1	1.00	1.00	0.83	0.91
20	Calendula arvensis	5	0	524	0	1.00	1.00	1.00	1.00

TABLE II: Comparison of Parameter Size, Average Accuracy, Loss and Time of Various Models without Pre-Trained Weights

Model	Total Parameters	Trainable Parameters	Average Accuracy	Average Loss	Average Time (s)
ResNet50	2,57,01,263	2,57,01,263	0.8649	0.1351	196.5
Inception	2,39,16,335	2,39,16,335	0.8824	0.1176	208.5
MobileNet	35,85,103	35,85,103	0.8839	0.1161	167.3
Xception	2,29,75,031	2,29,75,031	0.882	0.118	241.2
EfficientNet	1,95,25,230	1,95,25,230	0.8773	0.1227	50.5
Proposed Model	2,40,719	2,40,719	0.98	0.0615	61.5

TABLE III: Comparison of Parameter Size, Average Accuracy, Loss and Time of Various Models with Pre-Trained Weights

Model	Total Parameters	Trainable Parameters	Non-trainable Parameters	Average Accuracy	Average Loss	Average Time (s)
ResNet50	2,57,01,263	21,13,551	2,35,87,712	0.8546	0.1652	98.5
Inception	2,39,16,335	21,13,551	2,18,02,784	0.8925	0.1156	99.5
MobileNet	35,85,103	13,27,119	22,57,984	0.9826	0.0354	38.3
Xception	2,29,75,031	21,13,551	2,08,61,480	0.802	0.1081	40.2
EfficientNet	1,95,25,230	18,51,407	1,76,73,823	0.8817	0.0191	50.5
Proposed Model	2,40,719	1,47,471	93,248	0.9826	0.0076	37.5

highest average accuracy of (0.9826) while maintaining a significantly lower parameter count as (35 Million) as compared to deeper architectures like ResNet50 as (257 Million) and Xception as (229 Million). Additionally, MobileNet achieves this performance with a relatively low training time of 38.3 seconds per epoch. In contrast, Xception has the lowest accuracy of (0.802), despite having a large number of trainable parameters, indicating that its performance might not be optimal in this setting. EfficientNet, on the other hand, achieves a strong accuracy of 0.8817 with the lowest loss as 0.0191, highlighting its efficiency in balancing performance and parameter count, whereas the proposed model significantly outperforms all traditional models in terms of efficiency and accuracy. It achieves an accuracy of 0.9826, matching with MobileNet, while having only 2,40,719 total parameters a drastic reduction as compared to all other models. The number of trainable parameters is also only 1,47,471, with a minimal number of non-trainable parameters i.e. 93,248. Additionally, it achieves the lowest average loss as 0.0076 and the fastest training time as 37.5 seconds per epoch, making it the most efficient model in comparison. Overall, the results demonstrate that the proposed model maintains state-of-the-art accuracy while being computationally lightweight and highly efficient. This makes it a promising alternative for deployment in environments with limited computational resources, where both accuracy and speed are critical factors.

B. Comparison of Training and Validation Accuracy vs Loss

Fig. 7 illustrates training and validation accuracy vs loss performance of a proposed deep learning model trained from scratch, without using pre-trained weights, over multiple epochs for all 1000 classes of PlantCLEF 2015. The x-axis represents the number of epochs, while the y-axes display two critical metrics: loss on the primary axis (left) and accuracy on the secondary axis (right). These metrics are evaluated for both training and validation datasets, with four distinct curves providing insights into the model's learning behavior. The training loss, represented by the red line, starts at a high value, but decreases steadily as the model learns, indicating effective optimization of parameters. Similarly, the validation loss, shown in blue, follows a declining trend, albeit with fluctuations in the initial epochs. These fluctuations suggest early instability as the model adjusts its weights, but the loss eventually stabilizes at a low value, indicating improved generalization. The training accuracy, depicted by the purple line, increases rapidly and eventually plateaus close to 1.0, demonstrating that the model has effectively learned to classify the training data. Mean while, the validation accuracy, represented by the green line, exhibits some fluctuations in the early epochs before stabilizing at a high level. These fluctuations are common during the initial training phase as the model fine-tunes its internal parameters, but the convergence of validation accuracy with training accuracy suggests strong generalization capability. The model does not appear to suffer from significant over-fitting, as evidenced by the minimal gap between training and validation accuracy. If over-fitting were an issue, the validation loss would increase, and validation accuracy would stagnate or decline, diverging from the training metrics. Similarly, if the model were under-fitting, both training and validation accuracy would remain low, indicating inadequate learning. During the early learning phase, spanning the first 20 to 30 epochs, both training and validation loss decrease significantly while accuracy increases, demonstrating effective learning. In the mid-to-late training phase, beyond 50 epochs, the training loss stabilizes, suggesting that the model has nearly reached an optimal configuration of weights. Validation accuracy also remains stable, further confirming the model's ability to generalize well to unseen data. Given the observed trends, further fine-tuning could be beneficial to enhance stability and performance. Overall, the graph in Fig. 7 demonstrates that the model, despite being trained from scratch without pre-trained weights, has successfully learned patterns from the dataset while maintaining strong generalization capabilities.

