Current **A**pplied **S**cience and **T**echnology e-ISSN: 2586-9396

Curr. Appl. Sci. Technol. ….., Vol... (No...), e0264760

Research article

A Comparison of Deep Neural Network for Hevea Clone Identification

Thiraphat Romruensukharom and Sarayut Nonsiri*

Artificial Intelligence and Internet of Things Research Laboratory (AIoT),
Faculty of Information Technology, Thai-Nichi Institute of Technology, Bangkok, Thailand

Received: 19 September 2024, Revised: 9 April 2025 , Accepted: 10 June 2025, Published: 10 October 2025

Abstract

Hevea brasiliensis Muell. Arg, a rubber tree, is a highly heterozygous perennial plant usually grown from seed (seedlings). The tree exposed the disadvantage of no genetic uniformity. Unlike a clone, it was propagated by bud grafting from a single tree, possessing an identical genetic constitution and exhibiting uniformity among them. The leaf shape of seedlings is highly variable, while the leaf shape of clones is slightly variable. It also appears in similar characteristics to other clones. Therefore, the variation of leaf shape becomes the critical concern to distinguish them. The common cultivation clone RRIM 600 was considered for experiments, the dataset of RRIM 600 clones and seedlings was used for training the model. The objective of the research was to compare the performance of deep neural networks for *H. brasiliensis* clone identification, including VGG16, ResNet50, InceptionV3, MobileNet, Xception, DenseNet201, NASNetLarge, MobileNetV2, EfficientNetB7, RegNetX064, RegNetY064, ResNetRS50 and ConvNeXtBase. The appropriate hyperparameters were found through k-fold cross validation. The models were trained using transfer learning technique with FEA. Various augmentation techniques were applied in order to improve the performance. The results revealed that improved retraining the model on low resolution images by implementing ConvNeXtBase as feature extractor with S1 achieved the highest accuracy of 97.82% on a quarter of dataset (E3) and outperformed classification performance across all thresholds. This research suggests the potential for developing this Hevea clone identification application as a tool to overcome the lack of experienced Hevea clone inspectors.

Keywords: *Hevea brasiliensis;* Hevea clone identification; precision agriculture; image classification; deep neural network

*Corresponding author: E-mail: sarayut.n@tni.ac.th
https://doi.org/10.55003/cast.2025.264760

1. Introduction

Hevea brasiliensis Muell. Arg is one of the important economic trees. It produces natural rubber, which is an essential raw material broadly used in many products such as aircraft and car tires, automotive parts, medical devices, surgical gloves, condoms, baby pacifiers, shoes, elastic and adhesives, toys, etc. Thailand has been the world's largest producer and exporter of natural rubber (Arias & van Dijk, 2019). In 2024, Thailand produced 4.79 million tons of natural rubber (Office of Agricultural Economics, 2024a), 3.91 million tons for export (valued at 270,706 million baht), and 1.25 million tons for domestic consumption. Currently, the cultivated area has grown to exceed 3.81 million hectares throughout the country (Office of Agricultural Economics, 2024b). Traditionally, rubber was grown only in the southern and eastern regions of Thailand where environmental conditions were favorable. However, the rubber plantations rapidly expanded to the northern and northeastern regions between 2004 and 2006 because the government supported and promoted rubber as a new cash crop that would increase income and stabilize the economy of small landholders (Poungchompu & Chantanop, 2015).

The rubber tree is a highly heterozygous perennial plant with a harvest lifespan of more than 20 years. The yield is largely dependent on the rubber clone planted and the agro-management. Rubber clones developed through genetic improvement with consistent yield potential and cultivation adaptability are recommended for commercial cultivation. Clone inspection can help cultivators to ensure that the right clones are used for planting because it can guarantee that recommended rubber clones produce the maximum yield in the future. Commonly, rubber clones are identified by recognizing its specific organs such as leaf storey, leaf, petiole, bark, axillary bud, or combinations thereof (Liyanage, 2021). These visual observations of the morphological parts of the trees are likely to be influenced by personal skill and environmental conditions (Saraswathyamma, 2000). Accurate clone identification requires experts with adequate experience. In fact, molecular markers are more reliable for clone identification than morphology, but they are often time and cost consuming.

