Innovative Informatics and Artificial Intelligence Research (IIAIR)
Vol. 2, issue 1, pp. 17-24, 2026
Received 10 Apr 2026; accepted 28 Apr 2026; published 30 Apr 2026
https://doi.org/10.35718/iiair.v2i1.8481959

Pretrained Deep Learning Models for Classifying Plant Species: A Comparative Study in Balikpapan Botanical Gardens

Dimas Dewanto[1], Rizky Amelia[1], and Rufai Yusuf Zakari[2]
[1]Department of Informatics, Institut Teknologi Kalimantan, Balikpapan, Indonesia
[2]School of Digital Science, Universiti Brunei Darussalam, Brunei Darussalam

Corresponding author: *Rizky Amelia (rizky.amelia@lecturer.itk.ac.id)*

To cite this article: D. Dewanto, R. Amalia, R. Y. Zakari, "Pretrained Deep Learning Models for Classifying Plant Species: A Comparative Study in Balikpapan Botanical Gardens" *Innovative Informatics and Artificial Intelligence Research*, vol. 2, issue 1, 2026. [Online]. Available: https://doi.org/10.35718/iiair.v2i1.8481959

Abstract
Urban plant biodiversity is crucial for maintaining ecosystem functioning, yet it faces numerous serious threats such as infrastructure development, habitat fragmentation, pollution, and climate change. Species inventory is also challenging due to limited data and changes in scientific nomenclature. To address this, various conservation efforts are being carried out, all of which require accurate species identification. However, manual classification is often inefficient and inaccurate due to similarities between species as well as variations within a single species. Therefore, automated approaches based on machine learning, such as Convolutional Neural Networks (CNN) are used to assist plant classification, although challenges remain, including data imbalance and high image variability. Various CNN architectures have been developed, and no single model outperforms others in all situations. Therefore, the choice must be tailored to the characteristics of the task and the dataset being used. This study aims to compare the performance of several pretrained models and determine the most optimal one for classifying plant species in natural images from the Balikpapan Botanical Garden (BBG52). The focus of this research is to evaluate the performance of four CNN architectures (ResNet50, DenseNet121, MobileNetV3 Large, and ConvNeXt Tiny), all of which are pretrained models with weights obtained from training on the ImageNet dataset. This study employs two configurations (without additional hidden layers and with the addition of one hidden layer) to examine the effect of the number of hidden layers on model performance, and evaluates them based on accuracy, F1-score, and computation time. The results of this study show that ConvNeXt Tiny with the added hidden-layer configuration is the most optimal model for plant species classification on the BBG52 dataset, achieving the highest accuracy (94.66%), the highest F1-score (0.946), and the fastest computation time (0.0098 seconds) using a GPU. These findings provide practical guidance for selecting efficient and high-performing CNN architectures for real-world plant species classification in biodiversity conservation contexts.

Keywords: Plant Species Classification; Comparative Deep Learning Models; Convolutional Neural Networks (CNN); Balikpapan Botanical Garden.

1. Introduction
Urban biodiversity, particularly plants in green spaces, plays a vital role in maintaining ecosystem function, mitigating climate change, and improving human physical and mental well-being. However, this biodiversity is threatened by infrastructure development, pollution, and weak green space governance, which trigger habitat degradation [1] . Conservation challenges are further compounded by obstacles to research and documentation, such as data uncertainty and taxonomic changes, which can put species at risk of extinction before they can be properly identified and protected [2] .

As a first step in conservation, accurate species identification is essential. However, traditional methods relying on manual morphological observations are considered inefficient and time-consuming. These methods often struggle with fine-grained classification due to visual variation within a species or similarities between different species. Therefore, automated approaches using machine learning and image processing are a necessary solution, although their implementation still faces technical challenges such as data imbalance and high visual variation in images, which affect model generalization [3] .

Several studies have demonstrated the effectiveness of CNN-based methods for plant species classification while also highlighting the context-dependent nature of model performance. For instance, Araújo et al. proposed a fine-grained classification approach based on a two-view leaf representation and hierarchical strategy based on botanical taxonomy. Siamese Convolutional Neural Network (S-CNN) method is used to measure image similarity through global features (shape and color) at genus level, and local features

(texture and vein) at species level. This approach effectively reduces dependence on large training data and supports scalability of new species without the need for retraining. Based on testing, this architecture achieves an accuracy of 0.87 on PlantCLEF 2015 dataset and 0.96 on LeafSnap [3] .

Evaluation of the models used should be based on the No-Free-Lunch (NFL) principle, which is often misinterpreted as the claim that all machine learning algorithms are equal and lack formal justification. However, this interpretation ignores the crucial distinction between data-only and model-dependent algorithms. The NFL theorem strictly applies to data-only algorithms, that is, algorithms that do not encapsulate explicit inductive biases regarding the data structure. In practice, standard algorithms such as Empirical Risk Minimization (ERM) do not operate solely on data but rely on the selection of models or classes of hypotheses that represent specific structural assumptions. Therefore, the effectiveness of an algorithm is relative and is largely determined by the match between the model's inductive biases (e.g., network architecture or representation complexity) and the underlying regularities of the processed image data. Thus, the NFL theorem does not negate the value of commonly used learning algorithms, but rather emphasizes that the justification for their performance is local and conditioned on the selection of architectures and parameters appropriate to the characteristics of the data and the context of the image classification task at hand [4] .