Fig. 8 represents the training and validation accuracy vs loss performance of the proposed deep learning model utilizing pre-trained weights. The x-axis denotes the number of epochs, while the primary y-axis (left) measures the loss, and the secondary y-axis (right) tracks the accuracy. Four distinct curves illustrate the model's learning behavior: training loss (red), validation loss (blue), training accuracy (purple), and validation accuracy (green). The use of pre-trained weights accelerates convergence, as seen in the rapid decline in both training and validation loss within the

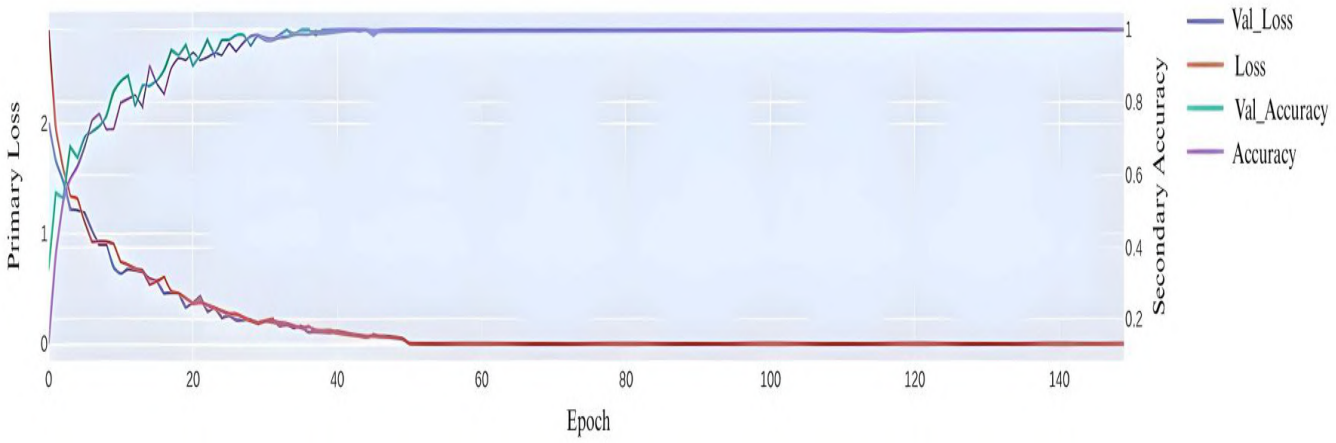

Fig. 7: Comparison of Training and Validation Accuracy vs Loss of Proposed Model without Pre-Trained Weights

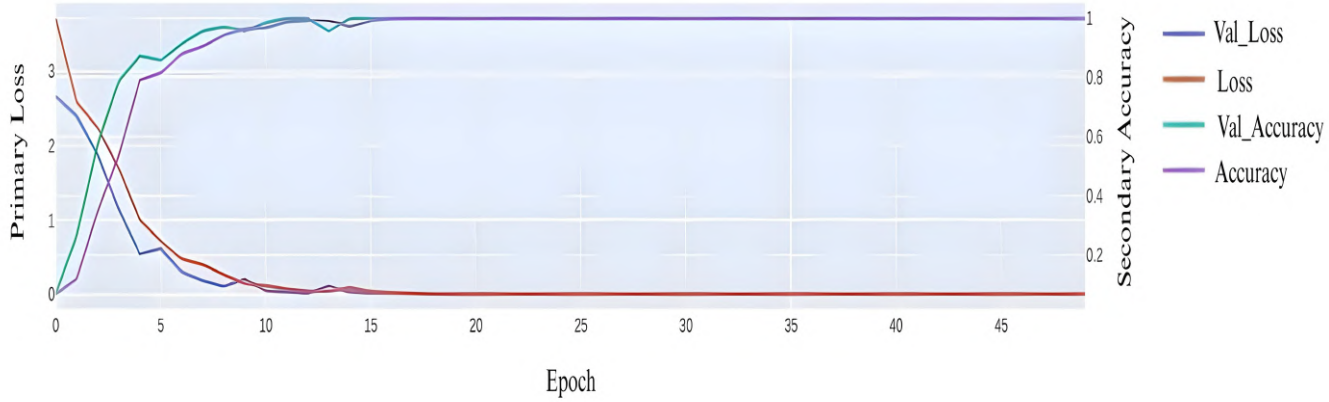

Fig. 8: Comparison of Training and Validation Accuracy vs Loss of Proposed Model with Pre-Trained Weights

