Journal of Applied Data Sciences
Vol. 6, No. 3, September 2025, pp. 1723-1740
ISSN 2723-6471

A New Data Preprocessing Framework to Enhance the Accuracy of Herbal Plants Classification Using Deep Learning

Attapol Kunlerd[1], Atipat Ritthiron[2] , Boonlueo Nabumroong[3], Sakchan Luangmaneerote[4], Anyawee Chaiwachirakhampon[5], Jakkrit Kaewyotha[6,*]

[1,2,3,4,5] *Department of Computer Technology, Rajamangala University of Technology Isan Surin Campus, Thailand*

[6] *College of Computing, Khon Kaen University, Thailand*

(Received: December 1, 2024; Revised: January 20, 2025; Accepted: March 30, 2025; Available online: June 15, 2025)

Abstract

This research proposes to solve the problem of herbal plant classification, which plays a key role in Thai pharmacy and traditional medicine. Moreover, there are limitations due to similar physical characteristics of plants and the reliance on specialists to classify herbal plants, which hinder the utilization of herbal plants by the general public at the local level. To solve this problem, this research presents a new preprocessing framework called P4, which integrates 7 techniques as follow: Image Cropping, Resizing, Normalization (0–1), Data Augmentation, Label Noise, Label Cleaning, and Dataset Quality Score (DQS). The prominent point of P4 technique is the combination of intentional mislabeling and label cleaning process, as well as, quantitative data quality assessment and additional expert review in order to filter out potentially inaccurate data before inputting to Deep Learning model. In the experiment, a dataset of 4,211 herbal images covering 30 herbal plant species is used and compared with 3 proposed techniques in previous research (P1–P3) with 5 deep learning architectures, namely DenseNet201, EfficientNetB7, ViT, Swin Transformer, and ConvNeXt. The experimental results showed that the P4 technique combined with DenseNet201 model provided the highest performance in herbal plant classification, with an Accuracy of 92%, Precision of 92%, Recall of 91%, and a training time of merely 22.92 minutes. This was a result of combining the good data quality from the P4 technique, which enhanced to increase efficiency in producing higher quality and more balanced data. When combined with the structural capability of DenseNet201 that supported feature reuse from previous layers, it increased the robustness to mislabeled data and was able to accurately distinguish plants with similar characteristics. The results of this experiment are able be applied as a guideline for future application in Thai traditional medicine support system and herbal plant learning system.

Keywords: Herbal Plant Classification, Data Preprocessing, Deep Learning, Image Classification

1. Introduction

Thai herbs are perceived as exceedingly valuable resources in both economic and traditional Thai medicine including the body of knowledge that has been continuously transferred and expanded. The World Health Organization and the Department of Thai Planned Medicine and Alternative Medicine has recognized Thai herbs as a considerable component of the public health system that participate in treating diseases and improve the quality of local people's life [1]. Traditional medicine holds considerable value in using herbal plants to cure and maintain communal health of the community. It is the foundation of traditional Thai medicine that has inherited herbal knowledge for generations. It promotes communal self-sufficiency, reduces the use of synthetic medications, and aids in the conservation of herbal plants through usage and cultivation. The traditional Thai medicine has been integrated into the public health system, such as the use of herbs in hospitals through the Royal Thai Traditional Medicine project. It also contributes to spread knowledge and develop community economies through the cultivation and distribution of herbs, making traditional medicine a key mechanism for conserving medicinal plants and promoting the use of resources in the country's health system. Currently, Thai herbs are being developed as portions of the medical and health industry, combining traditional knowledge with modern technology [2], resulting in innovations that meet the requirements of the world population while also serving as a resource for the herbal processing, alternative medical, and biotechnology sectors [3]. However, despite the promotion of Thai herbal medicine as a part of the healthcare system and economy, its application at the

***Corresponding author: Jakkrit Kaewyotha (jakkritk@kku.ac.th)**
DOI: https://doi.org/10.47738/jads.v6i3.733

local level faces many obstacles even now. Especially in the process of plant species identification with similar physical characteristics, this results in classification by conventional methods is tent to be prone to errors. This creates the necessity to rely on experts in order to distinguish medicinal plants. Therefore, opportunities to acquire and utilize herbal knowledge at the community level are still limited [4].

From such study of the solution approach, it was found that previous research has applied Artificial Intelligence (AI) and Deep Learning (DL) technologies, which have garnered a lot of interest in the past several years, especially the use of Convolutional Neural Network (CNN) models to extract and analyze the physical characteristics of plants from pictures [5]. Aside from CNN, there are other models which are available to be used in image classification, such as Vision Transformer (ViT), which is particularly prominent in capturing the spatial structure of pictures [6]. Although CNN and ViT models produce accurate results, there are still significant limitations, including the demand for large data sets, which is a resource and data collection obstacle [7], [8]. Moreover, CNN models are also sensitive to data quality, such as noise and incomplete data [8], [9], [10] and the distribution of imbalanced data [11], [12]. The sensitivity is another factor influencing the learning of model. Given these limitations, this research is dedicated to enhancing the Data preprocessing step, which is an important step to enhance the quality of data before inputting into the model. There is previous research which found that selecting the suitable Data preprocessing technique increased the accuracy and reduced the model error [13], [14].

From the issues and relevant study, it has been applied as a guideline of herbal plant classification method which focuses on appropriate data preparation techniques and comparing between various deep learning models to obtain the most appropriate herbal plant classification system for each problem. This study concentrates on introducing a new data preprocessing technique called p4, which incorporates many techniques, including image cropping, image resizing, normalization (0-1), data augmentation, label noise [15] , label cleaning and DQS [16]. Those are applied along with densenet201 model. When comparing with the data pre-treatment approaches in previous research (P1–P3) which focus on basic techniques such as image resizing, normalization, and addressing data imbalance issues such as SMOTE and oversampling, the P4 framework proposed in this research is more strategic and increasingly attends to data quality. The prominent point of P4, in addition to combining multiple techniques, includes Label Noise Injection in order to test the models' robustness, label cleaning using model loss criteria and expert validation, and dataset quality assessment using DQS technique to systematically analyze data balance and label accuracy. The combination of these techniques creates a "reverse circuit" data preparation process that emphasizes continuous quality improvement. This makes P4 different from traditional approaches in both structure and purpose. Moreover, to evaluate the efficiency of presented P4, P4 is compared to 3 Data preprocessing methods which have been used in the previous study. The 3 methods include P1 (Image Resizing, Data Augmentation, Normalization and One-Hot Encoding) [17], [18], P2 (Image Resizing, Normalization, Data Augmentation and SMOTE) [19], [20] and P3 (Image Resizing, Normalization, Data Augmentation, and Oversampling) [21] In addition, the DenseNet201 model is compared with 4 other deep learning architectures which are EfficientNetB7, ViT, Swin Transformer, and ConvNeXt. This study utilizes 4,211 pictures of dataset which divides the type of herbal plants into 30 types. The model performance evaluation consists of Accuracy, Precision, Recall, and Confusion Matrix values. This study aims to present the performance of P4 Data preprocessing technique into DenseNet201 model by emphasizing on enhancing the accuracy of herbal plants classification and endurance of the model [3].

2. Related work

2.1. Data preprocessing

Previously, many studies have applied preprocessing techniques to advance the performance of Deep Learning and Machine Learning models in image classification and imbalanced data analysis. It typically involves data augmentation techniques such as random flip, shear transformation, zoom, rotation, horizontal/vertical flip, brightness transformation, shift, and shear, which are the processes that applied to maximize the diversity of the dataset and minimize model overfitting [17], [18], [22], [23], [24], [25]. Furthermore, Normalization is also achieved via Min-Max scaling, Z-score normalization, and mean substitution to normalize feature values within a proper range for model learning [17], [19], [26], [27] to correct for data imbalance. From the study, it was found that oversampling techniques such as SMOTE (Synthetic Minority Oversampling Technique), SMOTE-ENN and CTGAN-MOS have been applied.

ISSN 2723-6471

It generated synthetic data for small sample sizes, particularly in plant disease classification, healthcare data analysis, and cyber intrusion detection [16], [19], [26], [27]. In the meantime, there has been also the study that utilized Random Oversampling (ROS) technique to balance the cluster distribution in image datasets. For example, the usage of ROS technique with data augmentation has been applied to develop a classification model for poultry diseases using chicken feces images, which ameliorated the detection accuracy. The study showed that the usage of Data preprocessing technique alongside with others encouraged the model to be able to learn from the dataset which went through the adjustment. It helped in bettering the classification accuracy even under the situations with data limitations and imbalanced class distribution [28].