The right clone of rubber planting materials has been inspected on immature stage; some feature cannot be considered in the identification process. The propagation of *H. brasiliensis* is implemented by seed and bud graft. Trees raised from seeds are referred to as seedling trees, which are not genetically uniform. Since rubber is a cross-pollinated plant and its genetics are heterozygous, each seedling tree has a distinct genotype. On the contrary, a population of budded trees from a single tree is known as a "clone". All such trees possess identical genetic constitutions and existing uniformity among them. The leaf shapes of seedlings are highly variable, while the leaf shapes of clones are slightly variable. In addition, advances in genetic improvement produce new clones, but the genetic base of the existing population is very narrow. Therefore, it appears in similar characteristics to former clones. Variation of leaf shape is one of the critical concerns to distinguish them. Since the limitation of morphological characteristics makes accurate identification more difficult, Pratomo et al. (2021) and Pasaribu et al. (2022) suggested that the characteristics present in *H. brasiliensis* leaves were suitable for consideration as a useful variable in clone classification. Thus, leaves are the only suitable morphological features considered for identification.

Recently, the characteristics of *H. brasiliensis* leaves were studied using traditional feature extraction techniques (Anjomshoae et al., 2015; Anjomshoae & Rahim, 2018) and machine learning algorithms such as Logistic Regression, Naïve Bayes, K-Nearest Neighbors (KNN), Random Forest and Artificial Neural Network (ANN) (Thurachon &

Sumethawatthanaphong, 2014; Pongsomsong & Ratanaworabhan, 2021; Yaiprasert, 2021). These were utilized in attempts to overcome the challenges in classifying the complicated traits. An impressive type of ANN is Convolutional Neural Network (CNN) which is specifically designed for image recognition (Zeiler & Fergus, 2014; O'Shea & Nash, 2015). The algorithm can find the best patterns through feature extraction. This process can be done automatically through CNN with less preprocessing and provided better classification performance than using hand-crafted features (Huang et al., 2019; Tiwari, 2020; Kanda et al., 2021). Furthermore, CNN also achieved satisfactory performance on various tasks related to *H. brasiliensis* leaves (Hassan et al., 2022; Zeng et al., 2022; Kaewboonna et al., 2023; Balaga & Patayon, 2024;) and other plant classification (Chang & Lai, 2024; Li et al., 2024; Ngugi et al., 2024; Nibret et al., 2025).

The challenges in computer vision that required highly experienced specialist led to the development of a more complex architecture (Bengio et al., 2021) called "Deep Neural Network". This architecture was built up based on ANN and CNN with the ability to classify new unseen data more accurately. The advantages of this technology may result in a more efficient *H. brasiliensis* classification process. Therefore, the objective of this research was to compare the performance of a deep neural network that was suitable for *H. brasiliensis* clone identification from leaf characteristics.

2. Materials and Methods

This research was conducted using a deep learning approach. The experiment was performed based on related theory and research.

2.1 Artificial neural network

Artificial neural network (ANN) is a computational model inspired by biological neural network in the human brain (Rosenblatt, 1958; Kelleher, 2019). It consists of several interconnected artificial neurons. Data from an input layer is processed through hidden layers. In an artificial neuron, weights, biases and activation functions are applied. The data is fed forward to the next hidden layer and flows to the output layer for decision making. This procedure allows the network to learn complex relationships between data.

2.2 Convolutional neural network

Convolutional neural network (CNN), a scalable approach that is an extension of ANN, was specifically designed for image recognition and was inspired by human visual systems (Fukushima, 1980; LeCun & Bengio, 1995). It leverages mathematical principles and contains a set of convolutional and pooling layers. In the CNN, multiple filters are applied to recognize specific pattern and allow the computer to derive meaningful information from the images, called convolutional operations. The feature extraction process in convolutional layer (Dumoulin & Visin, 2018), in which useful features of the image are extracted by sliding filters over the image and feeding to a convolutional function. This process is performed by matrix multiplication and storing them as a multidimensional array (width, depth and height) corresponding to RGB in an image. The product between each element (pixels) of the overlapped input element is computed and the results are summed up to obtain the output in the current location. The procedure can be repeated using various filters to generate multiple feature maps as desired. The produced feature maps consist of different type of features such as curves, edges, texture and other patterns (Figure 1).