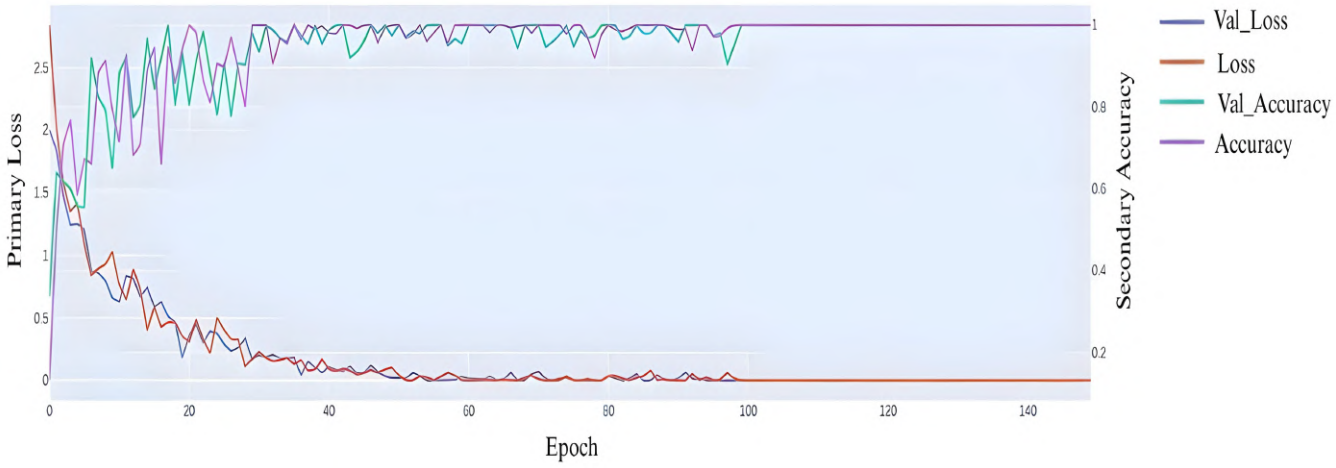

Fig. 9: Training and Validation Performance of the Proposed Deep Transfer Learning Model for Cluster-3

first few epochs. Unlike training from scratch, where the model takes longer to learn meaningful features, pre-trained weights provide a strong initialization, allowing the model to achieve high accuracy more quickly. At the beginning of training, the training loss starts at a higher value, but decreases sharply, indicating effective optimization. Similarly, the validation loss follows a step downward trajectory before stabilizing at a low value, demonstrating strong generalization capabilities. The early fluctuations in validation loss are expected, as the model fine-tunes itself to the PlantCLEF 2015 dataset. These fluctuations diminish as training progresses, suggesting that the model has adapted well to the new task without significant over-fitting. The training accuracy rises rapidly, reaching near 1.0 within a

few epochs, confirming that the model quickly learns to classify the training data with high confidence. The validation accuracy follows a similar trend, showing a consistent increase before plateauing at a high level. The close alignment between training and validation accuracy indicates minimal over-fitting and effective knowledge transfer from the pre-trained model. Compared to the proposed model trained from scratch, this transfer learning approach achieves optimal performance in a fewer epochs. The rapid stabilization of both accuracy and loss highlights the advantages of transfer learning, where the model leverages previously learned features from large-scale ImageNet dataset to improve performance on a new, domain specific dataset i.e. PlantCLEF 2015. Since pre-trained models already capture essential low-level and high-level representations, only the final layers need fine-tuning for the new plant classification task. This results in faster convergence, reduced computational cost, and improved generalization. Overall, the graph represented in Fig. 8 suggests that the proposed model with pre-trained weights outperforms the non-pre-trained counterpart in terms of training efficiency and accuracy. The lack of significant divergence between training and validation metrics further indicates that the model generalizes well to unseen data.

C. Performance of Classifier for Cluster 1-15

A detailed class-wise analysis revealed that the model performed exceptionally well across most plant species, with precision and recall values exceeding 95% for the majority of classes. However, a few species with subtle morphological differences, such as variations in leaf texture or flower shape, exhibited slightly lower performance. Misclassification were analyzed for 15 clusters using the confusion matrix, which highlighted these challenging cases. It revels that collecting more diverse samples of these challenging species could improve performance. For sample, to show the training and validation performance of the proposed deep learning model across multiple epochs for cluster number 3 is illustrated in Fig. 9. The x-axis represents the number of epochs, while the y-axis on the left denotes the primary loss values, and the y-axis on the right represents accuracy. The training loss (red) and validation loss (blue) exhibit a declining trend over time, indicating that the model is progressively learning and reducing errors. In contrast, the training accuracy (purple) and validation accuracy (green) depict an increasing trend, showcasing the model's ability to generalize well over the PlantCLEF 2015 dataset. However, fluctuations in validation accuracy suggest some level of variance, possibly due to class imbalance or complexity in the PlantCLEF 2015 dataset. The eventual stabilization of both accuracy and loss towards the later epochs signifies convergence, implying that the model has learned meaningful patterns. This visualization provides insights into the learning dynamics of the proposed deep transfer learning classifier for cluster number 3 within the broader classification task involving 15 distinct clusters.