2.2. Deep Learning Models

DenseNet201, EfficientNetB7, ViT, Swin Transformer and ConvNeXt are Deep Learning models which are widely used for plant classification and plant disease diagnosis. The DenseNet201 model excels in deep feature learning via Dense Connections, which improves the accuracy of tea cultivar classification [29] and herbal plants classification [30]. The result showed that the accuracy rate was as high as 99.64%. In addition, the EfficientNetB7 model which utilized hybrid scaling improved performance in olive cultivar classification [31] and rose leaf disease detection [32] achieving an accuracy greater than 98%. Furthermore, the ViT model, which involved a self-attention mechanism, was ideal for analyzing complex and detailed structures, and the result indicated that it outperformed CNN models on all datasets in rice, corn, and tea disease classification tasks. [33], [34] Moreover, there was Swin Transformer model that served as an improved Transformer model with hierarchical feature representation and sliding window mechanism. The result demonstrated that there was high accuracy in identifying the soybean seedling growth stages [35] and outperformed former CNN model [36], and the ConvNeXt model, an improved CNN model, which was designed to contend with the ViT model, achieved a Top-1 accuracy of 87.80% on ImageNet-1K [36]. Additionally, improvements were provided in MCCM-ConvNeXt, granting a 3.38% increased in accuracy in classifying chili leaf disease compared to the traditional ConvNeXt [37]. These studies accentuated the effectiveness of these models, certifying the improved efficiency and accuracy for applications in further smart agriculture and plant image analysis.

3. Methodology

Figure 1 demonstrates the overview of this study which was implemented in the systematic learning process of machine for herbal plant classification by applying Deep learning. The process consists of data collection, preprocessing, model selection, evaluation, and testing. The details of each process are described as follows:

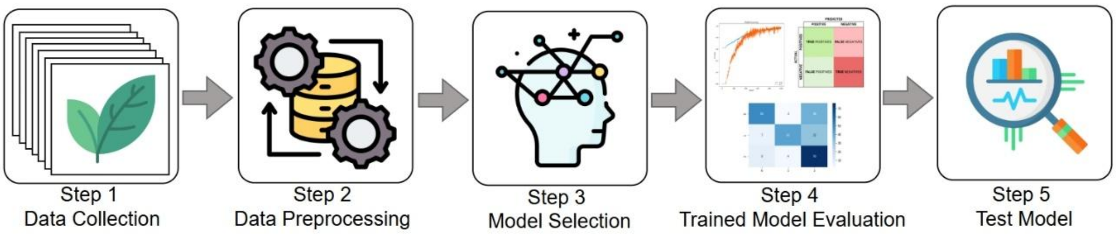


Figure 1. Overview of methodology

3.1. Data Collection

This study is carried out by the team to gather the pictures to support herbal classification. The dataset consists of 4,211 images (resolution 6,000 × 4,000 pixels) taken with a DSLR camera. All pictures are captured under natural light in actual environment. The dataset consists of pictures with 2 types of backgrounds: white background and natural background in actual environment. The dataset includes 30 herbal plants with different physical characteristics (codes of herb class used to develop the model were H1–H30) with varying number of pictures per herb species. Subsequently, the data are divided into training and testing sets in the ratio of 70:30. The example of herbal pictures are illuminated in figure 2.

Figure 2. Example of the herbal plant dataset

3.2. Data Preprocessing

This study targets to propose a data preprocessing framework designed to enhance model performance. The proposed approach integrates different techniques as illustrated in figure 3. The detailed workflow is outlined as follows:

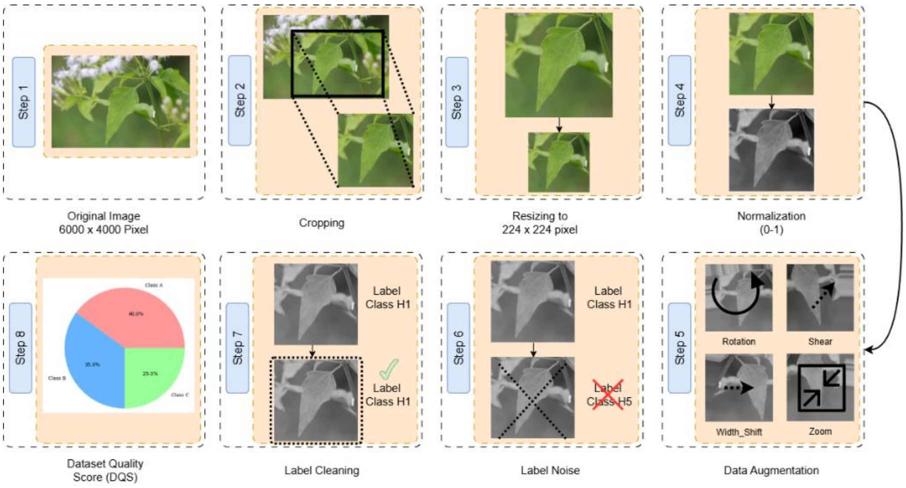


Figure 3. Proposed data preprocessing process (P4)

3.2.1. Image Cropping

The process is to eliminate the unnecessary part out of the picture by keeping only the necessary parts for classification such as leaves or flowers of the herbal plants. This method diminishes the noise and enhances the prominence of salient features which is significant for the model. Then, this allows to learn more about the relevant features. In addition, cropping the picture decreases the data size, resulting in much faster processing by the model and reducing the computational burden in the step of distinguishing irrelevant data [38].

3.2.2. Resizing

This method is to resize the picture to be line with the requirement of Deep Learning model. In this study, image resizing was utilized while maintaining the aspect ratio. This process enhances the image size to be consistent since Deep Learning models require a constant input size. In this study, the pictures are resized to 224×224 pixels. In addition, resizing minimizes the computational burden in terms of memory usage and increases the speed of model training.

3.2.3. Normalization (0-1)

This is a process of adjusting image pixels to the range of [0,1], by converting the original pixels from [0,255] to be compatible with Deep Learning models. This process enhances the stability of the computation and minimizes the data variance. Besides, it as well accelerates the model training because values within the range of [0,1] allow Gradient Descent to perform much better. Equation (1) is applied for the calculation.

$$X_{normailzed} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

3.2.4. Data Augmentation

The process involves the increase of diversity data by modifying the original pictures using various techniques to enable the model to learn from a wider range of samples, decrease overfitting, and increase the generalization ability of models. This study applies image rotation within ±30 degrees, horizontal and vertical shifts of up to 20%, shear up to 20%, internal zoom of 20%, and horizontal flip, with missing pixel values being fulfilled by applying the "nearest" process [39], [40]. The equation (2) is utilized for calculation.

$$I'(x, y) = I\left(round(x), round(y)\right) \quad (2)$$

3.2.5. Label Noise

This process randomly converts the labels (H1-H30) of some data samples to increase the model's robustness to mislabeled data. This study applies the add label noise (labels, noise ratio=0.1) function to randomly select 10 percents of all data samples and replace the labels with classes that are different from the original labels. This process allows the model to learn from the data with noise and reduce the risk of overfitting [15]. The proportion of randomly changed samples is calculated using equation (3).

$$N_{noisy} = N \times r \quad (3)$$

3.2.6. Label Cleaning

This process requires expert's review to correct the potential mislabeled data. The process consists of applying a baseline model and loss-base filtering to identify samples with potential labeling errors by detecting scenarios when the model has poor confidence or cross-entropy loss values that exceed a specified threshold which is calculated by using equation (4). where L represents loss value, y_i represents actual value of the class at i, $\hat{y}_i$ represents the probability predicted by the model and C represents the total number of classes (30 classes in this dataset). In this research, the loss value is calculated for each sample data. Furthermore, mean value (u) and standard deviation (σ) of all loss values in the dataset are calculated to set the standard criteria as $u + \sigma$. Samples with loss values higher than this criterion are considered to have high uncertainty and are forwarded to a botanical expert to examine each herbarium images against a validated database and adjust the labels accordingly. In this experiment, the mean and standard deviation of the loss values were 0.42 and 0.19, respectively, resulting in a screening criterion of 0.61 [41].

$$L = -\sum_{i=1}^{C} y_i \log\left(\hat{y}_i\right) \quad (4)$$

3.2.7. Dataset Quality Score (DQS)

This process includes the quality assessment of the dataset by evaluating the balance of class and the label correctness. Each data sample is evaluated based on Class Balance Score (CBS) [16]. These consist of 3 parts as follows: Class Balance Score (CBS), Label Consistency Score (LCS), and Label Noise Score (LNS). CBS is calculated from equation (5), which is the standard deviation divided by mean of the number of samples in each class and subtracted from 1 to assess inter-class balance. If the dataset has a similar number of samples in each class, the CBS value approaches 1. For example, if $u(n)$=140 and $\sigma(n)$=21 then CBS=1-(21/140) = 0.85. The LCS, utilizing equation (6), is calculated as the ratio of the number of samples correctly predicted by the model, e.g. if the model correctly predicts 3,790 out of 4,211 samples, LCS =3,790/4,211 ≈ 0.90 is obtained Whereas LNS from equation (7) is the proportion of samples that are randomly relabeled to model the noisy data, e.g. LNS= 421/4,211 ≈ 0.10. Lastly, The DQS score is calculated from

