ECTI Transactions on Computer and Information Technology
Journal homepage: https://ph01.tci-thaijo.org/index.php/ecticit/
Published by the ECTI Association, Thailand, ISSN: 2286-9131

Comparison of CNN Architectures for Thai Medicinal Plant Classification

Sompong Valuvanathorn[1] and Chanchai Supaartagorn[2]

ABSTRACT

Thai medicinal plants are essential to traditional healthcare and local livelihoods. However, many Thai medicinal plants have similar morphological characteristics such as shape, colour, and texture. This problem leads to misidentification and misclassification. Image classifiers utilizing convolutional neural networks (CNNs), which are a class of deep learning models, provide a scalable substitute for manual classification. This study aims to evaluate and compare the performance of three CNN architectures (DenseNet-121, EfficientNet-B3, and MobileNetV2) for classifying 10 species of Thai medicinal plants. The dataset comprises 5,000 leaf images representing 10 species (500 images per species). This study partitioned the dataset into 80% training set and a 20% test set. To enhance model generalization, we applied data augmentation techniques—specifically rotation, flipping, and colour manipulation. Furthermore, we utilized TensorFlow and Keras on Google Colab with GPU acceleration to train the models. Evaluation metrics include accuracy, precision, recall, F1 score, model size, inference time, and CPU utilization. The results highlight a trade-off between accuracy and efficiency: DenseNet-121 achieved the highest accuracy at 96.0% and a Matthews Correlation Coefficient (MCC) of 0.9558. Statistical analysis confirmed that DenseNet-121 significantly outperformed the other architectures ($p < 0.05$), albeit with a higher inference time (579.22 s). Notably, EfficientNet-B3 and MobileNetV2 both achieved an accuracy of 93.4%, with MobileNetV2 performing the best in terms of model size (11.07 MB) and inference time (3.86 s). In conclusion, DenseNet-121 is the most accurate model, while MobileNetV2 is best suited for real-time applications due to its lightweight and rapid inference time. EfficientNet-B3 offers an optimal balance between accuracy and computational efficiency.

Article information:

Keywords: Thai Medicinal Plants, CNN Architectures, DenseNet-121, EfficientNet-B3, MobileNetV2

Article history:
Received: October 17, 2025
Revised: January 8, 2026
Accepted: March 12, 2026
Published: March 28, 2026 (Online)

DOI: 10.37936/ecti-cit.2026202.264238

1. INTRODUCTION

Thai medicinal plants are essential to both traditional healthcare and local livelihoods. Due to their affordability and accessibility, people in Thailand widely utilize herbal medicines to cure illnesses and preserve health [1]. Furthermore, botanists have long acknowledged the relevance of herbal plants in traditional medicine, as seen in the large number of contemporary medications inspired by or derived from medicinal plant substances [2]. Consequently, accurate identification and classification of these medicinal species is important, yet it remains challenging for the general public [3]. Due to their highly similar morphological characteristics, many Thai medicinal plants are difficult to distinguish visually without the assistance of an expert. For instance, many basil or leafy medicinal plants exhibit striking similarities [4]. Non-expert users often struggle to distinguish one species from another, which can lead to misidentification. Using the incorrect plant might make a cure ineffective or even harmful; therefore, these mistakes are highly significant. Indeed, misidentifying plants as toxic or medicinal can have detrimental effects on both human and animal health [5]. This highlights the need for better, more approachable ways to recognize Thai medicinal plants based on appearance.

[1,2]The authors are with the Department of Mathematics and Digital Technology, Faculty of Science Ubon Ratchathani University, Ubon Ratchathani, Thailand, 34190, Email: sompong.v@ubu.ac.th and chanchai.s@ubu.ac.th
[2]Corresponding author: chanchai.s@ubu.ac.th

Manual observation and specialized knowledge are key components of traditional plant classification techniques. To classify species based on morphology, botanists and herbalists have long relied on field guides, taxonomic keys, and their expertise [4]. However, these conventional approaches have notable limitations. They are time-consuming and often impractical for everyday users or farmers, as accurate classification may require examining fine details or the plant's full life cycle. Even commonly used techniques like macroscopic inspection demand a complete specimen and do not work well when only fragments are available [6]. Novices seeking to classify a medicinal plant based on leaf morphology or pigmentation can easily make mistakes, as numerous species exhibit overlapping visual traits [4]. Furthermore, expert-dependent approaches are not scalable since few people possess the specific skills necessary to classify the hundreds of Thai medicinal plants. This gap between the widespread use of medicinal plants and the difficulty of traditional classification highlights a pressing problem in herbal medicine practice and biodiversity management.

Deep learning methodologies have revolutionized plant disease identification and diagnosis, demonstrating remarkable progress in recent years [7]. These advancements make deep learning a promising approach to tackle the classification of Thai medicinal plants, which involve subtle visual distinctions that CNNs are adept at learning. In recent years, deep learning—particularly Convolutional Neural Networks (CNNs)—has emerged as a highly effective approach for image-based plant classification. CNNs are especially well suited to this task because they automatically learn discriminative visual features (such as shape, texture, and colour patterns) directly from training images, thereby eliminating the need for manual feature engineering [4]. Modern CNN architectures are more accurate at identifying plant species than previous computer vision techniques [8]. Several studies show that CNN models can accurately classify medicinal substances or plant leaves. For example, a previous study classified the leaves of five medicinal plants (Lemon Balm, Stevia, Peppermint, Bael and Tulsi) collected in the northern area of Iran, achieving a classification accuracy of 99.3% [4]. CNNs typically employ a supervised learning approach, where the model receives a set of labeled training images and learns to map these inputs to their correct labels. Therefore, the development of robust and accurate plant identification systems can significantly enhance the accessibility and utilization of medicinal plants. However, selecting the most suitable architecture for specific applications remains a challenge, particularly when considering the trade-off between model performance and resource usage. This study aims to address this gap by investigating the following research questions: 1) How do the prominent CNN architectures (DenseNet-121, EfficientNet-B3, and MobileNetV2) compare in terms of classification accuracy for Thai medicinal plant species? 2) Which CNN architecture demonstrates the optimal balance between high classification performance and computational efficiency (inference time and model size)? To answer these questions, we examine three prominent CNN architectures: DenseNet-121, EfficientNet-B3, and MobileNetV2. We compare their characteristics and performance for plant classification, with an emphasis on Thai medicinal plants. The results from this research will help in selecting the appropriate architecture for specific deployment needs, ranging from accuracy-prioritized applications to resource-constrained real-time environments.