This study aims to conduct a comparative analysis of the performance of several pretrained models in plant species classification on the BBG52 dataset using accuracy and F1-score metrics, and to analyze the effect of adding hidden layers on model performance. In summary, the main contributions of this study are as follows:

1. Comparing the performance of pretrained models for plant species classification on the BBG52 dataset based on accuracy and F1-score, and the effect of adding hidden layers.
2. Comparing the computation time of pretrained models based on the number of parameters, and the effect of adding hidden layers.

Determining the most optimal model for plant species classification on the BBG52 dataset based on performance and computation time.

2. Related Works

Previous studies that discuss the problem of image-based plant species classification, it can be concluded that deep learning, especially the Convolutional Neural Networks (CNN) and transfer learning approaches, consistently show superior performance in overcoming the complexity of plant morphological variations in natural images. Ramadhani et al. [5] in their study introduced a new BBG52 dataset, which contains 5200 natural images of 52 different plant species captured using mobile devices. ResNet variants (ResNet-34, ResNet-50, and ResNet-101) were used as pretrained models for plant species classification. All models were tested by varying the number of hidden layers, namely without hidden layers, one hidden layer, and two hidden layers. The results showed that ResNet-50 without hidden layers achieved the best performance with an accuracy of 96.88% and an F1-score of 0.9689, while the addition of one or two hidden layers tended to decrease the performance of all ResNet variants. In addition, the manual split method produces better performance, shown by ResNet-34 without hidden layers with an accuracy of 95.12% and an F1-score of 0.9512, and in terms of computational efficiency, ResNet-34 is the fastest model with a computational time of 0.0765 seconds compared to ResNet-50 (0.0861 seconds) and ResNet-101 (0.0916 seconds) on GPU devices.

Salsabila et al. [6] focused on identifying 3,500 images of medicinal plants using Deep Learning CNN Transfer Learning models such as MobileNet, VGG16, DenseNet121, ResNet50V2, and NASNetMobile. The dataset used was the "Indonesian Herb Leaf Dataset 3,500" which consists of 10 classes of medicinal plants. This study compared the performance of the five models with data sharing schemes of 80:10:10 and 70:20:10, and compared augmented and unaugmented data. The results showed that MobileNet provided the best performance with accuracy, precision, recall, and F1-score of 98.86%.

A study Bui et al. [7] compared leaf image classification using deep artificial neural networks (DNNs) and manual feature extraction methods on public datasets to develop smart solutions in agriculture. Manual feature extraction methods (SIFT, HOG, and DWT) and CNN models (AlexNet, ResNet-50, Inception-v3, and DenseNet-201) were optimized and applied to accurately classify leaf images. The results showed the highest classification accuracy of 94% on the grayscale Plant Village dataset, 97.78% on the color Plant Village dataset, and 95.53% on apple leaf disease images using DenseNet-201. This study concluded that DNNs are more efficient and robust in leaf image classification compared to manual feature extraction methods.

Dey et al. [8] applied seven Deep Convolutional Neural Network (DCNN) architectures, namely VGG16, VGG19, DenseNet201, ResNet50V2, Xception, InceptionResNetV2, and InceptionV3 for automatic identification of 30 medicinal plant species from leaf images. Model training was carried out using two dataset scenarios: public data (P_1) with a uniform background and a combination of public and field data (PF_1) with a complex background. All images were preprocessed by scaling the size to 224×224 pixels. The models were evaluated using accuracy, precision, recall, and F1-score metrics, and normalized by the leverage factor ($\gamma\omega$) for comprehensive comparison. The evaluation results showed that DenseNet201 recorded the best performance in both dataset scenarios, with an accuracy of 99.64% and a precision of 98.31% on the P_1 data, and an accuracy of 97% on the PF_1 data. This model also had the highest $\gamma\omega$ values (0.19 for P_1 and 0.15 for PF_1), indicating superior stability and reliability compared to other models. These findings confirm that DenseNet201 is the most effective architecture for medicinal plant species identification, even under varied and complex image backgrounds.

Zhou et al. [9] examined various deep learning approaches for plant image classification comprehensively evaluated using five plant image datasets consisting of four public datasets (Flavia, Swedish Leaf, Oxford Flower102, and D-leaf) and one custom-built Camellia@clab dataset specifically for cultivar-level testing. Six DL models (LeNet, AlexNet, ResNet, InceptionNet, DenseNet, and MobileNet) were tested using 10-fold cross-validation to assess prediction accuracy and classification loss. The results showed that DenseNet provided the best stable performance across all datasets, with a median accuracy of over 90%, while LeNet also showed good and stable performance with a median

accuracy above 82.3%. Overall, most models achieved a median classification accuracy above 62%, with DenseNet achieving nearly 100% accuracy. The study also highlighted that datasets with a larger number of images per species tend to improve the classification accuracy of the models.

Mehdipour Ghazi et al. [10] applied a transfer learning method by utilizing three pretrained deep convolutional neural network (CNN) architectures, namely AlexNet, GoogLeNet, and VGGNet, for plant species identification using the LifeCLEF 2015 dataset. To improve performance and reduce overfitting, data augmentation was performed through image transformations such as rotation, translation, reflection, and scaling, as well as optimization of parameters such as the number of iterations, batch size, and the number of extracted patches. The results showed that fine-tuning the pretrained models significantly outperformed training from scratch, with VGGNet achieving the highest validation accuracy of 78.44% and GoogLeNet at 76.87%. Through score-based fusion of the two best models, the combined system successfully achieved an overall accuracy of 80.18% on the validation set and an inverse rank score of 0.752 on the test set, outperforming the LifeCLEF 2015 winner by 15% points in accuracy. Parameter analysis also revealed that increasing the number of iterations had the most significant impact on performance, followed by data augmentation, while increasing the batch size had a more limited impact.