ISSN 2723-6471

equation (8) by summing all 3 values, e.g. DQS = (0.85+0.90-0.10)/2 = 0.825, which indicates a high level of quality of the dataset. In general, a DQS value greater than 0.80 indicates balanced and reliable data, while values below 0.50 is capable of reflecting imbalances or label errors that require to be addressed.

$$CBS = 1 - \frac{\sigma\,(n)}{\mu\,(n)} \tag{5}$$

$$LCS = \frac{\sum_{i=1}^{N} 1(\hat{y}_i = y_i)}{N} \tag{6}$$

$$LNS = \frac{N_{noisy}}{N} \tag{7}$$

$$DQS = \frac{CBS + LCS - LNS}{2} \tag{8}$$

3.2.8. Synthetic Minority Over-sampling Technique (SMOTE)

This process is designed for tabular data, but in this work, it is applied to images, to create new samples for a small number of classes. The procedure starts by loading images from the training set and resizing them via an ImageDataGenerator, with pixel values converted to the range [0,1] (as known as Normalization). Furthermore, all image data and labels are extracted from the generator and the labels are converted from one-hot to integer labels for use in SMOTE. Each image is transformed from a 3D structure (224×224×3) to a one-dimensional vector and fed into the SMOTE process to generate new samples from a small number of classes. Afterwards, the SMOTE results are converted back to 224×224×3 images and the labels are converted back to one-hot encoding for further model training [42], [43].

3.2.9. Random Oversampling

It is a technique for balancing a dataset by randomly re-sampling the minority class to equal the majority class. This technique is unable to generate new data, but there is ability to increase the frequency of original data to solve the class imbalance problem. In this experiment, images from the training dataset are loaded and resized with ImageDataGenerator with normalization performed. Then, all image data and labels are then loaded into memory and the labels are converted from one-hot to integer labels so that they are able to be used with RandomOverSampler. Each image is converted to a one-dimensional vector before being fed into an oversampling step to randomly add samples from lesser classes. Afterwards, the results are converted back to a 224 × 224 × 3 image and the labels are converted back to one-hot encoding for further model practice [44].

3.3. Train Model

The structure of the Deep Learning model architecture for herbal plants classification operates through a sequential process. The RGB picture is processed via feature extraction, dimensionality reduction, classification, and fully connected layers to generate the final prediction. Figure 4 demonstrates the structure and the details of model structure in each hierarchy are as follows:

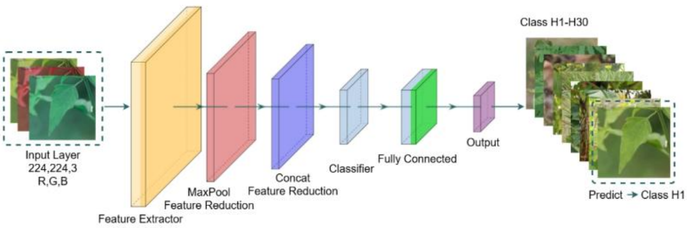


Figure 4. DenseNet201 architecture

3.3.1. Feature Extractor

This research applies the DenseNet201 network which undergoes the training on the ImageNet dataset as a feature extractor, with only the upper layers tuned for herbal plant image classification. Meanwhile, the lower layers are fixed to preserve basic features such as edges, patterns, and shapes, The network consists of a Conv2D layer that applies a small (3×3) kernel to slide over the image to extract spatial features. This is followed by Batch Normalization to adjust the feature values to an appropriate range and ReLU activation, in which Conv2D calculates the feature map values according to Equation (9).

$$O_{i,j} = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} I_{i+m,j+n} \, .K_{m,n} + b \tag{9}$$

3.3.2. Feature Reduction

Feature Reduction is implemented via the Transition Layer, which is interposed between Dense Blocks, with a 1X1 convolution in order to reduce the number of feature channels, followed by a 2x2 AveragePooling2D to reduce spatial size. It decreases model tightness without loss of critical data by utilizing DenseNet, continuous feature connectivity within each Dense Block. the results from all previous layers in the same block are taken to combined with Concatenation type. This combination of features is calculated by using Equation (10).

$$F_{concat} = [F_1, F_2, \dots, F_n] \tag{10}$$

3.3.3. Classifier

This step utilizes GlobalAveragePooling2D to convert the spatial feature map from the final layer of DenseNet201 into a smaller statistical vector without relying on a large number of Fully Connected layers. The result vector undergoes a dropout process at 0.5 followed by a 256-unit Dense layer that implements ReLU stimulation function and enters the last layer that uses Softmax in order to classify herbal images into 30 categories by calculating the average value of each channel in the GlobalAveragePooling process from Equation (11).

$$G_c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} F_c \, (i, j) \tag{11}$$

3.3.4. Fully Connected

After feature downsizing with GlobalAveragePooling2D, the data is fed into the 256-unit Dense Layer, which serves as a Fully Connected layer by converting the feature vector to suit the final classification. Each neuron computes the output value by weighting the inputs and biasing them according to equation (12). Furthermore, the result is transmitted via the ReLU function, which enhances nonlinearity according to equation (13) and finally, the resulted value enters the dropout layer with a rate of 0.5 to disable some random neurons during the model's ability to infer results according to equation (14).

$$z_i = \sum_{j=1}^{n} w_{ij} \, x_i + b_i \tag{12}$$

$$a_i = \max\,(0, z_i) \tag{13}$$

$$\tilde{a}_i = \begin{cases} 0, \; if \; r_i < p \\ a_i \; if \; r_i \geq p \end{cases} where \; r_i \sim u(0,1) \tag{14}$$

3.3.5. Output

This Output is the final process of model, which applies a Dense Layer with Softmax activation function to classify the herb pictures into 30 classes. The output values from the Fully Connected Layer are converted to class probabilities via the Softmax function to ensure that all values are in the range of [0,1] and the sum is equal to 1. This allows the model to accurately determine the probability of each class. Equation (16) is used for calculation, and the class with the highest probability is selected as the final output.

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{30} e^{z_j}} \tag{16}$$

3.3.6. Hyperparameter Configuration

Table 1 illuminates Hyperparameters which is specified in the training of 5 models for herbal plants classification. Such Hyperparameters are the values which directly affects to the performance and learning process of the models.

ISSN 2723-6471

Table 1. Model Training Hyperparameters

No	Hyperparameter	Description	Current Value
1	Input Size	The size of the input image fed into the model.	224 x 224, 3
2	Batch Size	Number of images processed together in one training step.	32
3	Optimizer	Algorithm used to adjust model weights to minimize loss.	Adam
4	Loss	Function that measures difference between predicted and actual values.	Categorical Cross-Entropy
5	Learning Rate	Step size that determines how much the model updates weights in each iteration.	0.00001
6	Epochs	Number of times the entire dataset is passed through the model during training.	50
7	Dropout Rate	Fraction of neurons randomly disabled during training to prevent overfitting.	0.5

3.4. Evaluation Metrics

Evaluation Metrics for the classification models are as follows: Accuracy, which represents the proportion of samples that the model is able to correctly classify relative to the total number of samples associated with the positive class, both correctly predicted (TP predicts yes and actually yes) and incorrectly predicted (FP predicts yes but actually no and FN predicts no but actually yes), is calculated using Equation (17), and example of correctly predicted (TP) image is shown in figure 5. Precision is the proportion of the number of samples that the model correctly predicts as positive compared to the total number of samples that the model predicts as positive, calculated using Equation (18); Precision which is the proportion of sample number that the model correctly predicts as positive compared to the total number of samples that the model predicts as positive, calculated by using Equation (18); Recall which is the proportion of total number of positive samples correctly classified by the model compared to the total number of true positive samples and is calculated by using Equation (19); and Confusion Matrix, which is presented as a table for evaluating the performance of classification models by comparing actual and predicted labels.

$$Accuracy = \frac{TP}{TP+FP+FN} \tag{17}$$

$$Precision = \frac{TP}{TP+FP} \tag{18}$$

$$Recall = \frac{TP}{TP+FN} \tag{19}$$

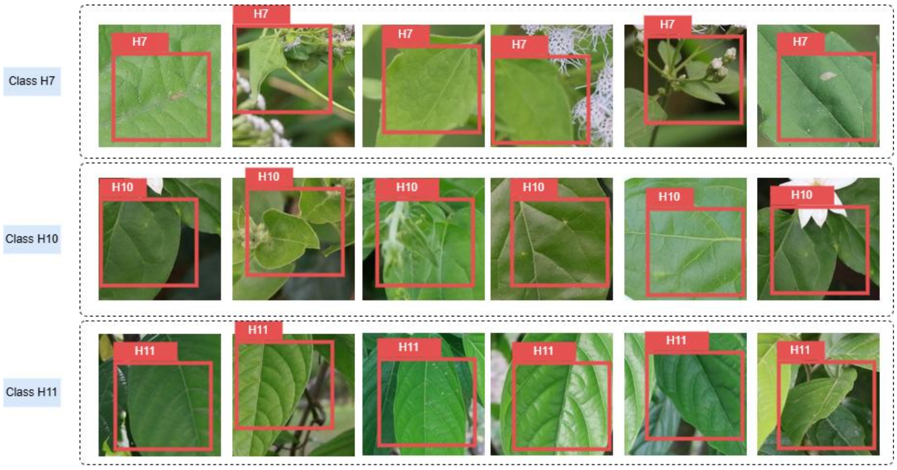


Figure 5. True positive classification for herbal plant identification.