2. RELATED WORK

Deep learning is a subset of machine learning that uses multilayered neural networks to learn data representations at multiple levels of abstraction. Convolutional Neural Networks (CNNs), a prominent deep learning technique, have made significant progress in image identification applications and have been widely adopted for image recognition [9]. Various applications, including image and video processing, natural language processing (NLP), and recommendation systems, utilize CNN architectures. Structurally, CNNs use nonlinear activations, pooling, and stacked convolutional layers to automatically learn visual properties. During processing, convolutional layers receive the image and apply kernels (filters) to extract meaningful features, such as edges and textures, rather than merely modifying pixel shades. This convolution process generates feature maps, modifying the spatial dimensions of the input based on hyperparameters such as kernel size, stride, and padding. Subsequently, pooling layers reduce feature map dimensions by summarizing features present in specific regions of the maps. The flatten layer converts the feature map received from the pooling layer into a format that the dense layers can understand. Since a feature map is essentially a multi-dimensional array containing pixel values, the dense layers require a one-dimensional array as input for processing. CNNs end with fully connected layers, whose inputs match the one-dimensional matrix flattened by the final pooling layer. Finally, an output prediction layer produces probability values for every potential label. The model ultimately predicts the label with the highest probability score. Fig. 1 illustrates the standard CNN architecture. CNNs are very effective for classifying plant images because they can extract discriminative features (leaf form, texture, venation patterns, etc.) without the need for manual feature engineering [10]. In plant identification tasks, CNN-based approaches significantly outperform traditional plant identification methods by reducing the complexity of expert-driven processes. A recent review found that CNNs represent the predominant technique for herbal plant identification, appearing in 64.5% of research papers published between 2018 and 2022 [10].

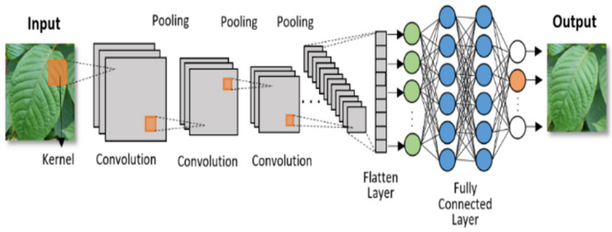


Fig.1: *Convolutional Neural Network (CNN).*

Several CNN architectures exist, each designed with distinct operational mechanisms. For example, Residual Networks (ResNet) [11] represent a significant milestone in computer vision, utilizing shortcut connections to enable deeper architectures and improve convergence. However, recent advancements have shifted towards maximizing parameter efficiency. Furthermore, Vision Transformers (ViTs) [12] have emerged as powerful alternatives, leveraging self-attention mechanisms to capture global context. Yet, ViTs typically require massive datasets to generalize effectively and demand substantial computational resources, posing challenges for deployment in resource-constrained environments or with limited regional datasets.

Consequently, this study selects DenseNet-121, EfficientNet-B3, and MobileNetV2 to represent distinct design philosophies suitable for Thai medicinal plant classification. We chose DenseNet-121 for its feature reuse capability through dense connectivity, which is advantageous for capturing fine-grained morphological details of leaves with fewer parameters than traditional ResNets. We included EfficientNet-B3 to represent the state-of-the-art in compound scaling, offering an optimal balance between accuracy and model complexity. Finally, MobileNetV2 was selected to address the practical necessity for lightweight models deployable on mobile devices, ensuring accessibility for local farmers and practitioners. This selection allows for a comprehensive evaluation covering high-precision, balanced, and efficiency-oriented scenarios.

2.1 DenseNet-121

Proposed by Huang *et al.* [13] in 2017, DenseNet-121 is a convolutional neural network that introduces dense connectivity, where each layer receives feature maps from all preceding layers via concatenation rather than summation. This design encourages feature reuse, improves gradient flow, and minimizes redundant feature learning. Dense blocks in DenseNet consist of multiple convolution layers that are densely connected, which leads to improved parameter efficiency and model compactness. The architecture includes transition layers between dense blocks that apply 1×1 convolution and average pooling to reduce spatial dimensions and the number of feature maps. DenseNet-121, with its 121 layers arranged in four dense blocks, achieves high accuracy with fewer parameters and is particularly suitable for classification tasks involving limited image data [13].

2.2 EfficientNet

Introduced by Tan and Le [14], EfficientNet aims to improve CNN performance by optimizing network scaling. EfficientNet-B3 is an upscaled version of EfficientNet-B0 and leverages the principle of compound scaling, which uniformly scales depth, width, and input resolution using a fixed ratio. This balanced scaling strategy allows EfficientNet to achieve better accuracy with significantly fewer parameters and FLOPs compared to conventional models. EfficientNet-B3 employs MBConv blocks (Mobile Inverted Bottleneck Convolution) and Squeeze-and-Excitation (SE) modules to recalibrate channel-wise feature responses. It also adopts the Swish activation function to enhance learning dynamics.