Hama et al. [11] proposed a deep learning-based ornamental plant leaf classification system by modifying the ResNet-50 architecture. The applied method involved creating a new dataset containing 2,500 images of 10 ornamental plant leaf species, which was then expanded to 12,000 images through data augmentation techniques such as rotation, flipping, zooming, and shifting. The ResNet-50 model was modified by performing hyperparameter fine-tuning and selective layer freezing to reduce the number of parameters and training time. Evaluation results showed that this improved model achieved an accuracy of up to 99% on the augmented dataset and 98.60% on the unaugmented dataset, outperforming the original ResNet-50 model and MobileNet_v2, thus proving the effectiveness of the proposed approach in automatic ornamental plant leaf classification.

Khalid et al. [12] applied transfer learning and fine-tuning methods to a ResNet-50 model pretrained on the ImageNet dataset to classify images of two local Malaysian herbal plant species from a subset of the MYLPherbs-1 dataset. Two main approaches were explored: using the pretrained model as a feature extractor with different classifiers, and fine-tuning by unfreezing certain layers in the ResNet-50 architecture, specifically in Block 1 and Block 3 in Stage 4. Experimental results showed that fine-tuning by unfreezing layers starting from Stage 4 Block 1 produced the best performance, achieving the highest accuracy of 97.08% with an input hyperparameter of 224x224x3 shape, 150 epochs, and a batch size of 64, while using transfer learning as a feature extractor only achieved the lowest accuracy of 88.45%. These findings confirm that deep layer adaptation through fine-tuning significantly improves the model's ability to recognize specific characteristics of the herbal plant dataset, with increasing the number of epochs also contributing positively to classification accuracy.

Gustineli et al. [13] proposed a transfer learning approach utilizing a self-supervised trained Vision Transformer (DINOv2) model to handle multi-label classification of plant species on the PlantCLEF 2024 dataset. The proposed method involves feature embedding extraction using both baseline and fine-tuned DINOv2 models, followed by linear classifier training on the embedding. To address the large data scale, a distributed processing pipeline with Apache Spark is implemented that includes image preprocessing (cropping and resizing to 128×128 pixels), conversion to Parquet format, and efficient feature extraction. In the inference stage, two strategies are applied: full-image prediction and a grid-based approach by dividing the image into tiles that are then classified independently. Experimental results show that the fine-tuned DINOv2 model combined with grid-based inference (3×3) and argmax aggregation per tile yields the best performance, with significant improvements in the Macro F1 Averaged Per Plot (17.76) and Macro F1 Averaged Per Species (44.76) metrics compared to the full-image approach. These findings confirm the effectiveness of utilizing the fine-tuned DINOv2 embedding and grid processing strategies in handling the complexity of multi-label classification on large-scale vegetation plot images.

Araújo et al. [3] proposed a new method for fine-grained plant species classification using a two-view leaf image representation and a hierarchical classification strategy. This method uses botanical taxonomy as the basis for a coarse-to-fine strategy in identifying plant genera and species. The two-view representation provides complementary global and local features of leaf images, and a deep metric based on Siamese Convolutional Neural Networks (S-CNN) is used to reduce the dependence on many training samples and make the method scalable to unknown plant species. Experimental results on the PlantCLEF 2015 and LeafSnap datasets demonstrate the effectiveness of the proposed method, achieving recognition accuracies of 0.87 and 0.96, respectively.

Lapkovskis et al. [14] proposed an innovative multimodal deep learning approach for plant identification by applying automatic fusion using the Multimodal Fusion Architecture Search (MFAS) algorithm. This method integrates four plant organs (flower, leaf, fruit, and stem) as fixed input modalities, which are processed through a pretrained unimodal model based on MobileNetV3Small. To overcome the limitations of existing multimodal datasets, the authors developed a preprocessing pipeline that transforms the PlantCLEF2015 dataset into Multimodal-PlantCLEF, consisting of 979 classes. The architecture fusion process is performed automatically by MFAS to determine the optimal fusion points and strategies between modalities, supported by the multimodal dropout (MD) technique to improve the model's robustness to missing modalities. Evaluation results show that the proposed model achieves an accuracy of 82.61%, outperforming the baseline late fusion by 10.33% and other state-of-the-art methods on the same dataset. Furthermore, the MD model demonstrated significant robustness when tested on an incomplete subset of modalities, with consistent performance improvements compared to the baseline. The statistical superiority of this model was also confirmed by McNemar's test ($p < 0.001$), further confirming the validity of the results.

Mareta et al. [15] in their research used the TensorFlow-based Convolutional Neural Network (CNN) method to classify three types of ornamental plants, namely Aglonema, Coleus, and Croton. The data used consisted of 300 images of

ornamental plants divided into training and testing data. The implemented CNN has three feature extraction layers that include convolution, pooling, and fully connected, with several convolutional layers whose filters increase from 16 to 96 kernels, resulting in 55,392 final parameters. The model was trained for 50 epochs. The results showed a high level of accuracy, namely 98.30% for training and 98.75% for testing. Specifically, the testing accuracy for Aglonema was 92.16%, Coleus 97.25%, and Croton 92.94%, which indicates the effectiveness of this method in automatically recognizing ornamental plant types.