4. Results and Discussion

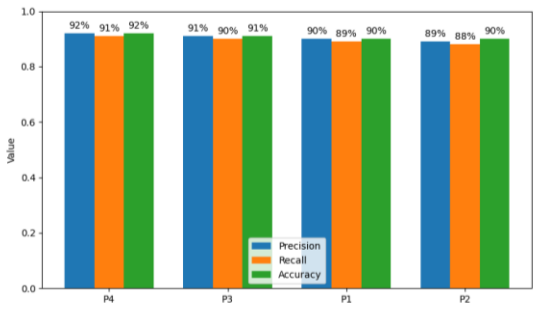
The results are divided into 2 parts as follows: 4.1) Impact of Data Preprocessing on the model performance and; 4.2) the comparative results of the model performance. The details of the experimental results are presented in the following sections:

Journal of Applied Data Sciences
Vol. 6, No. 3, September 2025, pp. 1723-1740

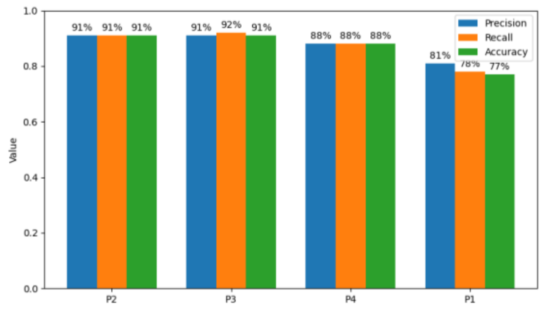
4.1. Impact of Data Preprocessing

The impact of Data Preprocessing on the models' performance displays in figure 6. The results are showed that the selection of 4 Data Preprocessing methods (P1-P4) directly affected to accuracy and capability of models' learning for herbal plants classification. Therefore, the results are concluded as follows:

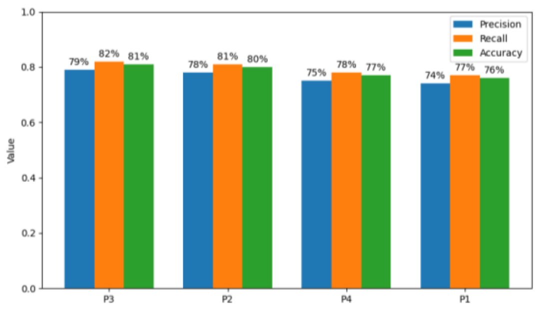
Preprocessing method P1 consists of Image Resizing, Data Augmentation, Normalization, and One-Hot Encoding. This creates an impact on DenseNet201 and Swin Transformer models. Especially, DenseNet201 with Dense Connectivity structure [45] is capable of making use of the enhanced data [46]. This allowed the test accuracy to be 90% show in figure 6 (a) while Swin Transformer was at 76% which caused by the exceeding requirement of data variety show in figure 6 (c) [47]. Nevertheless, EfficientNetB7 and ViT were negatively affected which EfficientNetB7 achieved the test accuracy only at 58%. This showed that the model was unable to take advantage of the addition of fake data show in figure 6 (d). ViT model achieved the test accuracy of 77%, however, it was still lower than P3 or P4 with better oversampling and label cleaning steps show in figure 6 (b). Finally, the ConvNeXt model had the lowest test accuracy of 7%, indicating that the model structure was unable to utilize the data augmentation featured as well as other CNNs or Transformer models. In the context of this study show in figure 6 (e), it was concluded that P1 was suitable for DenseNet201 and Swin Transformer, but not for EfficientNetB7, ViT, and ConvNeXt.



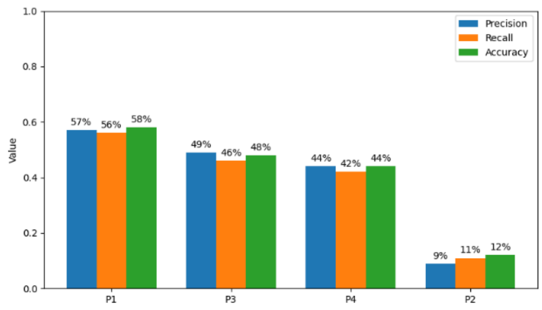
(a) DenseNet201



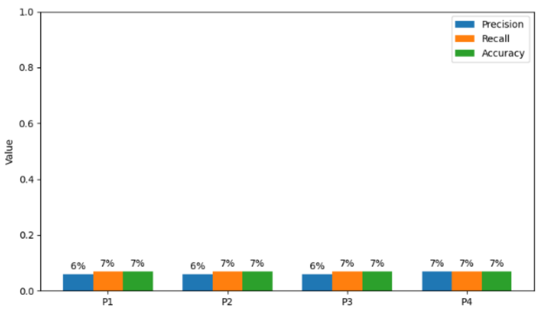
(b) ViT



(c) Swin Transformer



(d) EfficientNetB7



(e) ConvNeXT

Figure 6. Performance Comparison Across Data Preprocessing Stages for Each Model

Preprocessing method P2 involves Image Resizing, Normalization, Data Augmentation, and SMOTE techniques with different models, giving a variety of results. DenseNet201 achieved high accuracy 90% show in figure 6 (a) and ViT achieved the highest accuracy of 91%, indicating that both structures were able to support the augmentation of artificial data well [48] show in figure 6 (b). Meanwhile, the swing transformer was at 80% show in figure 6 (c), however, EfficientNetB7 dropped to 12%, indicating a difference between oversampling techniques and model design show in figure 6 (d). Ultimately, the ConvNeXt model achieved the lowest accuracy of 7% show in figure 6 (e), suggesting that oversampling and data augmentation methods impacted the model's capacity to identify significant features [49] show in figure 6 (e).

Preprocessing method P3 includes Image Resizing, Normalization, Augmentation, and Oversampling techniques for preprocessing data before inputting data from the model. The results indicated that DenseNet201 and ViT fully utilize these technologies, with test accuracy of 91% show in figure 6 (a) and (b). For Swin Transformer section, the test accuracy was at 81% show in figure 6 (c), but EfficientNetB7 merely achieved 48% show in figure 6 (d). These test results showed that the simulation architecture was believed that it responded as well to Oversampling and Augmentation techniques as other architectures [50]. The ConvNeXt model had the lowest accuracy of 7% show in figure 6 (e), demonstrating its limitations in using artificial data [49]. Conversely, CNNs using Dense and Transformer demonstrated higher performance in extracting features from auxiliary data [51].

Preprocessing method P4, proposed in this study, contains Image Cropping, Resizing, Normalization (0-1), Data Augmentation, Label Noise, Label Cleaning and DQS assessment for data preparation before model training. It was found that DenseNet201 achieved the highest accuracy of 92% show in figure 6 (a). This result emphasized its effectiveness in taking benefit from the enhanced data quality via both of Label Cleaning and Data Augmentation [52]. Even though, ViT had a high accuracy of 88% with similar accuracy show in figure 6 (b), Swin Transformer had an accuracy of 77%, which was lower than other data preparation techniques but still satisfactory show in figure 6 (c). The EfficientNetB7 model achieved an accuracy of 44% show in figure 6 (d), which showed the limitation of using data processed using Labeled Noise and Augmentation methods. Finally, the ConvNeXt model achieved the lowest accuracy of 7% show in figure 6 (e), indicating a mismatch between the model architecture and the label tuning and cleaning steps, as well as the formation of Label Noise, which is capable of having even more negative impact on ConvNeXt's learning than other models.