D. Performance Comparison of Deep Learning Models with and without Pre-trained Weights with Respect to Accuracy

Fig. 10 illustrates the training accuracy progress over epochs for six different deep learning models such as ResNet50, Xception, Inception, EfficientNet, MobileNet, and the Proposed pre-trained Model with and without pre-trained weights. Each model is trained under two conditions: without pre-trained weights (red curve) and with pre-trained weights (blue curve). The x-axis represents the number of training epochs, while the y-axis denotes accuracy. From the Fig. 10, it is evident that utilizing pre-trained weights (blue curves) significantly accelerates convergence for all models. This effect is noticeable in all deep learning models, where the pre-trained models reach near-perfect accuracy much faster than their non-pre-trained counterparts. The Inception model, while benefiting from pre-trained weights, exhibits a slower convergence rate when compared to other architectures. The proposed pre-trained model demonstrates exceptional efficiency, achieving rapid convergence in fewer epochs than all other models. Even when trained without pre-trained weights, the proposed pre-trained model reaches high accuracy significantly faster than traditional architectures. This suggests that the model is inherently well-optimized for the classification of PlantCLEF 2015 dataset, requiring fewer training iterations to achieve optimal performance. Furthermore, the gap between red and blue curves in models like MobileNet and ResNet50 highlights that pre-training provides a crucial advantage in early training stages, enabling faster feature learning and improved accuracy. In contrast, for the proposed pre-trained model, the difference between pre-trained and non-pre-trained versions is minimal after a certain number of epochs, suggesting that it is highly effective even without external knowledge transfer. Overall, these results emphasize the importance of pre-training in deep learning models, as it significantly boosts convergence speed and final accuracy. Moreover, the exceptional performance of the proposed model in both conditions showcases its superiority in terms of learning efficiency and generalization capability.

E. Performance Comparison of Deep Learning Models with and without Pre-trained Weights with Respect to Loss

Fig. 11 illustrates the comparative training loss curves of six different deep learning models considered in this research which are evaluated with and without using pretrained weights from ImageNet dataset with 1000 classes. Across all models, it is evident that the use of pretrained networks significantly accelerates the convergence of loss and enhances overall training efficiency. Specifically, models initialized with pretrained weights (depicted in blue) demonstrate a steeper and more consistent decline in loss during the early epochs, achieving convergence well before their non-pretrained counterparts (depicted in red). This trend is particularly prominent in architectures such as EfficientNet, Xception, MobileNet, and proposed model where pretrained models reach near-zero loss within the first 50 to 75 epochs, whereas non-pretrained models require more epochs and show relatively slower learning behavior. Notably, the proposed model outperforms all other architectures in both scenarios, achieving rapid convergence and maintaining minimal loss throughout the training process. The pretrained version of the proposed model exhibits the best overall performance, validating the effectiveness of its architectural design and the benefits of leveraging transfer learning. These results clearly demonstrate that the integration of pretrained weights not