In summary, the problem of plant species classification in natural images is most effectively solved by utilizing pretrained CNN models with appropriate architectural configurations, selecting models that balance performance and computational efficiency, and supporting adequate datasets. These findings emphasize that there is no single, universally superior model; rather, model effectiveness depends heavily on dataset characteristics, training scenarios, and application objectives.

3. Methods

3.1 ResNet50

One of the models used in this study is ResNet model, which is a residual convolutional neural network (CNN) architecture that introduces skip connections (shortcut connections) to facilitate the flow of information for image recognition. This model has the ability to train very deep networks without degradation problems by preserving the original signal through the residual path [16] .

This model is divided into four main stages with varying numbers of residual blocks, namely 3, 4, 6, and 3 blocks at each stage, and is equipped with shortcut projections at the beginning of each stage when there is a change in spatial dimensions or the number of channels. For the ResNet50 variant, it has 50 layers with a bottleneck structure consisting of 1×1, 3×3, and 1×1 convolutions in sequence, which efficiently reduces the channel dimension before the spatial convolution operation. In this study, the model is used as a reference or basic architecture that will be used to compare other pretrained models [16] .

3.2 DenseNet121

Next is the DenseNet model proposed by Huang et al. [17] , a densely connected convolutional neural network architecture in which each layer is connected to all subsequent layers through feature concatenation. This dense connectivity improves information flow, encourages feature reuse, mitigates the vanishing-gradient problem, and significantly enhances parameter efficiency. Due to these characteristics, DenseNet is less prone to overfitting, even when trained on limited data, and is capable of extracting rich hierarchical features while preserving fine visual details such as leaf texture and vein patterns.

DenseNet121 is a variant of the DenseNet architecture implemented with 121 layers and specifically designed for large-scale image classification such as ImageNet. DenseNet121 consists of four consecutive dense blocks with 6, 12, 24, and 16 convolutional layers, respectively, and is equipped with transition layers between blocks that perform 1×1 convolution and 2×2 average pooling to reduce spatial dimensionality. Each layer in the dense block uses a bottleneck structure (BN-ReLU-Conv(1×1)-BN-ReLU-Conv (3×3)) and a compression factor of θ=0.5 in the transition layer, making it part of the DenseNet-BC family that has proven to be the most efficient in parameter usage. With a growth rate of k=32, each layer adds 32 new feature maps to the network's collective representation [17] .

3.3 MobileNet V3 Larger

Next is the MobileNet model, which is a convolutional neural network architecture built on the foundation of depthwise separable convolution, which is a factorization of standard convolution into two separate layers: depthwise convolution, which applies one filter to each input channel, and pointwise convolution (a 1×1 convolution), which combines the outputs of depthwise convolutions to form new features. This factorization significantly reduces computational complexity and the number of parameters compared to regular convolutions. Each convolutional layer in this architecture is followed by batch normalization and ReLU, except for the final fully-connected layer, which is connected with a softmax activation function for classification purposes [18] .

MobileNetV3 Large was developed through a hybrid approach that combines platform-aware neural architecture search (NAS) and the NetAdapt algorithm to find the optimal global architecture while locally adjusting the number of filters per layer based on real-world device latency targets. The architecture is built on an inverted residual block with a linear bottleneck enriched by a fixed 1:4 Squeeze-and-Excite module and h-swish activations selectively applied to the middle to late layers to balance computational efficiency and accuracy. Early layers were redesigned by reducing the number of first-filter convolutions from 32 to 16 without any loss in accuracy, while late layers were optimized by moving the expansive 1x1 convolutions after global pooling, significantly reducing computational costs. As a result, MobileNetV3 Large achieved 75.2% top-1 accuracy on ImageNet with a latency of only 75 ms at Pixel 1, making it a superior backbone in efficiency and feature representation for classification, detection, and segmentation tasks in resource-constrained environments [19] .

3.4 ConvNeXt Tiny

ConvNeXt was proposed by Liu et al. [20] , is a purely convolutional neural network architecture that modernizes ResNet using design principles inspired by Vision Transformers, enabling it to achieve performance comparable to transformer-based models such as Swin Transformer while maintaining a simpler, fully convolutional structure. The model demonstrates high accuracy and computational efficiency, making it effective in capturing detailed visual features such as leaf texture and shape. Its fully convolutional design also allows flexible adaptation to different image resolutions and efficient training and inference on limited computational resources.

ConvNeXt Tiny is the smallest variant of the ConvNeXt architecture built entirely from standard convolutional modules without an attention mechanism. This model adopts a four-stage hierarchical design with a channel configuration of (96, 192, 384, 768) and a block distribution of (3, 3, 9, 3). Each residual block consists of a 7x7 depthwise convolutional layer, two 1x1 convolutional layers with an expansion ratio of 4, as well as a GELU activation function and a LayerNorm normalization layer placed minimally. ConvNeXt Tiny uses a 4x4 stride 4 patchify stem and a separate downsampling layer

in the form of a 2x2 stride 2 convolution between stages. With a computational complexity of 4.5 GFLOPs and 28.6 million parameters, this model is designed to balance efficiency and representation capabilities at medium computing scales [20] .

3.5 Experimental Settings

The experiments were conducted on four pretrained models with weights from ImageNet. The backbone of each model was frozen which means that the parameters are not updated based on the gradient. The classifier attached to the backbone was experimented with or without additional hidden layer and updated via loss values.

Each model was evaluated under two different scenarios: one without a hidden layer and one with an additional hidden layer. In this study, the training hyperparameters were determined as shown in Table 1. The training and testing process in this study was conducted on a device running the Windows 11 operating system, using an NVIDIA GTX 1650 Ti graphics card with 4GB of memory and supporting CUDA computational acceleration.