2.3 MobileNetV2

Developed by Google Research in 2018, MobileNetV2 targets mobile and embedded vision applications with constrained resources. It introduces two key architectural innovations: Inverted Residuals and Linear Bottlenecks. Unlike traditional residuals, inverted residuals expand features first and then compress them, while linear bottlenecks avoid nonlinearities at the output to preserve essential information. MobileNetV2 consists of 17 bottleneck blocks using 1×1 pointwise and 3×3 depthwise separable convolutions, significantly reducing computational overhead while maintaining accuracy. Its lightweight and efficient design makes it a strong candidate for deploying real-time classification models on edge devices such as smartphones [15].

3. METHODOLOGY

This section outlines the research methodology employed for classifying Thai medicinal plants. We divide the experimental framework is divided into five main parts: (1) Datasets, detailing the collection and preparation of 5,000 leaf images; (2) Image Augmentation, describing techniques to artificially expand the dataset; (3) Training, covering the model construction and learning process; (4) Classification, which explains the model deployment and saving procedures; and (5) Experimental Environment, specifying the hardware and software configurations utilized.

3.1 Datasets

Data collection was the foundational stage of this research. Given Thailand's status as a tropical country rich in biodiversity, with over 20,000 plant species and 1,800 medicinal herbs recorded by the Ministry of Agriculture and Cooperatives, reliable references were essential. We based the selection of Thai medicinal plants for this study was based on the Thai Herbal Pharmacopoeia (Bureau of Drug and Narcotic, Department of Medical Sciences, Ministry of Public Health) and the Herbal Medicine Database of the Faculty of Pharmacy, Ubon Ratchathani University.

The study focused on 10 specific species: Tiliacora triandra (T. triandra), Ocimum africanum (O. africanum), Ocimum tenuiflorum (O. tenuiflorum), Citrus hystrix (C. hystrix), Mentha cordifolia (M. cordifolia), Thunbergia laurifolia (T. laurifolia), Piper retrofractum (P. retrofractum), Chromolaena odorata (C. odorata), Centella asiatica (C. asiatica), and Piper sarmentosum (P. sarmentosum).

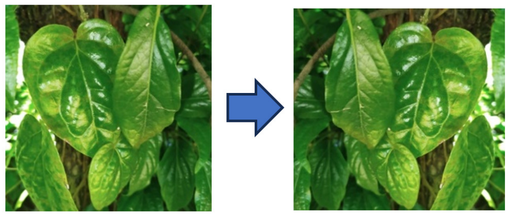
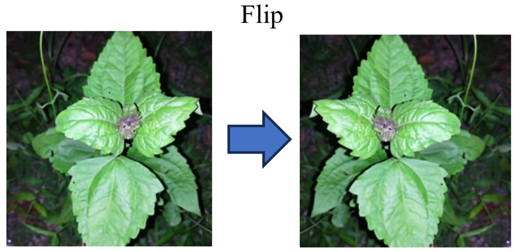
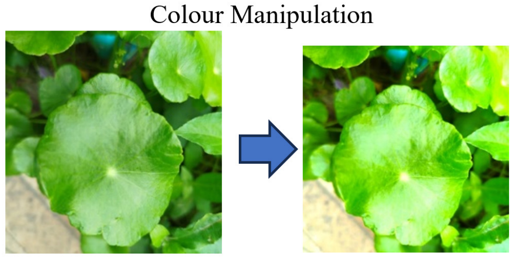
The final dataset employed in this study comprises 5,000 images representing these 10 distinct species. To ensure a balanced distribution and mitigate class imbalance bias, the dataset contains exactly 500 images for each species. We collected the images from medicinal plant gardens and natural habitats, primarily using smartphone cameras. To address potential sampling biases and ensure the model's robustness to real-world variability, we conducted data acquisition under uncontrolled conditions. This included capturing images at various times of the day (morning and afternoon) to cover diverse lighting environments—ranging from bright sunlight to low-light conditions—as well as varying angles and complex natural backgrounds. This diversity is crucial to prevent the model from learning spurious correlations with specific background features.

For model training and evaluation, we partitioned the dataset into training (80%) and validation (20%) sets using stratified random sampling. This method ensures that the class distribution remains consistent across both subsets, preserving the balanced nature of the data. This balance allows for a fair evaluation of the model's performance on unseen data without the immediate necessity for k-fold cross-validation given the dataset size.

3.2 Image Augmentation

Image augmentation approaches are employed to artificially increase the size of the dataset. Image augmentation can improve the performance of deep learning models and expand limited datasets to take advantage of the capabilities of big data [14]. Previous research demonstrates that this technique helps improve the model's average accuracy. For instance, Roslan NAM *et al.* [16] developed a CNN model for Malaysian medicinal herbs. Models trained on real herb data obtained an average accuracy of 75%, while those trained on augmented data achieved an average accuracy of 88%. Table 1 presents examples of Thai medicinal plants before and after applying augmentations such as rotation, flipping, and colour manipulation.

Table 1: *Example of Image augmentation.*

No	Augmentation Techniques
I	Rotate 
II	Flip 
III	Colour Manipulation 

3.3 Training

In this stage, we implemented and trained the CNN architectures using the prepared dataset. We evaluated three specific models—DenseNet-121, EfficientNet-B3, and MobileNetV2—and monitored their classification accuracy on both training and validation sets. Given the multi-class nature of the task, we employed categorical cross-entropy as the loss function. For the intermediate dense layers, we selected the Rectified Linear Unit (ReLU) activation function due to its ability to mitigate the vanishing gradient problem, enabling faster and more stable convergence compared to saturating activation functions. Furthermore, ReLU induces sparsity in the hidden units, allowing the model to efficiently learn representations of the discriminative morphological features extracted by the convolutional base. The final output layer utilized the softmax function to generate class probabilities.