The overall test results indicated that the proposed P4 data preparation process was the most suitable for DenseNet201, achieving the highest accuracy of 92%. Although ViT and Swin Transformer outperformed the P3 data preparation process 91% and 81%, respectively, EfficientNetB7 was the most proper for P1, achieving an accuracy of 58%, outperforming the other processes. Ultimately, in the case of ConvNeXt, despite its low performance in all data preparation processes, P4 achieved the highest Precision, Recall and Accuracy = 7% compared to the other processes. This was possibly because the sequential convolutional learning structure was unable to handle data with similar visual characteristics, such as groups of herbs with similar shapes. Therefore, the future research should consider learning approaches that focus on comparing the relationships between images. Figure 7 demonstrates the models' execution time in minute unit of the 5 models under 4 preprocessing conditions. The performance values of Precision, Recall, and Accuracy for each condition are shown in table 2.

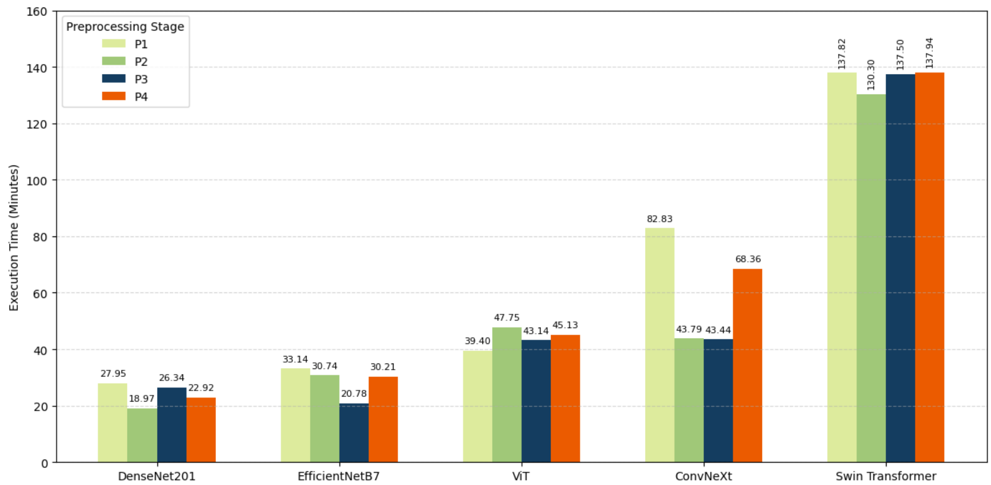


Figure 7. Execution time comparison of 5 models across four data preprocessing stages (P1–P4)

 ISSN 2723-6471

DenseNet201 was the best performing model, especially under P4, with an Accuracy of 92% for only 22.92 minutes of training and a good balance between accuracy and time cost. ViT performed less well, especially P2 and P3, with an Accuracy of 91% for 43–47 minutes of training. However, in P4, the performance dropped to 88%, possibly due to the effects of overfitting or over-modified label data. Although Swin Transformer achieved a relatively good maximum accuracy of 81% in P3, it was the model that took the longest training time which was approximately 137 minutes. Across all preprocessing runs, EfficientNetB7 showed inconsistent results, with a maximum Accuracy of only 58% in P1 and decreasing to only 12% in P2, even at a moderate training time of 30 minutes. This demonstrated that the model was sensitive to changes in preprocessing. Ultimately, ConvNeXt was the model with the lowest classification performance with an Accuracy of only 7% in all preprocessing stages. This indicated a learning problem of the model. Even with a training time of up to 82 minutes in some cases, its performance still was not improved.

Table 2. Comparison of performance across 5 models and 4 data preprocessing

Model	Preprocessing	Precision	Recall	Accuracy	Time Execution (Min)
DenseNet201	P1	0.90	0.89	0.90	27.95
	P2	0.89	0.88	0.90	18.97
	P3	0.91	0.90	0.91	26.34
	P4	**0.92**	**0.91**	**0.92**	**22.92**
EfficientNetB7	P1	0.57	0.56	0.58	33.14
	P2	0.09	0.11	0.12	30.74
	P3	0.49	0.46	0.48	20.78
	P4	0.44	0.42	0.44	30.21
ViT	P1	0.81	0.78	0.77	39.40
	P2	0.91	0.91	0.91	47.75
	P3	0.91	0.92	0.91	43.14
	P4	0.88	0.88	0.88	45.13
Swin Transformer	P1	0.74	0.77	0.76	137.82
	P2	0.78	0.81	0.80	130.30
	P3	0.79	0.82	0.81	137.50
	P4	0.75	0.78	0.77	137.94
ConvNeXt	P1	0.06	0.07	0.07	82.83
	P2	0.06	0.07	0.07	43.79
	P3	0.06	0.07	0.07	43.44
	P4	0.07	0.07	0.07	68.36

4.2. Model Performance Comparison Results

This section compares the performance of different Deep Learning models on herbal plant classification. The analysis covers Validation Loss, Test Accuracy, Time Execution, and Confusion Matrix evaluation. The details of the experimental results at each step are presented as follows:

4.2.1. Training and Testing Loss for Each Model

DenseNet201 earned the lowest loss (training loss = 0.05, testing loss = 0.33), indicating that the model was capable of good learning and testing ability [53]. Meanwhile, the ViT and Swin Transformer models achieved balanced loss values (ViT: 0.40, 0.29, Swin Transformer: 0.58, 0.62), indicating that no overfitting or underfitting problem was available at this point. On the contrary, EfficientNetB7 achieved higher testing loss compared to the training loss (0.92, 1.63), which stood a chance of overfitting. Meanwhile, ConvNeXt achieved the highest loss values (3.34, 3.34), which could be the result of underfitting or inappropriate model structure for the dataset in this study. The experimental results concluded that DenseNet201 was the most effective model for herbal plants classification in this study. The experimental results are presented in figure 8.

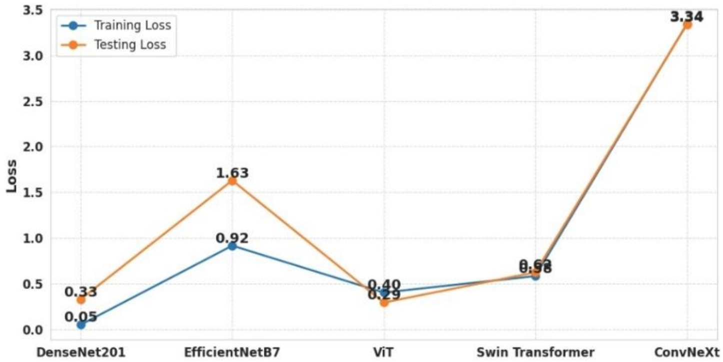


Figure 8. Training and Validation Loss for Each Model

4.2.2. Comparison of Model Test Accuracy and Execution Time

Figure 9 illuminates the model performance in terms of Accuracy and Execution Time. The experimental results indicated that DenseNet201 was the most cost-effective model with an accuracy of 92% but only 22.92 minutes of training time, which was possibly due to the Dense Connectivity structure, which enabled the model to learn and distinguish complex characteristics of similar herbal plants well [46].