Table 1: Hyperparameter Settings

Hyperparameter	Value
Epoch	50
Optimizer	Adam
Learning Rate	0.001
Batch Size	32
Random Seed	42
Hidden Layer Neurons	512
Dropout	0.5

The process begins with data preprocessing. This stage involves preparing the dataset to meet modeling requirements, including normalization, augmentation, and data division into training and test sets. This study used the BBG52 dataset with a 60:40 data split ratio following the organized split from the original paper. In the experimentation, the data division was also set to a manual division, as it is considered more reliable.

The experiment evaluated the performance of each model on the BBG52 dataset. Each model trained and tested in two scenarios: one without adding a hidden layer and one with adding a hidden layer. The goal of this testing was to determine the performance of each pretrained model on the dataset used, determine the best model, and evaluate the effect of adding a hidden layer.

The next experiment was conducted using a GPU to measure the computation time required for each model to perform inference on several images. The test consisted of 100 images randomly selected from the validation data. The goal of this testing was to evaluate the efficiency of each model in performing classifying, as well as the effect of adding hidden layers.

The final stage is determining the optimal model, which is done by comparing the results of the two previous tests: performance testing and computational time testing. This comparison aims to assess the balance between model performance in producing accurate predictions and the computational efficiency of each pretrained model.

3.6 Evaluation Metrics

In this study, model performance evaluation focused on accuracy and F1-score, while other metrics were not the main focus. Accuracy is the ratio of the number of correct predictions (positive and negative) by the model to the total amount of data. The accuracy formula is shown in Equation 1.

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

Precision is expressed as the ratio between the number of correct positive predictions (True Positive) to the total number of positive predictions made by the model, which includes both correct positive predictions (True Positive) and incorrect positive predictions (False Positive). The precision formula can be seen in Equation 2.

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

Recall or sensitivity is used to measure the ability of a classification model to detect all events that are truly included in the positive class. The recall formula can be seen in Equation 3.

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

F1-score is the harmonic or aligned average of precision and recall, which provides a balanced picture when there is an imbalance between the two. F1-score formula can be seen in Equation 4.

$$F1_{score} = 2x\frac{PrecisionxRecall}{Precision+Recall} \quad (4)$$

4. BBG52 Dataset

The Balikpapan Botanical Gardens is a nature conservation area located in North Balikpapan District, Balikpapan City. Inaugurated in 2014, the area also serves as a research, education, and natural tourism site [21] .

From this area, images of 52 plant species were captured using mobile devices. All images were captured under various lighting conditions, angles, and distances to represent natural conditions in the field This collection of native plant images is used as the BBG52 dataset. This dataset is expected to make a significant contribution to plant species classification research, particularly in natural image processing [5] .

5. Result and Discussions

5.1 Pretrained Model Performance Testing

This testing was conducted on four pretrained models with two different scenarios: one with no hidden layer and one with an additional hidden layer. Performance evaluation was conducted using two main metrics: accuracy and F1-score, which are presented graphically to facilitate comparative analysis between models and scenarios.

Figure 1 shows that the ConvNeXt Tiny model with the addition of a hidden layer achieved the best performance, with a validation accuracy of 94.66%. This superiority is driven by the adoption of modern CNN designs, such as a wider receptive field and efficient parameter allocation, which allows it to consistently learn global features of plant species. In contrast, ResNet50 without a hidden layer had the lowest

accuracy among the other best models, at 89.85%. This is because the direct connection of the backbone to the classification layer limits the capacity for fine-grained feature separation and non-linear representation in the final stage.

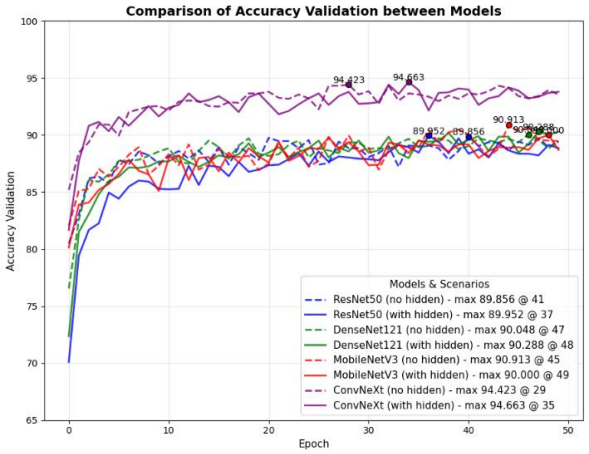

Figure 1: Comparison of Accuracy Validation

Analysis of the impact of adding a single hidden layer showed varying results across models. In ResNet50, DenseNet121 and ConvNeXt Tiny, the additional layer provided a limited but consistent improvement in accuracy by providing a non-linear space to strengthen generic features. However, the addition of the same layer actually decreased accuracy in MobileNetV3 Large. This decrease indicates that the backbones of models already produce compressed and optimal feature representations, so the addition of a hidden layer actually impairs their generalization ability.