The training configuration included resizing input images to 224×224 pixels with pixel values rescaled by 1/255. We adopted an 80:20 data split to ensure sufficient training data while maintaining a statistically significant validation set for monitoring performance. We utilized the Adam optimizer for weight updates with a learning rate of 1×10^{-4}, which proved optimal for fine-tuning; higher rates tended to destabilize the pre-trained weights, while lower rates resulted in slow convergence. GPU memory limitations determined the batch sizes: EfficientNet-B3 necessitated a size of 16 due to its higher computational overhead, whereas a size of 32 was feasible for the lighter DenseNet-121 and MobileNetV2. Regarding training duration, preliminary runs indicated that the models converged consistently within 30 epochs, with validation loss stabilizing; therefore, we used a fixed epoch count. To prevent overfitting, we primarily enforced regularization through the data augmentation techniques detailed in Section 3.2, rather than additional weight decay or dropout layers, as the pre-trained architectures already include inherent regularization mechanisms (e.g., batch normalization). The pseudocodes in Fig. 2 - Fig. 4 illustrate the model construction and training loop for each architecture.

```
Algorithm 1: DenseNet-121 Architecture Setup and Training
------------------------------------------------------------------------------
Input: Image dataset D (Thai medicinal plants, 224 x 224 x 3)
Output: Best Trained DenseNet-121 Classification Model

// 1. Data Preparation
Apply rescale factor 1./255 to D
Split D into Training_Set (80%) and Validation_Set (20%)

// 2. Initialize Base Model
Base_Model ← Load DenseNet121(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
Set all layers in Base_Model to non-trainable (Freeze weights)

// 3. Construct Classification Head
x ← Base_Model.output
x ← GlobalAveragePooling2D()(x)
x ← Dense(units=512, activation='relu')(x)
Predictions ← Dense(units=10, activation='softmax')(x)

// 4. Build and Compile Final Model
Model ← Build_Model(inputs=Base_Model.input, outputs=Predictions)
Compile Model USING:
    Optimizer = Adam(learning_rate=0.0001)
    Loss = Categorical Cross-entropy
    Metrics = ['accuracy']

// 5. Model Training
Checkpoint ← Save model when 'val_accuracy' is maximum
Train Model on Training_Set for 30 epochs
Evaluate Model simultaneously on Validation_Set

Return Best Model from Checkpoint
------------------------------------------------------------------------------
```

Fig.2: *Pseudocode of the DenseNet-121 architecture and training setup.*

3.4 Classification

In this phase, we utilized the trained neural networks (DenseNet-121, EfficientNet-B3, and MobileNetV2) for the automatic classification of Thai medicinal plants. We selected neural networks as the primary classification tool due to their proven success in various real-world applications. Upon completing the training process, we saved the models in a .keras format. This binary format stores the complete Keras model, including its architecture, trained

```
Algorithm 2: EfficientNet-B3 Architecture Setup and Training
------------------------------------------------------------------------------
Input: Image dataset D (Thai medicinal plants, 224 x 224 x 3)
Output: Trained EfficientNet-B3 Classification Model

// 1. Data Preparation
Apply rescale factor 1./255 to D
Split D into Training_Set (80%) and Validation_Set (20%)
Set batch_size = 16

// 2. Initialize Base Model
Base_Model ← Load EfficientNetB3(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

// 3. Construct Classification Head
x ← Base_Model.output
x ← GlobalAveragePooling2D()(x)
x ← Dense(units=256, activation='relu')(x)
Predictions ← Dense(units=10, activation='softmax')(x)

// 4. Build and Compile Final Model
Model ← Build_Model(inputs=Base_Model.input, outputs=Predictions)
Compile Model USING:
    Optimizer = Adam(learning_rate=0.0001)
    Loss = Categorical Cross-entropy
    Metrics = ['accuracy']

// 5. Model Training (Epoch-by-epoch Execution)
For epoch = 1 to 30 do:
    Train Model on Training_Set for 1 epoch
    Evaluate Model simultaneously on Validation_Set
    Save Model state to Google Drive
    Record 'accuracy' and 'loss' to training log
End For

Return Trained Model
------------------------------------------------------------------------------
```

Fig.3: *Pseudocode of the EfficientNet-B3 architecture and training setup.*

```
Algorithm 3: MobileNetV2 Architecture Setup and Segmented Training
------------------------------------------------------------------------------
Input: Image dataset D (Thai medicinal plants, 224 x 224 x 3)
Output: Trained MobileNetV2 Classification Model

// 1. Data Preparation
Apply rescale factor 1./255 to D
Split D into Training_Set (80%) and Validation_Set (20%)
Set batch_size = 32

// 2. Initialize Base Model
Base_Model ← Load MobileNetV2(weights='imagenet', include_top=False, input_shape=(224, 224, 3))
Set all layers in Base_Model to non-trainable (Freeze weights)

// 3. Construct Classification Head
x ← Base_Model.output
x ← GlobalAveragePooling2D()(x)
x ← Dense(units=128, activation='relu')(x)
x ← Dropout(rate=0.5)(x)
Predictions ← Dense(units=num_classes, activation='softmax')(x)

// 4. Build and Compile Final Model
Model ← Build_Model(inputs=Base_Model.input, outputs=Predictions)
Compile Model USING:
    Optimizer = Adam(learning_rate=0.0001)
    Loss = Categorical Cross-entropy
    Metrics = ['accuracy']

// 5. Training State Management & Segmented Training
Load Model_Weights and Training_History if previous checkpoint exists
Set Total_Epochs = 30
Set Step_Size = 5

If Last_Epoch < Total_Epochs Then
    Train Model on Training_Set for Step_Size epochs
    Evaluate Model simultaneously on Validation_Set
    Save current Model weights to Checkpoint
    Update and save Training_History state (JSON)
End If

Return Trained Model
------------------------------------------------------------------------------
```

Fig.4: *Pseudocode of the MobileNetV2 architecture and training setup.*

Table 2: *Model configuration and hyperparameters.*

Parameter	Specification
Input Image Size	224 × 224 (pixels)
Base Architectures	DenseNet-121, EfficientNet-B3, MobileNetV2
Feature Extraction	Global Average Pooling 2D
Intermediate Layers	ReLU
Output Layer	Dense (10 nodes)
Output Activation	Softmax
Loss Function	Categorical Cross-entropy
Optimizer	Adam (Learning rate = 1×10^{4})
Batch Size	16 (EfficientNet-B3), 32 (DenseNet-121, MobileNetV2)
Epochs	30

weights, and training configuration (optimizer, loss, and metrics), allowing for efficient loading and reuse of the Convolutional Neural Network (CNN) models for deployment. Fig. 5 illustrates the overall experimental process.