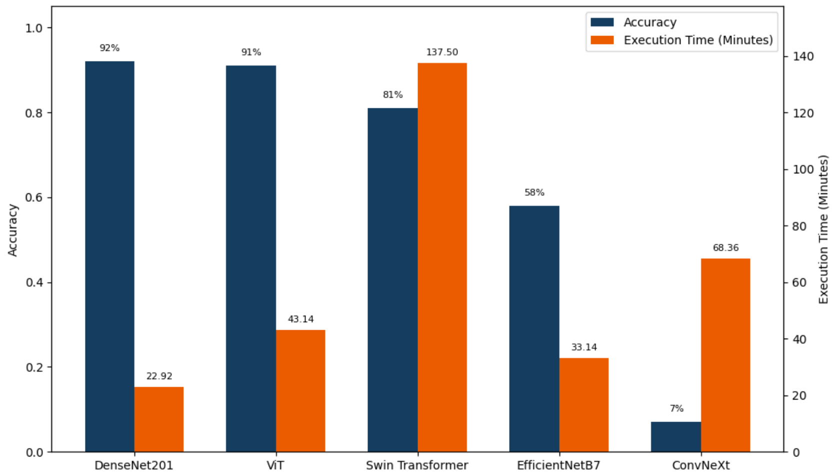


Figure 9. Comparison of Test Accuracy and Execution Time Across Five Models

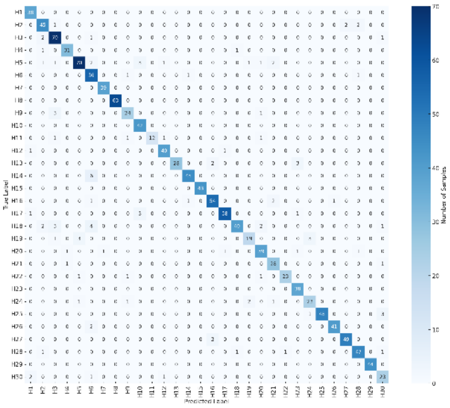
Since the model architecture was able to pass through and reuse features from all previous layers at any level, this dense connectivity results in a model that was robust to label noise by relying on redundant feature paths to compensate for skewed label information. When combined with label cleaning process and using balanced data, DenseNet201 was able to extract features with consistency and good discrimination between each class [54]. ViT resulted with similarity which showed an Accuracy of 91%, but took longer to train at 43.14 minutes, due to the combination of pixel-level image adjustment and data balance. Applying the Random Oversampling technique reduced the problem of class imbalance, it allowed the Self-Attention mechanism to distribute the focus to data from all classes equally, without biasing towards the class with more data. The diverse availability and balanced data, which was augmented and oversampled, enhanced the strengths of the ViT architecture in dealing with complex image structural features such as herbaceous leaf patterns [55], [56], [57]. Conversely, Swin Transformer exhibited a large cost that was incongruous with the results gained, taking 137.50 minutes to complete, despite its moderate accuracy of 81%. Meanwhile, EfficientNetB7 achieved an accuracy of 58% in 33.14 minutes. Despite training time being similar to the other models, the accuracy was low by dropping to only 12% in P2 using the SMOTE technique. This reduction indicated the sensitivity of EfficientNetB7 was compound scaling architecture to unrealistic or structurally discontinuous image data. Finally, ConvNeXt was the poorest performing model, with an accuracy of only 7% but a training time of 68.36 minutes. This was likely due to the model’s tendency to rely more on texture bias than shape, which prevented it from

ISSN 2723-6471

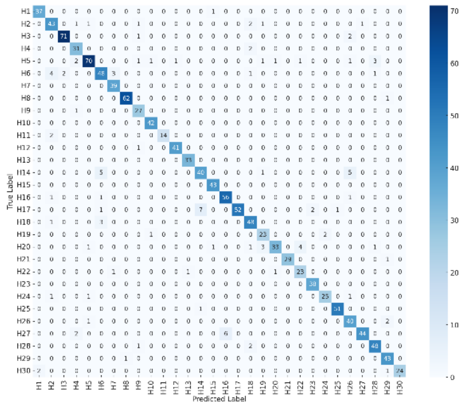
differentiating across classes similar characteristics. Additionally, ConvNeXt was designed to work on large datasets such as ImageNet, so it may not be appropriate for small datasets with limited images per class [58], [59].

4.2.3. Confusion Matrix analysis

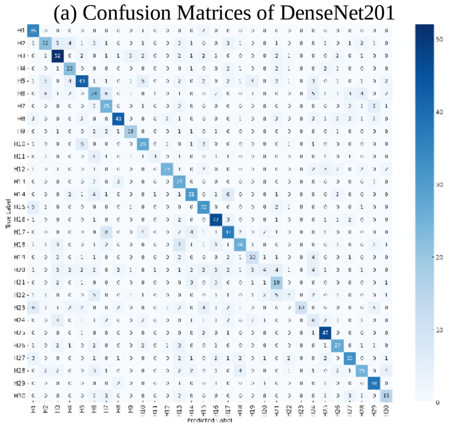
From figure 10, the Confusion Matrix shows the results of the herb classification accuracy test. The number of samples that the model classified into each class is displayed in each column. The diagonal column from the top left to the bottom right displays the number of samples that the model classified correctly (True Positive). In this experiment, there were 30 classes of herb images, each class had a different number of images.



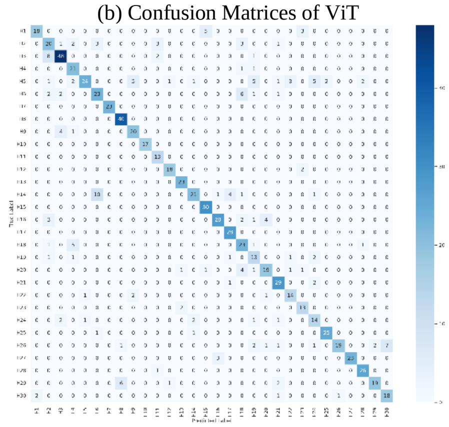
(a) Confusion Matrices of DenseNet201



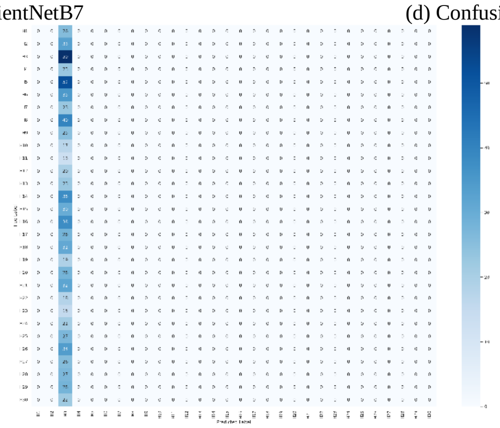
(b) Confusion Matrices of ViT



(c) Confusion Matrices of EfficientNetB7



(d) Confusion Matrices of Swin Transformer



(e) Confusion Matrices of ConvNeXT

Figure 10. Confusion Matrices of the Five Models

The overall herb image classification performance of the 5 models was able to be described as follows: From the analysis of the data of all 5 models, it showed that the 5 classes that was possible to be classified most accurately overall are H3, H15, H25, H13 and H8, with an average accuracy in the range of 73–88%. On the other hand, the most frequently misclassified class was H20, misclassified into classes H25, H26, H28 and H29, which recurs in many models, such as DenseNet201, ViT and EfficientNetB7. This suggested that the H20 class dataset from similar classes

 ISSN 2723-6471

should be improved. The large number of misclassifications, especially in EfficientNetB7 and ConvNeXt, which tended to highly predict H20 class as other classes, reflected the limitation of the model in recognizing the characteristics of herbs in this class with very similar physical characteristics. Therefore, deep image feature analysis in the similar classes should be performed to improve the accuracy in the classification process.

5. Conclusion

This study proposed the framework of P4 Data Preprocessing to enhance the performance of Deep Learning models in classification of 30 herbal plants. Such P4 consists of Image Cropping, Resizing, Normalization (0-1), Data Augmentation, Label Noise, Label Cleaning and DQS techniques. The performance of P4 is evaluated against the preprocessing techniques used in previous studies (P1-P3) and tested against 5 Deep Learning architectures: DenseNet201, EfficientNetB7, ViT, Swin Transformer, and ConvNeXt. The experimental results were found that P4 combined with DenseNet201 delivered the best results with a maximum accuracy of 92% and the lowest testing loss of 0.33 compared to other data processing techniques and models. This study emphasized the significant role of P4 in enhancing the efficiency of Data Preprocessing and Deep Learning model techniques for herbal plant classification. The usage of appropriate Data Preprocessing technique not only improved the accuracy but also enabled the model to manage with the data limitations and characteristics of herbal plants. The results showed that although data preprocessing improved the accuracy of herbal image classification in the DenseNet201 model, the same effect was unavailable as all other models, especially ConvNext, which exhibited low performance in all data preprocessing techniques. Furthermore, there was significant limitation which was class classification of plant with very similar physical characteristic. It then showed that most of the models tended to incorrectly predict for such class. Therefore, future research should focus on developing data preprocessing techniques that take into account the specific characteristics of each type of model architecture, and designing contrastive learning or class-wise embedding methods to systematically and specifically enhance the ability to distinguish similar classes.

6. Declarations

6.1. Author Contributions

Conceptualization: A.K., A.R.; Methodology: A.K., B.N., J.K.; Software: A.K., B.N., J.K.; Validation: A.K., J.K.; Formal Analysis: S.L., A.C.; Investigation: A.K., A.R.; Resources: S.L., A.C.; Writing – Original Draft Preparation: A.K., J.K.; Writing – Review & Editing: J.K.; All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding

This research was successfully conducted with the generous funding support from the Department of Thai Traditional and Alternative Medicine under the fiscal year 2020. The research funding was provided under Contract No. DTAM. 11/2563, dated January 30, 2020.

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] C. Nangsue, P. Srathong, K. Choowongkomon, K. Promnikorn, and C. Tocharus, "Anticancer Activity of the Thai Herbal Formula 'Kerra' Versus Afatinib and Doxorubicin In Vitro," *WSEAS Transactions on Biology and Biomedicine*, vol. 21, no. 4, pp. 155–161, 2024, doi: 10.37394/23208.2024.21.16.

[2] S. Pinmongkholgul, "Cultivation and production for enhanced commercial and pharmaceutical applications of Kariyat (Andrographis paniculata) products, Phayao, Thailand," *Pakistan Journal of Life and Social Sciences*, vol. 22, no. 2, pp. 3530–3538, 2024, doi: 10.57239/PJLSS-2024-22.2.00260.