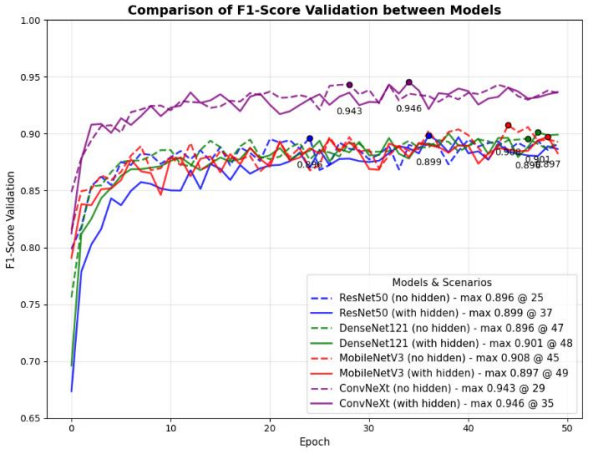

Figure 2: Comparison of F1-Score Validation

The first test also included F1-score validation, shown in Figure 2. Based on the results, the ConvNeXt Tiny model with the addition of a hidden layer performed best, achieving a score of 0.946 thanks to its architecture's ability to produce stable global feature representations, thus improving precision and recall. Conversely, the lowest performance was achieved by ResNet50 and DenseNet121 in the scenario without a hidden layer, with identical scores of 0.896. This indicates that the final feature representations of both models were less effective in separating classes of plant species with high visual similarity, resulting in the balance between precision and recall not being achieved optimally.

Further analysis showed that the addition of a single hidden layer did not provide consistent performance improvements across all architectures. The exception was MobileNetV3 Large, where the F1-score decreased from 0.908 to 0.897 after the addition of the layer. This decrease indicates that the model's feature representation was already compressed and relatively optimal, so the addition of the layer no longer helped class separation but instead disrupted the precision-recall balance compared to other models with a wider representation space.

5.2 Computation Time Testing

This test was conducted using a GPU device to measure the computation time required by each model to perform inference on several images. The evaluation results for the computation time testing of all pretrained models are shown in Figure 3.

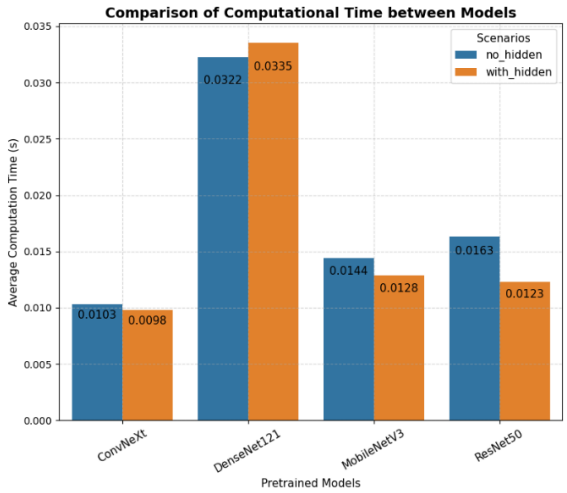

Figure 3: Comparison of Computational Time

Based on the graph in Figure 3, a comparison of computational times indicating that DenseNet121 had the slowest processing time among all tested models, with an average time of approximately 0.0335 seconds in the hidden layer addition scenario. This is related to its architectural characteristics, which utilize dense connectivity, resulting in a relatively large-dimensional final feature vector. Consequently, the classifier stage must handle feature vectors with high complexity, making matrix operations on the fully connected layer more expensive than other backbones that produce more compressed features.

Conversely, ConvNeXt Tiny was the model with the fastest computational time, with an average of 0.0098 seconds in the hidden layer addition scenario. This is due to the ConvNeXt Tiny backbone design, which produces compact and computationally efficient final features. The controlled feature dimensionality means that adding a single hidden layer to the classifier adds very little overhead. Furthermore, its simple and easily parallelizable design allows for efficient GPU execution, so adding layers does not significantly impact inference time.

Meanwhile, MobileNetV3 Large demonstrated stable and relatively efficient computation times in both scenarios. The graph shows its performance is faster than ResNet50 and significantly more efficient than DenseNet121, although still behind ConvNeXt Tiny. Its compact architectural design, through depthwise separable convolution and an inverted

residual bottleneck, produces small-dimensional final features, resulting in a relatively low computational burden in the inference stage.

ResNet50, on the other hand, exhibits intermediate computation times compared to the other models. The graph shows its performance is slower than ConvNeXt Tiny and MobileNetV3 Large, but faster than DenseNet121. This is related to the summation-based residual connection characteristic, which results in moderate feature dimensionality.

Furthermore, the addition of one hidden layer did not significantly improve computational efficiency for any model. Of the four models, only DenseNet121 experienced a slight slowdown after the addition of a hidden layer, from an average of 0.0322 seconds to 0.0335 seconds. This suggests that adding hidden layers to a model with large-dimensional final features tends to increase the computational bottleneck in the classifier, primarily due to the increased processing load of the fully connected layer and the memory access overhead of the feature concatenation mechanism inherent in the DenseNet121 architecture.

5.3 Optimal Model Determination

Before determining the optimal model, a comparison is necessary based on the results of two previous tests: performance testing and computation time testing. This determination is based on the results of tests using a manual dataset. The following comparison of the results can be seen in Figures 4 and Figure 5.