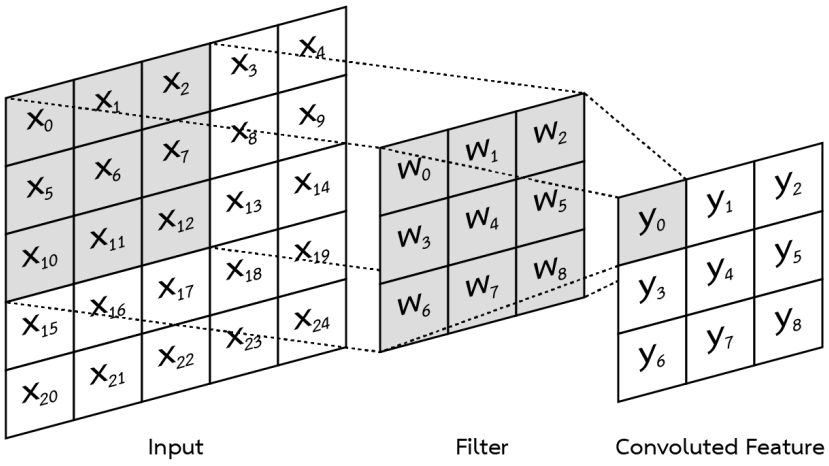

Figure 1. Convolutional operation

One of the key success in convolution is the simple process of removing pixels while maintaining important information (Dumoulin & Visin, 2018), called pooling operations. The feature maps obtained from convolutional operation are pooled with max pooling, in which the highest values in each filter are pooled, or average pooling, in which the average of all values in the filter is calculated and pooled to the new reduced feature maps (Figure 2). These operations enables the models to focus on high-level features more efficiently.

(a) (b)

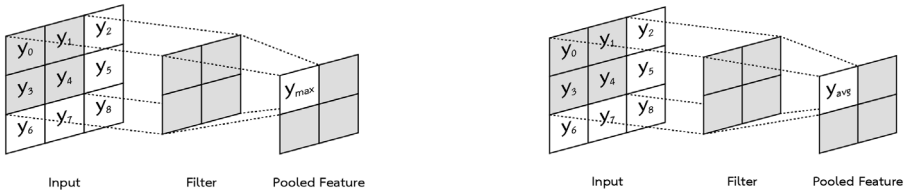

Figure 2. Max pooling (a) and Average pooling (b)

2.3 Pre-trained model

Pre-trained model is deep neural network (DNN) available on Keras (Chollet, 2015) with weight initialization that represents features of thousand different classes traind on ImageNet (Deng et al., 2009) dataset. These models were designed and developed in different architecture with high potential of feature extraction in multiple level of abstraction (LeCun et al., 2015). Their use as feature extractors can address the lack of data in DNN and reduce time consuming on training phase. Moreover, this pre-trained model also

improves robustness and uncertainty (Hendrycks et al., 2019). The success on ImageNet challenge in advancing computer vision and deep learning research can be achieved by high performance on image classification and object detection (Krizhevsky et al., 2012). It is established that DNN has become one of the mainstream developments in the field of Artificial Intelligence (AI). The models appear in various families as follows:

2.3.1 VGG

VGG (Visual Geometry Group) is an early generation of DNN architecture proposed by Simonyan and Zisserman (2014), which was based on the simple concept of enhancing the performance of feature extraction by increasing the depth of CNN in sequential form with very small convolutional filters to 16-19 weight layers. The architecture is beneficial for more complex classification task and achived state-of-the-art performance.