[3] A. Singh and P. S. Negi, "Biotechnological application of health-promising bioactive compounds," *in Biotechnological intervention in production of bioactive compounds*, J. Devi, Ed., in Sustainable Landscape Planning and Natural Resources Management. , Cham: Springer, 2025, vol. 25, no. 2, pp. 73–94. doi: 10.1007/978-3-031-76859-0_5.

[4] R. Azadnia, F. Noei-Khodabadi, A. Moloudzadeh, and A. Jahanbakhshi, "Medicinal and poisonous plants classification from visual characteristics of leaves using computer vision and deep neural networks," *Ecological Informatics*, vol. 82, no. 24, pp. 1-12, 2024, doi: 10.1016/j.ecoinf.2024.102683.

[5] D. Vinod et al., "An Android App for Ayurvedic Herb Identification and Classification Using Machine Vision," *in Proceedings of the 1st International Conference on Trends in Engineering Systems and Technologies (ICTEST)*, Kochi, India, 2024, vol. 24, no. 7, pp. 1–6. doi: 10.1109/ICTEST60614.2024.10576106.

[6] A. Garcia-Perez, M. J. Gomez-Silva, and A. de la Escalera-Hueso, "A generative AI approach to improve in-situ vision tool wear monitoring with scarce data," *Journal of Intelligent Manufacturing*, vol. 24, no. 5, pp. 1-19, 2024, doi: 10.1007/s10845-024-02379-2.

[7] X. Zhao, L. Wang, Y. Zhang, and et al, "A review of convolutional neural networks in computer vision," *Artificial Intelligence Review*, vol. 57, no. 3, p. 1-43, 2024, doi: 10.1007/s10462-024-10721-6.

[8] G. Cicceri, S. Di Bella, S. Di Franco, G. Stassi, M. Todaro, and S. Vitabile, "Deep learning approaches for morphological classification of intestinal organoids," *IEEE Access*, vol. 13, no. 4, pp. 62267-62287, 2025, doi: 10.1109/ACCESS.2025.3558621.

[9] L. Baloch, A. Sajid, C. Dewi, H. Christanto, and A. Zafar, "Deep Learning-Based Medical Image Reconstruction: Overview, Analysis, and Challenges," *Revue d'Intelligence Artificielle*, vol. 38, no. 2, pp. 417–429, Apr. 2024, doi: 10.18280/ria.380205.

[10] T. H. Kim, R. Chinthaginjala, A. Srinivasulu, and others, "COVID-19 health data prediction: a critical evaluation of CNN-based approaches," *Scientific Reports*, vol. 15, no. 3, pp. 1-13, 2025, doi: 10.1038/s41598-025-92464-0.

[11] K. Nawaz, A. Zanib, I. Shabir, and et al, "Skin cancer detection using dermoscopic images with convolutional neural network," *Scientific Reports*, vol. 15, no. 3, pp. 1-31, 2025, doi: 10.1038/s41598-025-91446-6.

[12] R. Roslidar, Y. Yunida, and M. Jurej, "Improved lung sound classification model using combined residual attention network and Vision Transformer for limited dataset," *Indonesian Journal of Electrical Engineering and Informatics*, vol. 12, no. 1, pp. 102–113, 2024.

[13] H. Rabie and M. A. Akhloufi, "A review of machine learning and deep learning for Parkinson's disease detection," *Discovery Artificial Intelligence*, vol. 5, no. 3, pp. 1-17, 2025, doi: 10.1007/s44163-025-00241-9.

[14] J. Lu, K. Wang, H. Ding, and et al, "MSCA-Sp R-CNN: a segmentation algorithm for pneumonia small lesions integrating multi-scale channel attention and sub-pixel upsampling," *Multimedia Systems*, vol. 31, no. 3, pp. 1-13, 2025, doi: 10.1007/s00530-025-01726-4.

[15] T. Burgert, M. Ravanbakhsh, and B. Demir, "On the Effects of Different Types of Label Noise in Multi-Label Remote Sensing Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, no. 12, pp. 1-13, 2022.

[16] A. Majeed and S. O. Hwang, "CTGAN-MOS: Conditional Generative Adversarial Network Based Minority-Class-Augmented Oversampling Scheme for Imbalanced Problems," *IEEE Access*, vol. 11, no. 8, pp. 37420–37435, 2023, doi: 10.1109/ACCESS.2023.3303509.

[17] A. Mouri Zadeh Khaki and A. Choi, "Optimizing Deep Learning Acceleration on FPGA for Real-Time and Resource-Efficient Image Classification," *Applied Sciences*, vol. 15, no. 1, pp. 1-13, 2025, doi: 10.3390/app15010422.

[18] R. Sujatha, J. M. Chatterjee, N. Z. Jhanjhi, and S. N. Brohi, "Performance of Deep Learning vs Machine Learning in Plant Leaf Disease Detection," *Microprocessors and Microsystems*, vol. 80, no. 2, pp. 1-11, 2021, doi: 10.1016/j.micpro.2020.103615.

[19] C. Zhou, Y. Xie, W. Wang, and Y. Zheng, "Machine Learning Driven Post-Impact Damage State Prediction for Performance-Based Crashworthiness Design of Bridge Piers," *Engineering Structures*, vol. 292, no. 10, pp. 1-15, 2023, doi: 10.1016/j.engstruct.2023.116539.

[20] A. R. Palakayala and P. Kuppusamy, "A qualitative and quantitative approach using machine learning and non-motor symptoms for Parkinson's disease classification: A hierarchical study," *Applied Computer Science*, vol. 20, no. 3, pp. 171–191, 2024, doi: 10.35784/acs-2024-35.

[21] C. Mo, K. Huang, H. Ji, W. Li, and K. Xu, "Efficient intelligent fault diagnosis method and graphical user interface development based on fusion of convolutional networks and vision transformers characteristics," *Scientific Reports*, vol. 15, no. 2, pp. 1-18, 2025, doi: 10.1038/s41598-025-88668-z.

[22] R. Bandla, S. K. M. Hattaraki, J. S. Priyanka, and Y. C. Dumpali, "Plant Identification and Analysis of Medicinal Properties Using Image Processing and CNN with MobileNetV2 and AlexNet," *in IEEE INNOVA 2024 Conference*, vol. 2024, no. 1, pp. 1-5, 2024. doi: 10.1109/INNOVA63080.2024.10847039.

[23] P. B. Vikhe and B. L. Gunjal, "Capsicum Leaf Disease Detection Using Different Learning Techniques," *in Proceedings of the International Conference on IoT Based Control Networks and Intelligent Systems (ICICNIS-2024), IEEE*, vol. 2024, no. 1, pp. 1742-1745, 2024. doi: 10.1109/ICICNIS64247.2024.10823333.

[24] H. Tao, "Smoke Recognition in Satellite Imagery via an Attention Pyramid Network With Bidirectional Multilevel Multigranularity Feature Aggregation and Gated Fusion," *IEEE Internet of Things Journal*, vol. 11, no. 8, pp. 5821–5837, 2024, doi: 10.1109/JIOT.2023.3339476.

[25] S. Islam et al., "BDMediLeaves: A Leaf Images Dataset for Bangladeshi Medicinal Plants Identification," *Data in Brief*, vol. 50, no. 10, pp. 1-13, 2023, doi: 10.1016/j.dib.2023.109488.

[26] H. M. Balaha, E. M. El-Gendy, and M. M. Saafan, "DMDRDF: Diabetes Mellitus and Retinopathy Detection Framework Using Artificial Intelligence and Feature Selection," *Soft Computing*, vol. 28, no. 10, pp. 11393–11420, 2024, doi: 10.1007/s00500-024-09873-y.

[27] D. Herzalla, W. T. Lunardi, and M. Andreoni, "TII-SSRC-23 Dataset: Typological Exploration of Diverse Traffic Patterns for Intrusion Detection," *IEEE Access*, vol. 11, no. 9, pp. 42562–42578, 2023, doi: 10.1109/ACCESS.2023.3319213.

[28] M. Z. Degu and G. L. Simegn, "Smartphone-Based Detection and Classification of Poultry Diseases from Chicken Fecal Images Using Deep Learning Techniques," *Smart Agricultural Technology*, vol. 4, no. 8, pp. 1-9, 2023, doi: 10.1016/j.atech.2023.100221.

[29] Z. Zhang et al., "Identification of tea plant cultivars based on canopy images using deep learning methods," *Scientia Horticulturae*, vol. 339, no. 1, pp. 1-13, 2025, doi: 10.1016/j.scienta.2024.113908.