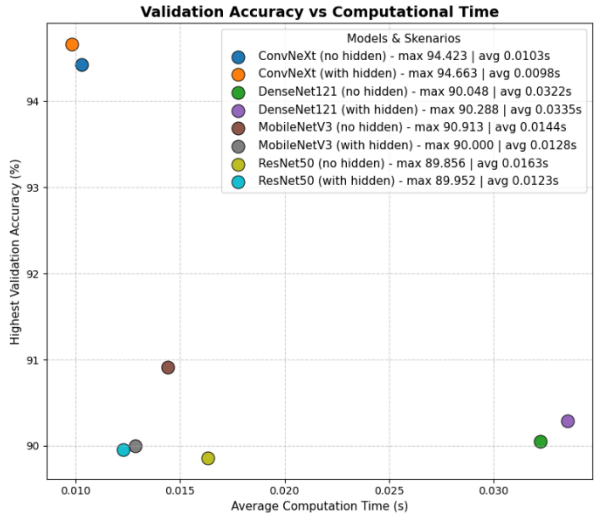

Figure 4: Comparison of Validation Accuracy and Computational Time

Figure 4 shows a graph comparing validation accuracy and computation time. The results show that the superior model is ConvNeXt Tiny with the addition of a hidden layer. It achieved the highest validation accuracy of 94.66% and an average computation time of approximately 0.010 seconds.

Conversely, the weakest model, based on the graph, is the DenseNet121 model without the addition of a hidden layer. It achieved a validation accuracy of 90.04% and a computation time of approximately 0.032 seconds. Although these models appear close to each other, the very small difference in computation time warrants greater focus on the difference in validation accuracy.

Figure 5 shows a graph comparing the results between the validation F1-score and computation time. The results show that the ConvNeXt Tiny model with the addition of a hidden layer holds the highest position, with the highest validation F1-score of 0.946 and an average computation time of only around 0.010 seconds.

Conversely, the weakest model is DenseNet121 without the addition of a hidden layer, with a validation F1-score of 0.896 and a computation time of around 0.032 seconds. Although the models appear close to each other, the very small difference in computation time warrants a closer focus on the difference in the validation F1-score.

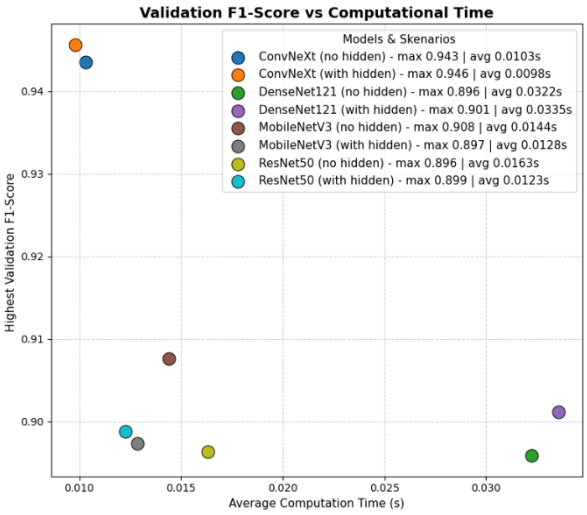

Figure 5: Comparison of Validation F1-Score and Computational Time

Considering the overall results, it can be concluded that the ConvNeXt Tiny model in the added hidden layer scenario is the most optimal among the four pretrained models tested. This model achieved the highest accuracy and F1-score while maintaining the most efficient computation time. This prove empirically that ConvNeXt Tiny not only excels in generalization capabilities on natural plant images but also has excellent computational efficiency for implementation in large-scale classification systems. In contrast, DenseNet121 in the scenario without hidden layers is in the bottom-right side of the graph, meaning this model is the weakest among the four pretrained architectures tested. Meanwhile, the ResNet50 and MobileNetV3 Large models show quite efficient performance despite their relatively low performance.

6. Result and Discussions

Based on the research results, all pretrained models showed quite good performance in plant species classification on the BBG52 dataset, but ConvNeXt Tiny with the addition of one hidden layer proved to be the most superior with the highest validation accuracy of 94.66%, the highest validation F1-score of 0.946, and the fastest computation time of 0.0098 seconds. In contrast, ResNet50 without the addition of a hidden layer showed the lowest performance in terms of accuracy (89.85%) and F1-score (0.896). The test results also showed that the addition of one hidden layer did not always provide a significant increase in performance or computational efficiency, even in some models such as DenseNet121 and MobileNetV3 Large actually caused a decrease in performance or slower computational time, with DenseNet121 being the model with the slowest computational time in that scenario. By considering the balance between classification performance and computational efficiency,

ConvNeXt Tiny with the addition of a hidden layer can be concluded as the best model under the experimental setup used in this study for plant species classification on the BBG52 dataset.

For future studies, it is recommended to test the model on datasets with larger sizes and levels of variation to improve generalization capabilities to natural image diversity, as well as adding a more in-depth analysis of computational efficiency, including testing on GPUs with different specifications to obtain a more accurate picture of the performance ratio and resource requirements. In addition, exploration of more ConvNeXt architecture variants, such as ConvNeXt Small, Base, and Large, as well as comparisons with state-of-the-art pretrained transformer-based models such as Vision Transformer (ViT) and Swin Transformer, is important to understand the effect of complexity and architectural differences on plant species classification performance. Furthermore, the development of the model into a real web-based or mobile application is also recommended so that the research results can be directly utilized by the public, researchers, and conservation institutions.

Authorship Contribution Statement

Dimas Dewanto: Responsible for writing the document, conducting the experiments, and analyzing and interpreting the research results.
Rizky Amelia: Contributed to providing input in the discussion section and document revision.
Rufai Yusuf Zakari: Contributed to the conceptualization of the research and substantial revision of the paper.

Declaration of Conflicting Interests

The authors declare that there are no conflicts of interest that could influence the results of our research.

References

[1] Zhao, X., Li, F., Yan, Y., & Zhang, Q. (2022). Biodiversity in Urban Green Space: A Bibliometric Review on the Current Research Field and Its Prospects. In *International Journal of Environmental Research and Public Health* (Vol. 19, Issue 19). MDPI. https://doi.org/10.3390/ijerph191912544.