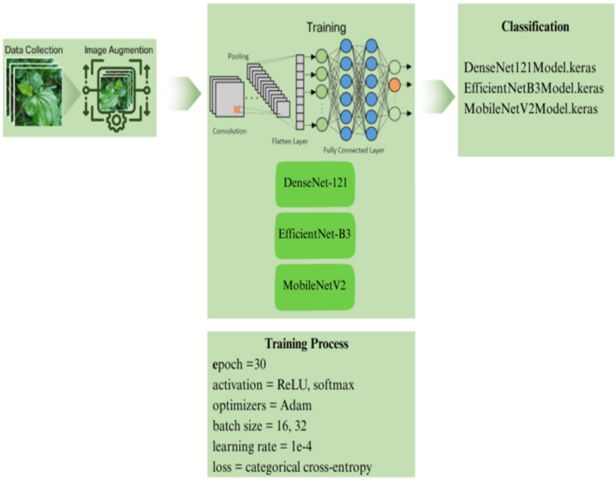


Fig.5: *Process of experiments.*

3.5 Experimental Environment

We conducted the experiments in this study on Google Colab (Python 3.8) using the Keras 2.10.0 library. Additionally, we utilized TensorFlow 2.10.0, which is one of the best Python deep learning libraries available for working with machine learning methods. In this study, we trained the original, transfer learning, and ensemble models using a Google Colab Tesla T4 Graphics Processing Unit (GPU).

4. MODEL EVALUATION

Our evaluation methodology relies on the confusion matrix, which assesses classification quality by comparing predicted classes with actual classes. We utilized four key parameters defined for each specific medicinal plant class: True Positives (TP) represent instances where the model correctly identifies a specific plant species; True Negatives (TN) represent instances where the model correctly identifies that an image does not belong to that specific species; False Positives (FP) occur when the model incorrectly classifies an image of another species as the specific target species; and False Negatives (FN) occur when the model fails to identify the specific target species, misclassifying it as another. The selection of evaluation metrics connects directly to the classification challenges and application scenarios highlighted in the introduction. Accuracy and F1-score serve as primary indicators of the model's reliability in distinguishing between species with high morphological similarity. Precision is particularly critical in the context of medicinal plants to minimize false positives, ensuring safety by preventing the misclassification of non-medicinal or toxic plants as target herbs. Consequently, we calculated and compared the accuracy, precision, recall, and F1-score. Equations (1) to (4) provide the mathematical definitions for each metric.

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

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

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

$$F1 - score = \frac{2 \times (Precision \times Recall)}{(Precision + Recall)} \quad (4)$$

In addition to these metrics, In addition to these metrics, we also evaluated model size, inference time, and CPU utilization as key factors in selecting a model for practical deployment:

Model Size refers to the total storage space required by a trained convolutional neural network (CNN) architecture, typically measured in megabytes (MB). It includes all learned parameters and any additional overhead needed for model execution. In practical deployment scenarios, model size directly influences memory usage, loading time, and suitability for resource-constrained environments such as mobile or embedded systems.

Inference Time denotes the average amount of time, measured in milliseconds (ms), that a trained model takes to process a single input image and generate a prediction. This metric reflects the model's computational efficiency during real-time classification and is critical for applications requiring fast or real-time responses.

CPU Utilization represents the percentage of total processing capacity used by the central processing unit (CPU) during the inference phase. It is a key indicator of how computationally demanding a model is, especially in non-GPU environments. Lower CPU utilization suggests better efficiency and a lighter computational load, which is advantageous for deployment on low-power or multi-tasking systems.

5. RESULTS AND DISCUSSION

This section presents the outcomes of training and comparing three convolutional neural network (CNN) architectures—DenseNet-121, EfficientNet-B3, and MobileNetV2—on a dataset of Thai medicinal plant images. The results are divided into two parts: training performance and testing performance. We evaluated the performance of the developed CNN models using a comprehensive dataset consisting of images from various plant species. Specifically, we sourced Thai medicinal leaves from 10 distinct species. Because the initially gathered data was inadequate for robust model training, we applied image augmentation techniques to the dataset. We then fed this amplified data into the models. We divided the total image dataset into an 80:20 ratio for training and testing, respectively. The final Thai medicinal plant dataset consists of 5,000 images; we utilized 4,000 images for training and 1,000 images for testing.

5.1 Model Training Results

In every experiment, we utilized classification accuracy and categorical cross-entropy loss to evaluate model performance. Each experimental run consisted of 30 epochs. We set the number of training epochs to 30 based on the convergence behavior observed during preliminary experiments. Since the proposed architectures utilized transfer learning with pre-trained ImageNet weights, the models were able to adapt to the Thai medicinal plant dataset rapidly without requiring extensive training durations. Our analysis of the training and validation curves (Fig. 6) indicated that the models reached stability and optimal performance metrics within this period. Extending the training beyond 30 epochs provided negligible gains in validation accuracy and increased the risk of overfitting, where the model begins to memorize the training data rather than generalizing to unseen samples. Fig. 6 depicts the training and validation trends, and Table 3 summarizes the final performance results on the test dataset.