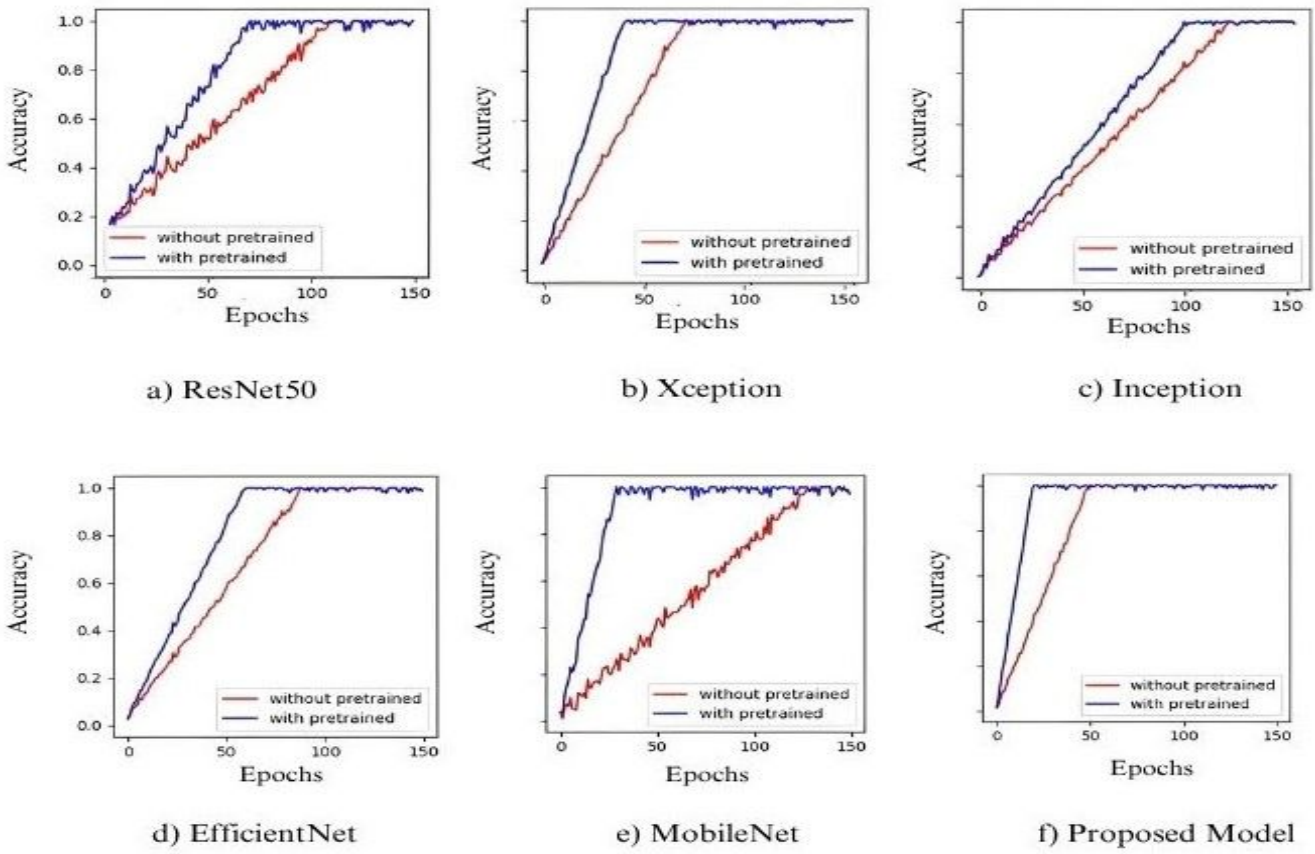

Fig. 10: Performance Comparison of Deep Learning Models with and without Pre-Trained Weights with Respect to Accuracy

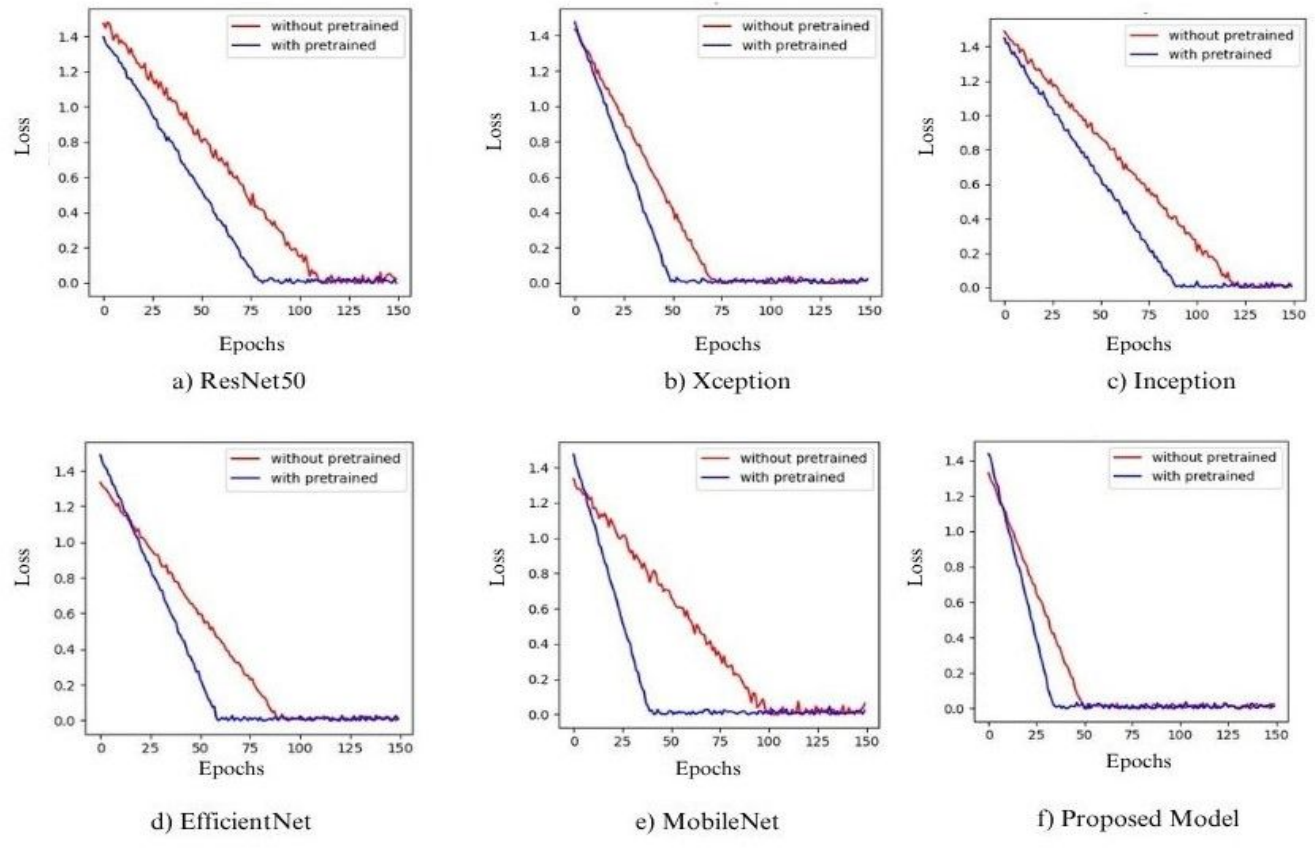

Fig. 11: Performance Comparison of Deep Learning Models with and without Pre-Trained Weights with Respect to Loss

only reduces training time but also improves model generalization and stability, making it a highly advantageous strategy in deep learning-based classification tasks.