[2] Christenhusz, M. J. M., & Byng, J. W. (2016). The number of known plants species in the world and its annual increase. In *Phytotaxa* (Vol. 261, Issue 3, pp. 201–217). Magnolia Press. https://doi.org/10.11646/phytotaxa.261.3.1.

[3] Araújo, V. M., Britto Jr., A. S., Oliveira, L. S., & Koerich, A. L. (2022). Two-view fine-grained classification of plant species. *Neurocomputing*, *467*, 427–441. https://doi.org/10.1016/j.neucom.2021.10.015.

[4] Sterkenburg, T. F., & Grünwald, P. D. (2021). The no-free-lunch theorems of supervised learning. *Synthese*, *199*(3–4), 9979–10015. https://doi.org/10.1007/s11229-021-03233-1.

[5] Ramadhani, R., Alfarisy, G. A. F., & Pratama. B. M. (2025). BBG52: A New Dataset for Plant Species Recognition in the Balikpapan Botanical Gardens, Borneo Island, *Innovative Informatics and Artificial Intelligence Research* (Vol. 1, Issue 1), 19-25. https://doi.org/10.35718/iiair.v1i1.127.

[6] Salsabila, S., & Suharso, A. (2024). Comparison of Deep Learning Architectures in Identifying Types of Medicinal Plant Leaf Images. In Journal of Applied Informatics and Computing (JAIC) (Vol. 8, Issue 1). http://jurnal.polibatam.ac.id/index.php/JAIC.

[7] Bui, H. P., Dinh, A. T., Nguyen, N. M., & Pham, V. K. (2023). A Quantitative Comparison of Classification Methods for Plant Leaf Images. Frontiers in Artificial Intelligence and Applications, 374, 52–59. https://doi.org/10.3233/FAIA230767.

[8] Dey, B., Ferdous, J., Ahmed, R., & Hossain, J. (2024). Assessing deep convolutional neural network models and their comparative performance for automated medicinal plant identification from leaf images. *Heliyon*, *10*(1). https://doi.org/10.1016/j.heliyon.2023.e23655.

[9] Zhou, C. L., Ge, L. M., Guo, Y. B., Zhou, D. M., & Cun, Y. P. (2021). A comprehensive comparison on current deep learning approaches for plant image classification. Journal of Physics: Conference Series, 1873(1). https://doi.org/10.1088/1742-6596/1873/1/012002.

[10] Mehdipour Ghazi, M., Yanikoglu, B., & Aptoula, E. (2017). Plant identification using deep neural networks via optimization of transfer learning parameters. *Neurocomputing*, *235*, 228–235. https://doi.org/10.1016/j.neucom.2017.01.018.

[11] Hama, H. M., Abdulsamad, T. Sh., & Omer, S. M. (2024). Houseplant leaf classification system based on deep learning algorithms. *Journal of Electrical Systems and Information Technology*, *11*(1). https://doi.org/10.1186/s43067-024-00141-5.

[12] Khalid, F., & Romle, A. A. (2024). Herbal Plant Image Classification using Transfer Learning and Fine-Tuning Deep Learning Model. *Journal of Advanced Research in Applied Sciences and Engineering Technology Journal Homepage: Journal of Advanced Research in Applied Sciences and Engineering Technology*, *35*(1), 16–25. https://doi.org/10.37934/araset.35.1.1625.

[13] Gustineli, M., Miyaguchi, A., & Stalter, I. (2024). *Multi-Label Plant Species Classification with Self-Supervised Vision Transformers*. http://arxiv.org/abs/2407.06298.

[14] Lapkovskis, A., Nefedova, N., & Beikmohammadi, A. (2025). *Automatic Fused Multimodal Deep Learning for Plant Identification*. https://doi.org/10.3389/fpls.2025.1616020.

[15] Mareta Tama, A., & Candra Noor Santi, R. (2023). Klasifikasi Jenis Tanaman Hias Menggunakan Metode Convolutional Neural Network (CNN) Ornamental Plant Classification Using The Convolutional Neural Network (CNN) Method. *Journal of Information Technology and Computer Science (INTECOMS)*, *6*(2).

[16] Duta, I. C., Liu, L., Zhu, F., & Shao, L. (2020). *Improved Residual Networks for Image and Video Recognition*. http://arxiv.org/abs/2004.04989.

[17] Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. Q. (2017). *Densely Connected Convolutional Networks*. https://github.com/liuzhuang13/DenseNet.

[18] Khasoggi, B., Ermatita, & Samsuryadi. (2019). Efficient mobilenet architecture as image recognition on mobile and embedded devices. *Indonesian Journal of Electrical Engineering and Computer Science, 16*(1), 389–394. https://doi.org/10.11591/ijeecs.v16.i1.pp389-394.

[19] Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. v, Adam, H., Ai, G., & Brain, G. (n.d.). *Searching for MobileNetV3*.

[20] Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., Xie, S., & Research, F. A. (2022). *A ConvNet for the 2020s*. https://github.com/facebookresearch/ConvNeXt.

[21] Nugroho, R. A., Syafitri, E. D., & Yorika, R. (2021). Konsep Pengembangan Kebun Raya Balikpapan Sebagai Destinasi Wisata Alam Unggulan. *Jurnal Pembangunan Wilayah Dan Kota*, *17*(3), 307–315. https://doi.org/10.14710/pwk.v17i3.33569.