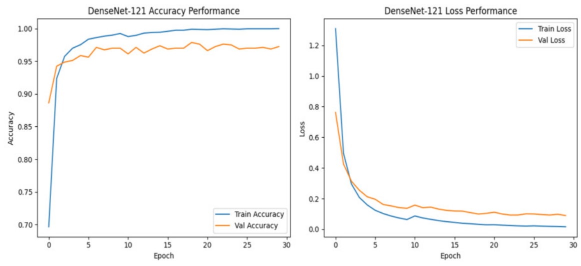
Table 3: *Training and validation performance per model.*

Model	Train Accuracy	Train Loss	Validation Accuracy	Validation Loss
DenseNet-121	100	0.01598	97.88	0.0894
Efficient Net-B3	99.91	0.0042	98.63	0.0622
MobileNet V2	99.31	0.0471	96.13	0.1114

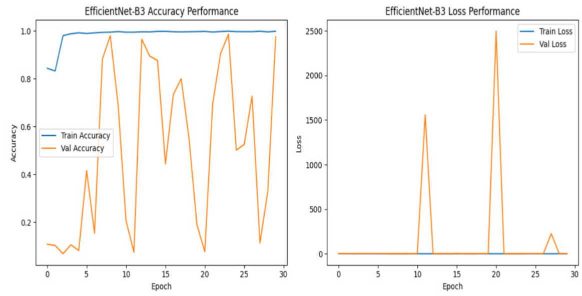
All three models achieved high training and validation accuracy, indicating effective learning. DenseNet-121 and EfficientNet-B3 showed particularly low training loss. However, the gap between training and validation loss in MobileNetV2 may indicate slight overfitting. Fig. 6 displays the comprehensive plots illustrating the accuracy and losses in relation to epochs for Thai medicinal plants. The x-axis represents the epochs, while the y-axis represents the metrics across all graphs.

5.2 Model Testing Results

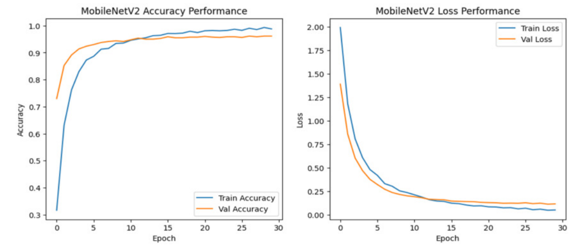
We categorized the quantitative evaluation of the proposed CNN architectures into classification performance and computational efficiency.



(a)



(b)



(c)

Fig.6: *Accuracy performance, Loss performance (a) DenseNet-121 (b) Efficient**Net**-B3 (c) MobileNetV2.*

5.2.1 Classification Performance

Table 4 summarizes the performance metrics on the test dataset. DenseNet-121 achieved the highest overall accuracy of 96.00%, outperforming other architectures. It also demonstrated superior consistency in macro-averaged precision (0.96), recall (0.96), and F1-score (0.96). EfficientNet-B3 and MobileNetV2 both achieved a comparable accuracy of 93.40%, indicating that they remain effective classifiers despite being lightweight. Beyond standard metrics, we calculated the Matthews Correlation Coefficient (MCC) to evaluate the quality of binary and multiclass classifications, providing a more reliable measure than accuracy for balanced and unbalanced

Table 4: *Classification performance metrics of the proposed CNN architectures.*

Model	Accuracy	Precision	Recall	F1-Score	MCC
Dense Net121	96.0	0.96	0.96	0.96	0.9558
Efficient Net-B3	93.4	0.94	0.93	0.93	0.9274
Mobile NetV2	93.4	0.94	0.93	0.93	0.9204

datasets. DenseNet-121 achieved the highest MCC of 0.9558, indicating a very strong correlation between the observed and predicted classifications, surpassing EfficientNet-B3 (0.9274) and MobileNetV2 (0.9204).

To determine whether the performance differences between the best-performing model (DenseNet-121) and the other architectures were statistically significant, we conducted McNemar's test. The results, summarized in Table 5, reveal that DenseNet-121 significantly outperformed both EfficientNet-B3 ($p = 0.00255$) and MobileNetV2 ($p = 0.00031$). Since the obtained p-values are less than the significance level of 0.05, we rejected the null hypothesis, confirming that the superior accuracy of DenseNet-121 is statistically significant.

Table 5: *Statistical comparison of model performance using McNemar's Test.*

Pairwise Comparison	p-value	Significance ($\alpha = 0.05$)	Result
Dense Net121 vs. EfficientNet-B3	0.00255	$P < 0.05$	Significant Difference
Dense Net121 vs. MobileNetV2	0.00031	$P < 0.05$	Significant Difference

To further analyze the misclassification patterns, Fig. 7 presents the confusion matrices for the three models.

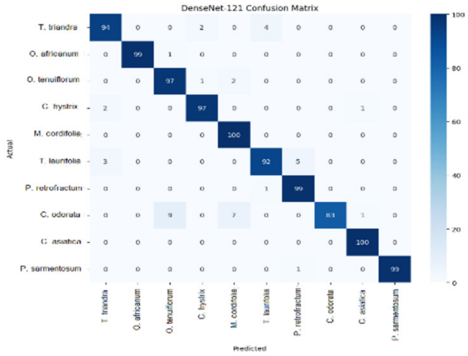
Fig. 7(a) shows that DenseNet-121 exhibits strong diagonal dominance, confirming its ability to correctly classify most species with minimal errors. In contrast, Fig. 7(b) and Fig. 7(c) (EfficientNet-B3 and MobileNetV2) reveal slightly more off-diagonal misclassifications. These errors primarily occurred between species with highly similar leaf morphologies, highlighting the challenge of visual ambiguity in smaller or more efficient models.