F. Comparative Analysis of Deep Transfer Learning Models

The comparative analysis of deep learning models with and without pre-trained weights is presented in Fig. 12, illustrating key performance metrics such as accuracy, precision, recall, loss, F1-score, and training time. The models included in this evaluation are ResNet50, InceptionV3, Xception, EfficientNet, MobileNet, and the Proposed Model. The results highlight the efficiency and effectiveness of different architectures under various training conditions. The accuracy comparison in Fig. 12(a) indicates that models with pre-trained weights generally perform little better than those trained from scratch. Among them, MobileNet and the proposed model achieve the highest accuracy, demonstrating their ability to generalize well to the PlantCLEF 2015 dataset. The proposed deep transfer learning model, in particular, exhibits a strong performance even without pre-trained weights, emphasizing its robustness in learning feature representations. In terms of precision shown in Fig. 12(b), pre-trained models tend to show slightly higher values than their non-pre-trained counterparts. MobileNet and the proposed deep transfer learning model achieve the highest precision, indicating their ability to reduce false positives while maintaining correct classifications. This suggests that these models are particularly effective in distinguishing between different classes with minimal misclassification errors. The recall values shown in Fig. 12(c) follow a similar trend, with pre-trained models consistently achieving higher recall scores. The proposed deep transfer learning model maintains a strong recall score, demonstrating

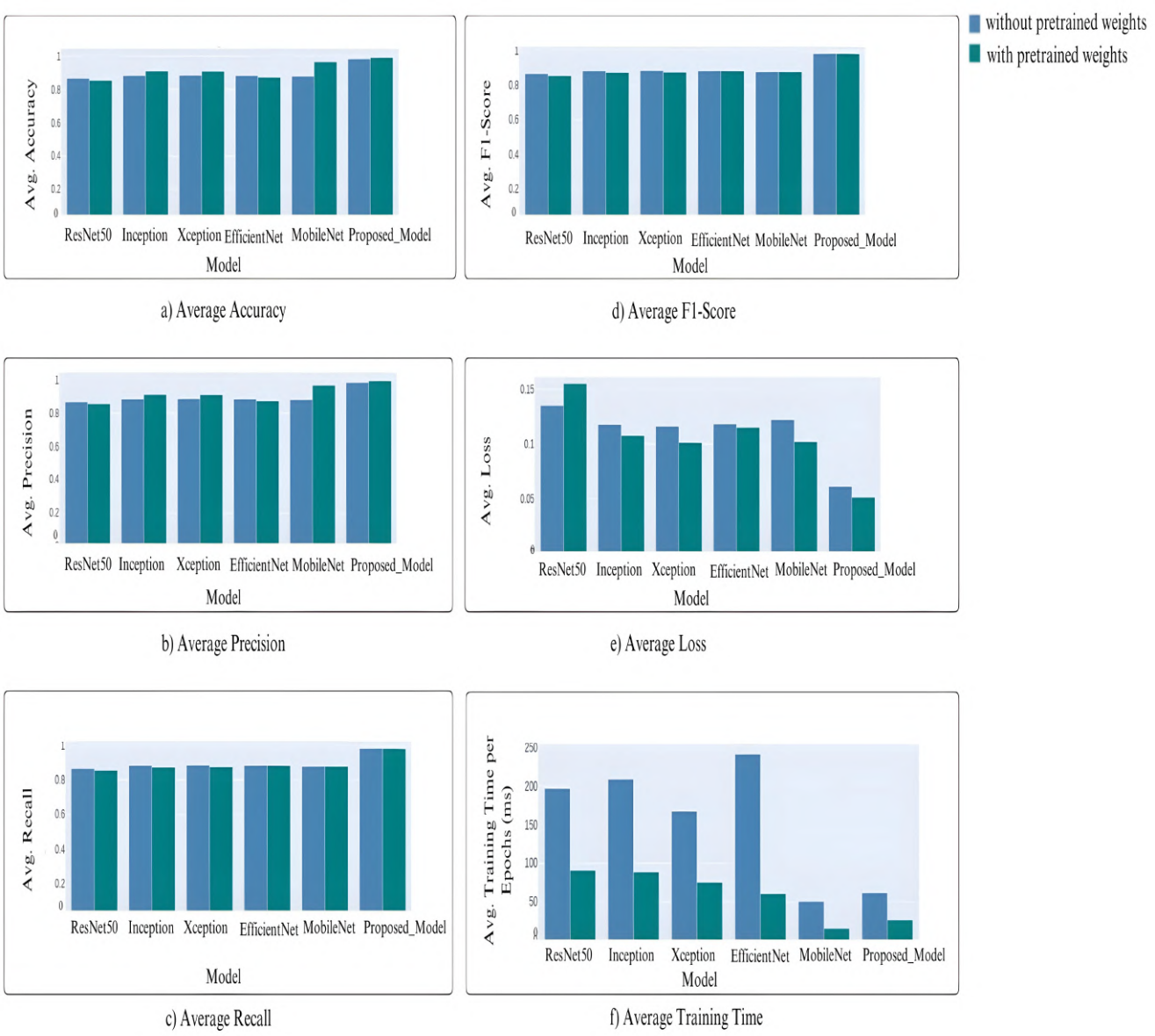

Fig. 12: Performance Comparison of Deep Transfer Learning Models with Respect to Accuracy, Precision, Recall, F1 Score, Loss, and Training Time per Epochs