[30] B. Dey and others, "Assessing deep convolutional neural network models and their comparative performance for automated medicinal plant identification from leaf images," *Heliyon*, vol. 10, no. 1, pp. 1-15, 2024, doi: 10.1016/j.heliyon.2023.e23655.

[31] J. Mendes, J. Lima, L. Costa, N. Rodrigues, and A. I. Pereira, "Deep learning networks for olive cultivar identification: A comprehensive analysis of convolutional neural networks," *Smart Agricultural Technology*, vol. 8, no. 8, pp. 1-13, 2024, doi: 10.1016/j.atech.2024.100470.

[32] S. Poornima, R. Divya, and R. S. Krishnan, "Advancing Rose Disease Diagnosis: A Deep Learning Framework with EfficientNet-B7," *in IEEE Conference on Intelligent Systems*, vol. 2024, no. 8, pp. 1747-1754, 2024. doi: 10.1109/IS-2024.10714594.

[33] E.-T. Baek, "Attention Score-Based Multi-Vision Transformer Technique for Plant Disease Classification," *Sensors*, vol. 25, no. 1, pp. 1-14, 2025, doi: 10.3390/s25010270.

[34] Q. Wang, Y. Dong, N. Xu, F. Xu, C. Mou, and F. Chen, "Image Classification of Tree Species in Relatives Based on Dual-Branch Vision Transformer," *Forests*, vol. 15, no. 12, pp. 1-13, 2024.

[35] K. Ma et al., "Research on Soybean Seedling Stage Recognition Based on Swin Transformer," Agronomy, vol. 14, no. 11, pp. 1-23, 2024, doi: 10.3390/agronomy14112614.

[36] Z. Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," *in IEEE/CVF International Conference on Computer Vision (ICCV)*, vol. 2022, no. 2, pp. 9992-10002, 2022. doi: 10.1109/ICCV48922.2021.00986.

[37] D. Li, C. Zhang, J. Li, M. Li, M. Huang, and Y. Tang, "MCCM: Multi-scale feature extraction network for disease classification and recognition of chili leaves," *Frontiers in Plant Science*, vol. 15, no. 5, pp. 1-19, 2024, doi: 10.3389/fpls.2024.1367738.

[38] P. Sajitha, A. D. Andrushia, N. Anand, and M. Z. Naser, "A review on machine learning and deep learning image-based plant disease classification for industrial farming systems," *Journal of Industrial Information Integration*, vol. 38, no. 3, pp. 1-18, 2024, doi: 10.1016/j.jii.2024.100572.

[39] H. El Akhal, A. Ben Yahya, and A. El Belrhiti El Alaoui, "Positive discrimination of minority classes through data generation and distribution: A case study in olive disease classification," *Engineering Applications of Artificial Intelligence*, vol. 139, no. 1, pp. 1-13, 2025, doi: 10.1016/j.engappai.2024.109646.

[40] M. Rezaei, S. Gupta, D. Diepeveen, H. Laga, M. G. K. Jones, and F. Sohel, "Barley disease recognition using deep neural networks," *European Journal of Agronomy*, vol. 161, no. 9, pp. 1-11, 2024, doi: 10.1016/j.eja.2024.127359.

[41] F. Nguleni, D. Nyambo, J. Lisuma, and S. Kaijage, "Dataset of Virginia Flue-cured Tobacco Leaf images based on stalk leaf position for classification tasks: A case of Tanzania," *Data in Brief*, vol. 56, no. 10, pp. 1-6, 2024, doi: 10.1016/j.dib.2024.110817.

[42] E. Chamseddine, N. Mansouri, M. Soui, and M. Abed, "Handling class imbalance in COVID-19 chest X-ray images classification: Using SMOTE and weighted loss," *Applied Soft Computing*, vol. 129, no. 11, pp. 1-14, 2022, doi: 10.1016/j.asoc.2022.109588.

[43] A. F. A. Alshamrani and F. S. Z. Alshomrani, "Optimizing Breast Cancer Mammogram Classification Through a Dual Approach: A Deep Learning Framework Combining ResNet50, SMOTE, and Fully Connected Layers for Balanced and Imbalanced Data," *IEEE Access*, vol. 13, no. 1, pp. 4815–4826, 2025, doi: 10.1109/ACCESS.2024.3524633.

[44] B. G. Deepa and S. Senthil, "Predicting invasive ductal carcinoma tissues in whole slide images of breast Cancer by using convolutional neural network model and multiple classifiers," *Multimedia Tools and Applications*, vol. 81, no. 3, pp. 8575–8596, 2022, doi: 10.1007/s11042-022-12114-9.

[45] C. Shorten and T. M. Khoshgoftaar, "A survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, no. 7, pp. 1-48, 2019, doi: 10.1186/s40537-019-0197-0.

[46] M. A. Ilani, D. Shi, and Y. M. Banad, "T1-weighted MRI-based brain tumor classification using hybrid deep learning models," *Scientific Reports*, vol. 15, no. 2, pp. 1-16, 2025, doi: 10.1038/s41598-025-92020-w.

[47] G. Imeraj and H. Iyatomi, "Waveshift Augmentation: A Physics-Driven Strategy in Fine-Grained Plant Disease Classification," *IEEE Access*, vol. 13, no. 2, pp. 31303–31317, 2025, doi: 10.1109/ACCESS.2025.3541780.

[48] G. H. Dagnaw, M. El Mouhtadi, and M. Mustapha, "Skin cancer classification using vision transformers and explainable artificial intelligence," *Journal of Medical Artificial Intelligence*, vol. 7, no. 6, pp. 1-17, 2024.

[49] M. Alsaidi, M. T. Jan, and A. Altaher, "Tackling the class imbalanced dermoscopic image classification using data augmentation and GAN," *Multimedia Tools and Applications*, vol. 83, no. 10, pp. 49121–49147, 2024, doi: 10.1007/s11042-023-17067-1.

[50] D. R. Raman, S. Nishanthi, and P. Babysha, "Diagnosis of Diabetic Retinopathy by using EfficientNet-B7 CNN Architecture in Deep Learning," *in Proceedings of the 2023 International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, Coimbatore, India, 2023, vol. 2023, no. 7, pp. 430–435. doi: 10.1109/ICSCSS57650.2023.10169453.

[51] P. Gupta, J. Nirmal, and N. Mehendale, "Custom CNN architectures for skin disease classification: binary and multi-class performance," *Multimedia Tools and Applications*, vol. 2024, no. 12, pp. 1-28, 2024, doi: 10.1007/s11042-024-20503-5.

[52] A. Alzahrani, "Digital Image Forensics: An Improved DenseNet Architecture for Forged Image Detection," *Engineering, Technology & Applied Science Research (ETASR)*, vol. 14, no. 2, pp. 13671–13680, 2024, doi: 10.48084/etasr.7029.

[53] J. Zhou et al., "Intelligent classification of maize straw types from UAV remote sensing images using DenseNet201 deep transfer learning algorithm," *Ecological Indicators*, vol. 166, no. 9, pp. 1-15, 2024, doi: 10.1016/j.ecolind.2024.0788X.

[54] C. González-Santoyo, D. Renza, and E. Moya-Albor, "Identifying and Mitigating Label Noise in Deep Learning for Image Classification," *Technologies*, vol. 13, no. 4, pp. 1-17, 2025, doi: 10.3390/technologies13040132.

[55] S. S. Alam, A. Ahad, S. Ahmed, J. Dudley, and T. H. Farook, "Using deep learning to segment impacted molar teeth from panoramic radiographs," *Digital Dentistry Journal*, vol. 1, no. 3, pp. 1-30, 2025, doi: 10.1016/j.ddj.2025.100007.

[56] A. Kuppusamy, S. K. Sundaresan, and R. Cingaram, "Enhancing sugarcane leaf disease classification through a novel hybrid shifted-vision transformer approach: technical insights and methodological advancements," *Environmental Monitoring and Assessment*, vol. 197, no. 37, pp. 1-20, 2025, doi: 10.1007/s10661-024-13468-3.

[57] B. Zheng, J. Liu, Y. Li, Y. Li, and Z. Qin, "MAS-PD: Transferable Adversarial Attack against Vision-Transformers-Based SAR Image Classification Task," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 18, no. 2, pp. 9509-9521, 2025, doi: 10.1109/JSTARS.2025.3546271.

[58] Q. Wu et al., "A classification method for soybean leaf diseases based on an improved ConvNeXt model," *Scientific Reports*, vol. 13, no. 11, pp. 1-11, 2023, doi: 10.1038/s41598-023-46492-3.

[59] D. Reyes and J. Sánchez, "Performance of convolutional neural networks for the classification of brain tumors using magnetic resonance imaging," *Heliyon*, vol. 10, no. 3, pp. 1-22, 2024, doi: 10.1016/j.heliyon.2024.e25468.