5.2.2 Computational Efficiency

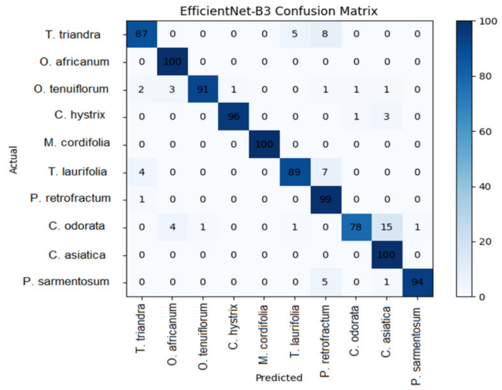
Table 6 highlights the trade-off between accuracy and resource utilization. While DenseNet-121 provided the highest accuracy, it was the most computationally intensive, requiring the longest inference time (579.22 s) and the largest model size (34.34 MB). Conversely, MobileNetV2 proved to be the most efficient, achieving an inference time of only 3.86 s (approximately 150 times faster than DenseNet-121) with a compact model size of 11.07 MB, making it highly suitable for real-time edge deployment.

Table 6: *Comparison of computational efficiency and resource utilization.*

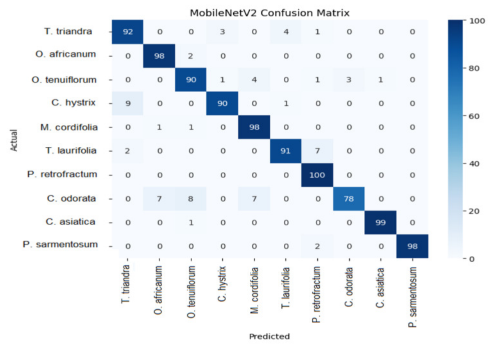
Model	Size (MB)	CPU (%)	Inference Time (Sec)
DensenNet121	34.34	44.4	579.22
EfficientNet-B3	128.78	57.1	150.52
MobileNetV2	11.07	2.0	3.86



(a)



(b)



(c)

Fig. 7: *Confusion Matrix (a) DenseNet-121 (b) EfficientNet-B3 (c) MobileNetV2.*

6. DISCUSSION

This study demonstrates the effectiveness of CNNs in classifying Thai medicinal plants by addressing the limitations of traditional classification methods. DenseNet-121, with its dense connectivity and efficient feature reuse, achieved the highest classification accuracy (96.00%). This finding aligns with Huang *et al.* [13] and Ahad *et al.* [17], who highlighted the architecture's superior capacity for feature extraction. Furthermore, our results are consistent with [18], which focused on evaluating state-of-the-art deep neural networks for plant disease classification; their study notably reported that DenseNet architectures achieve exceptional accuracy (up to 99.75% in their specific experiments) while requiring fewer parameters and reasonable computing time compared to other deep models. However, in our study, despite its high accuracy, DenseNet-121's inference time (579.22 s) and model size (34.34 MB) make it less suitable for real-time applications. EfficientNet-B3, based on compound scaling and MBConv blocks, achieved slightly lower accuracy (93.4%) but offered a better trade-off between performance and computational cost, consistent with Tan and Le's [14] findings. Meanwhile, MobileNetV2, designed for mobile and embedded systems, demonstrated the fastest inference (3.86 s) and smallest model size (11.07 MB), confirming its suitability for edge deployment as supported by [15].

The observed performance differences stem from the fundamental architectural designs of each model. DenseNet-121's achieves superior accuracy because its dense connectivity pattern facilitates direct feature propagation and reuse across layers. This architecture allows the model to learn more discriminative features from the subtle morphological details of medicinal leaves, albeit at the cost of higher computational demand. In contrast, MobileNetV2 prioritizes efficiency through inverted residual structures and linear bottlenecks, which significantly reduce the parameter count and computational complexity. While this design enables rapid inference suitable for mobile deployment, it slightly compromises the model's capacity to capture highly complex patterns compared to the deeper dense connections. Meanwhile, EfficientNet-B3 achieves a balanced performance by utilizing compound scaling. This approach uniformly scales network depth, width, and resolution, allowing the model to capture rich features without the excessive redundancy found in traditional deep networks.

However, despite these positive results, our research has several limitations that future work should address. First, the classification relies solely on leaf images. While leaves are a primary identification feature, they can be highly variable; incorporating images of flowers, fruits, or stems could further enhance identification accuracy, particularly for species with similar leaf structures. Second, although we used stratified sampling, a potential for sampling bias exists because we collected the dataset from specific regions in Thailand. The model's performance may vary when applied to plants grown in different geographical locations with varying environmental conditions. Finally, the study lacks external validation on a completely independent dataset. While the train-validation split provides internal consistency, testing the model on data collected from different sources or by different researchers is necessary to rigorously verify its generalizability in real-world scenarios.

7. CONCLUSION AND FUTURE WORK

This study evaluated the performance of three state-of-the-art CNN architectures—DenseNet-121, EfficientNet-B3, and MobileNetV2—for the classification of 10 Thai medicinal plant species. The experimental results demonstrated that while all models achieved high classification accuracy (above 93%), they exhibited distinct trade-offs between precision and computational efficiency. DenseNet-121 emerged as the most accurate model (96.0%), making it highly suitable for high-precision tasks where computational resources are not a primary constraint. Conversely, MobileNetV2 proved to be the most efficient architecture, offering a comparable accuracy of 93.4% with the fastest inference time (3.86 s) and the smallest model size (11.07 MB).

These findings have significant practical implications for the deployment of automated plant identification systems. The lightweight nature of MobileNetV2 makes it an ideal candidate for developing offline mobile applications that can assist farmers, herbalists, and the general public in identifying medicinal plants in real-time, even in remote areas with limited internet connectivity. On the other hand, the superior accuracy of DenseNet-121 suggests its potential for integration into centralized herbal databases or botanical inventory systems backed by high-performance computing servers to ensure maximum verification reliability.

Furthermore, the detailed evaluation of precision, recall, and F1-score highlights the models' effectiveness in addressing the critical challenge of morphological similarity among Thai medicinal plants. The consistently high scores across these metrics, particularly for DenseNet-121, demonstrate the architecture's robustness in minimizing misclassification risks. High precision ensures that the system does not mistakenly identify look-alike non-medicinal plants as target herbs (reducing safety risks), while high recall guarantees that it correctly retrieves the actual medicinal species. This balance, reflected in the F1-scores, confirms that the proposed models are not only accurate but also reliable for practical identification tasks where visual ambiguity is a primary concern.