its capability to correctly identify relevant instances with minimal false negatives. The F1-score given in Fig. 12(d), which balances precision and recall, further reinforces the trends observed in previous metrics. The proposed deep transfer learning model consistently achieves the highest F1-score, making it a well-balanced and robust choice for classification tasks. The improvements in F1-score for pre-trained models highlight the effectiveness of transfer learning in refining predictions. When evaluating average loss shown in Fig. 12(e), it is evident that the proposed deep transfer learning model achieves the lowest loss value, outperforming other models significantly, especially when using pre-trained weights. A lower loss indicates better generalization, meaning the model has effectively learned useful patterns in the PlantCLEF 2015 dataset without over-fitting. MobileNet also exhibits competitive performance, with lower loss values compared to more complex architectures like ResNet50 and Xception. As training efficiency is more crucial one, average training time per epoch is used for comparison and is shown in Fig. 12(f). Heavier models like ResNet50, InceptionV3, and EfficientNet require significantly longer training times, making them computationally expensive. In contrast, the proposed pre-trained model demonstrates the shortest training time while maintaining high accuracy, precision, and recall, making it the most computationally efficient model in this comparison. This efficiency makes it particularly suitable for applications requiring real-time or resource-constrained deployment. While pre-trained models generally exhibit better results, the proposed deep learning model achieves competitive performance even without pre-trained weights.

V. Conclusion and Future Work

In this work, a unique method for classifying plant species by combining ResNet50, k-means clustering, and transfer learning using cascaded CNN architecture has been crafted. PlantCLEF 2015 dataset has been used to asses the performance of the proposed model. When compared to conventional pre-trained deep learning models such as ResNet50, Inception, MobileNet, Xception, and EfficientNet, the proposed pre-trained CNN model has superior performance with respect to standard performance metrics. Particularly, the experimental findings show that the suggested model performs noticeably better than conventional pre-trained techniques with an average accuracy of 98.26%, average time of 37.5 seconds, average loss of 0.0076, and less number of trainable and non-trainable parameters. A notable improvement is observed in training time, parameter size and stability on pre-trained deep learning models when compared with non-pre-trained one. To conclude, it can be stated that the comparative evaluation emphasize the superior performance of the proposed deep transfer learning model, which achieves high accuracy, low loss, and fast training times as compared to traditional deep transfer learning models. To improve the model's performance, other data modalities such as environmental metadata and hyper-spectral photography can be incorporated in further work. Future studies could also examine methods for real-time classification in mobile applications, which would allow for greater accessibility and useful applications in conservation and agriculture.

References

[1] R. Zhang, Y. Zhu, Z. Ge, H. Mu, D. Qi, and H. Ni, "Transfer learning for leaf small dataset using improved ResNet50 network with mixed activation functions," Forests, vol. 13, no. 12, pp. 1-21, 2022.

[2] P. A. S. Rani and N. S. Singh, "Paddy leaf symptom-based disease classification using deep cnn with ResNet50," International Journal of Advanced Science Computing and Engineering, vol. 4, no. 2, pp. 88–94, 2022.

[3] T. W. Harjanti, H. Setiyani, J. Trianto, and Y. Rahmanto, "Classification of mint leaf types based on the image using euclidean distance and k-means clustering with shape and texture feature extraction," Tech-E, vol. 5, no. 2, pp. 115–124, 2022.

[4] H. Zhang, P. Yanne, and S. Liang, "Plant species classification using leaf shape and texture," in Proceedings of International Conference on Industrial Control and Electronics Engineering, pp. 2025–2028, 2012.

[5] P. Pungki, C. Atika Sari, D. R. Ignatius Moses Setiadi, and E. Hari Rachmawanto, "Classification of plant types based on leaf image using the artificial neural network method," in Proceedings of International Seminar on Application for Technology of Information and Communication (iSemantic), pp. 67–72, 2020.

[6] A. A. Gomaa and Y. M. Abd El-Latif, "Early prediction of plant diseases using CNN and GANs," International Journal of Advanced Computer Science and Applications, vol. 12, no. 5, pp. 514-519, 2021.

[7] P. B R and L. P, "Deep learning model for plant species classification using leaf vein features," in Proceedings of International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), pp. 238–243, 2022.

[8] B. Dudi, V. Rajesh, and G. Prasanna Kumar, "Plant leaf classification through deep feature fusion with bidirectional long short-term memory," in Proceedings of International Conference on Innovations in Science and Technology for Sustainable Development (ICISTSD), pp. 68–73, 2022.

[9] W. Tian, L. Tang, Y. Chen, Z. Li, S. Qiu, X. Li, J. Zhu, C. Jiang, P. Hu, J. Jia, H. Wu, L. Chen, and J. Hyyppa, "Plant species classification using hyper-spectral lidar with convolutional neural network," in Proceedings of IGARSS IEEE International Geoscience and Remote Sensing Symposium, pp. 1740–1743, 2022.