Future research can explore several avenues to further enhance classification performance. First, researchers can investigate state-of-the-art architectures such as Vision Transformers (ViTs) or incorporate attention mechanisms to help the model focus on more subtle morphological features of the leaves. Second, future studies can expand this work by utilizing multimodal data, combining image features with textual descriptions (e.g., habitat, flowering season) or geolocation data to improve identification accuracy in ambiguous cases. Finally, increasing the dataset to cover a wider variety of Thai medicinal species and varying environmental conditions is essential to build a more robust and comprehensive classification system.

AUTHOR CONTRIBUTIONS

Conceptualization, Sompong Valuvanathorn and Chanchai Supaartagorn; methodology, Chanchai Supaartagorn; software, Chanchai Supaartagorn; validation, Sompong Valuvanathorn and Chanchai Supaartagorn; formal analysis, Chanchai Supaartagorn; investigation, Sompong Valuvanathorn and Chanchai Supaartagorn; data curation, Sompong Valuvanathorn and Chanchai Supaartagorn; writing—original draft preparation, Sompong Valuvanathorn and Chanchai Supaartagorn; writing—review and editing, Chanchai Supaartagorn; visualization, Sompong Valuvanathorn and Chanchai Supaartagorn; supervision, Sompong Valuvanathorn. All authors have read and agreed to the published version of the manuscript.

References

[1] W. Ruangaram and E. Kato, "Selection of Thai medicinal plants with anti-obesogenic potential via in vitro methods," *Pharmaceuticals*, vol. 13, no. 4, p. 56, 2020.

[2] R. Geerthana, P. Nandhini and R. Suriyakala, "Medicinal plant identification using deep learning," *International Research Journal of Advanced Science Hub*, vol. 3, no. 05S, pp. 48–53, 2021.

[3] N. Duong-Trung, L. D. Quach, M. H. Nguyen and C. N. Nguyen, "A combination of transfer learning and deep learning for medicinal plant classification," in *Proceedings of the 4th International Conference on Intelligent Information Technology*, pp. 83–90, 2019.

[4] R. Azadnia, M. M. Al-Amidi, H. Mohammadi, M. A. Cifci, A. Daryab and E. Cavallo, "An AI based approach for medicinal plant identification using deep CNN based on global average pooling," *Agronomy*, vol. 12, no. 11, p. 2723, 2022.

[5] R. Azadnia, F. Noei-Khodabadi, A. Moloudzadeh, A. Jahanbakhshi and M. Omid, "Medicinal and poisonous plants classification from visual characteristics of leaves using computer vision and deep neural networks," *Ecological Informatics*, vol. 82, p. 102683, 2024.

[6] S. K. J. Urumarudappa *et al.*, "Development of a DNA barcode library of plants in the Thai Herbal Pharmacopoeia and monographs for authentication of herbal products," *Scientific Reports*, vol. 12, no. 1, 2022.

[7] S. S. Keh, "Semi-supervised Noisy Student pre-training on EfficientNet architectures for plant pathology classification," arXiv preprint, arXiv:2012.00332, 2020.

[8] K. Labrighli, C. Moujahdi, J.E. Oualidi and L. Rhazi, "Artificial intelligence for automated plant species identification: a review," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 10, 2022.

[9] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[10] A. K. Mulugeta, D. P. Sharma and A. H. Mesfin, "Deep learning for medicinal plant species classification and recognition: a systematic review," *Frontiers in Plant Science*, vol. 14, p. 1286088, 2024.

[11] W. Huang and H. Zhang, "Convergence analysis of deep residual networks," *Analysis and Applications*, vol.22, no.2, pp. 351-382, 2023.

[12] A. Dosovitskiy *et al.*, "An Image is Worth 16×16 Words: Transformers for Image Recognition at Scale," *arXiv preprint*, arXiv:2010.11929, 2021.

[13] G. Huang, Z. Liu, L. Van Der Maaten and K. Q. Weinberger, "Densely Connected Convolutional Networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA, pp. 2261-2269, 2017.

[14] M. Tan and Q. Le, "EfficientNet: rethinking model scaling for convolutional neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 8, pp. 6105–6114, 2019.

[15] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. -C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 4510-4520, 2018.

[16] N. A. M. Roslan, N. M. Diah, Z. Ibrahim, Y. Munarko and A. E. Minarno, "Automatic plant recognition using convolutional neural network on Malaysian medicinal herbs: the value of data augmentation," *International Journal of Advanced Intelligent Informatics*, vol. 9, no. 1, pp. 136–144, 2023.

[17] M. T. Ahad, Y. Li, B. Song and T. Bhuiyan, "Comparison of CNN-based deep learning architectures for rice diseases classification," *Artificial Intelligence in Agriculture*, vol. 9, pp. 22–35, 2023.

[18] E. C. Too, L. Yujian, S. Njuki and L. Yingchun, "A comparative study of fine-tuning deep learning models for plant disease identification," *Computers and Electronics in Agriculture*, vol. 161, pp. 272–279, 2019.

Sompong Valuvanathorn is a lecturer at the Department of Mathematics and Digital Technology, Faculty of Science, Ubon Ratchathani University, Thailand. He received his Ph.D. in Information Technology from King Mongkut's University of Technology North Bangkok (KMUTNB) in 2013. His research interests include computer vision, deep learning architectures, and machine learning techniques for intelligent systems and image analysis.

Chanchai Supaartagorn is an Associate Professor in the Faculty of Science, Department of Mathematics and Digital Technology, Ubon Ratchathani University. He received his M.SC. in Computer Science from Mahidol University, Thailand in 1998. His research interests include Machine learning and Data Visualization.