[10] J. w. Tan, S.W. Chang, S. Abdul-Kareem, H. J. Yap, and K.T. Yong, "Deep learning for plant species classification using leaf vein morphometric," IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 17, no. 1, pp. 82–90, 2020.

[11] A. Karnan and R. Ragupathy, "A comprehensive study on plant classification using machine learning models," Lecture Notes in Networks and Systems, vol. 10, pp. 187–199, 2024.

[12] S. T. Kakileti, G. Manjunath, and H. J. Madhu, "Cascaded CNN for view independent breast segmentation in thermal images," in Proceedings of 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, pp. 6294–6297, 2019.

[13] Z. Zhu, K. Lin, A. K. Jain, and J. Zhou, "Transfer learning in deep reinforcement learning: A survey," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 11, pp. 13 344–13 362, Nov 2023.

[14] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He, "A comprehensive survey on transfer learning," in Proceedings of the IEEE, vol. 109, no. 1, pp. 43–76, Jan 2021.

[15] Z. Zhai, B. Liu, H. Xu, and P. Jia, "Clustering product features for opinion mining," in Proceedings of the fourth ACM International Conference on Web Search and Data Mining, pp. 347–354, 2011.

[16] D. M. Farid, A. Nowe, and B. Manderick, "A feature grouping method for ensemble clustering of high-dimensional genomic big data," in Proceedings of Future Technologies Conference (FTC). IEEE, pp. 260–268, 2016.

[17] W. Wang, Y. He, L. Ma, and J. Z. Huang, "Latent feature group learning for high-dimensional data clustering," Information, vol. 10, no. 6, pp. 1-16, 2019.

[18] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, and L. Van Gool, "Weakly supervised cascaded convolutional networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 914–922, 2017.

[19] Z. Cai and N. Vasconcelos, "Cascade R-CNN: High quality object detection and instance segmentation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 43, no. 5, pp. 1483–1498, 2019.

[20] S. A. Ali Ahmed, B. Yanıkoğlu, and E. Aptoula, "Plant identification with large number of classes: Sabanciu-Gebzetu system in PlantCLEF 2017," in Proceedings of CEUR Workshop, vol. 1866, pp. 1-5, 2017.

[21] S. A. Ali Ahmed, "Deep learning ensembles for image understanding," Ph.D. dissertation, 2021.

[22] S. Atito, B. Yanikoglu, E. Aptoula, I. Ganiyusufoglu, A. Yıldız, K. Yıldırır, B. Sevilmis, and M. U. Sen, "Plant identification with deep learning ensembles in ExpertLifeCLEF 2018," in Proceedings of CEUR Workshop, vol. 2125, pp. 1-6, 2018.
[23] A. Ramdan, A. Heryana, A. Arisal, R. B. S. Kusumo, and H. F. Pardede, "Transfer learning and fine-tuning for deep learning-based tea diseases detection on small datasets," in Proceedings of International Conference on Radar, Antenna, Microwave, Electronics, and Telecommunications (ICRAMET), pp. 206–211, 2020.
[24] H. Goeau, P. Bonnet, and A. Joly, "Plant identification in an open-world (lifeclef 2016)," in Proceedings of CLEF: Conference and Labs of the Evaluation Forum, no. 1609, pp. 428–439, 2016.
[25] M. Sulc and J. Matas, "Fine-grained recognition of plants from images," Plant Methods, vol. 13, no. 12, pp. 1–14, 2017.
[26] M. Bello, G. Nápoles, R. Sánchez, R. Bello, and K. Vanhoof, "Deep neural network to extract high-level features and labels in multi-label classification problems," Neurocomputing, vol. 413, pp. 259–270, 2020.
[27] A. Karnan and R. Ragupathy, "A review of plant classification using deep learning models," Lecture Notes in Networks and Systems, vol. 10, pp. 113–125, Springer, 2024.
[28] S. Jia and Y. Tian, "Face detection based on improved multi-task cascaded convolutional neural networks," IAENG International Journal of Computer Science, vol. 51, no. 2, pp. 67-74, 2024.
[29] A. K. Pandey, D. Jain, T. K. Gautam, J. S. Kushwah, S. Shrivastava, R. Sharma, and P. Vats, "Tomato leaf disease detection using generative adversarial network-based ResNet50V2," Engineering Letters, vol. 32, no. 5, pp. 965-973, 2024.
[30] A. Joly, H. Goeau, H. Glotin, C. Spampinato, P. Bonnet,W.P. Vellinga, R. Planque, A. Rauber, S. Palazzo, B. Fisher, et al., "LifeCLEF 2015:Multimedia life species identification challenges," in Proceedings of Experimental IR Meets Multilinguality, Multimodality, and Interaction: 6th International Conference of the CLEF Association, CLEF15, Toulouse, France, September 8-11, 2015, Proceedings 6, pp. 462–483, Springer, 2015.
[31] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 4, pp. 491–502, 2005.
[32] Y. Y. Ding and L. Wang, "Research on the application of improved attention mechanism in image classification and object detection," IAENG International Journal of Computer Science, vol. 50, no. 4, pp. 1174–1182, 2023.