2.3.2 ResNet

The development of deeper sequential neural network exposed a degradation problem called “vanishing gradients” that leads to higher training error. ResNet (Residual Network) was introduced by He et al. (2016) to address this problem by introducing shortcut connections that can skip a few layers. The connections perform identity mapping referenced to the input layer followed by residual function. Preserved useful information flows directly from the initial layer to deeper layers resulting in easier weight updating for network training. This concept enabled CNN to be of greater depth and become the foundation of new architectures. Moreover, the model was developed into a new version (Xie et al., 2017). The training strategy and architecture ratio were improved (Bello et al., 2021). Some ResNet architectures proved superior to state-of-the-art architectures.

2.3.3 Inception

A complex DNN called “GoogLeNet” built up from Inception module was presented by Szegedy et al. (2015) to improved utilization of the computing resources. The module was inspired by the ability of the biological visual cortex to identify patterns at different scales. The Inception network was heavily constructed based on the network in network approach (Lin et al., 2014). The network consists of different sub-modules that are multi-scale feature extractors that work in parallel and outperform conventional deep neural network. According to complex architecture, this module was improved in later version (Szegedy et al., 2016; Szegedy et al., 2017) with higher efficiency.

2.3.4 MobileNet

The light weight DNN called MobileNet was presented by Howard *et al.* (2017). This network was designed and developed under limitations of memory and latency. It was built from depthwise separable convolutions. In the module, depthwise convolutions were performed by applying a single filter to each input channel followed by pointwise convolution to combine the outputs. This concept drastically affected reduced computational power and model size. Although MobileNet did not achieve new record accuracy, the model was smaller and faster than before and was highly efficient for mobile and embedded vision applications. The model in this family was also improved by applying the concept of residual function (Sandler et al., 2018) and network architecture search

(Howard et al., 2019) to the next generation achieved state-of-the-art performance on mobiles.

2.3.5 Xception

Xception is an extreme version of Inception proposed by Chollet (2017). The standard convolutional layers (Inception module) were replaced by modified depthwise separable convolutions with residual connections. The module performs cross - channel correlation mapping (pointwise convolution) then separates one spatial correlation for every output channel (depthwise convolution). The network outperformed the new version of Inception on large datasets.

2.3.6 DenseNet

The concept of shorter connections results in deeper CNN with higher efficiency training. DenseNet (Dense Convolutional Network) was introduced by Huang et al. (2017) with dense connectivity in a block of network. The connection performs reusable feature by connecting each layer to every other layers for improving information flow. Each layer receives feature maps of all preceding layers for concatenation. This architecture not only acheives high performance but also improves the efficiency of the network.

2.3.7 NASNet

A new technique for automating design of CNN architecture without human expertise, called Neural Architecture Search (NAS) was presented by Zoph et al. (2018). The proper architecture was discovered in search space with reinforcement learning on a dataset of interest and consisted of two types of repeated blocks or cells. The first type is convolutional cells that return a feature map of the same dimension (Normal Cell) and the second type (Reduction Cell) return a feature map where the dimension is reduced. This technique provided scalable architecture that required less computation than human designed and achieved accurate result.

2.3.8 EfficientNet

An efficient technique for scaling up CNN was introduced by Tan and Le (2019) based on observation of deep neural network development. This is commonly acheived under limited resource then scaled up for better performance. This network balancing technique uniformly scales up the width, depth and resolution at all dimensions at constant ratio, initializing from a new baseline model designed by NAS, resulting in various size of models in the family. A CNN can be scaled up effectively and set up new standard of CNN design.

2.3.9 RegNet

A new network design paradigm was presented by Radosavovic et al. (2020) to discover efficient design principles across all setting. It combining the adventages of manual design and NAS. The simple and fast DNN call RegNet was achieved through (AnyNet) network design spaces. This network outperformed some state-of-the-art DNN, and especially overcoming EfficientNet in term of training speed on GPU.

2.3.10 ConvNeXt

A new architecture ConvNeXt, inspired by the design of vision transformers (ViTs) (Dosovitskiy et al., 2021), was proposed by Liu et al. (2022) to improve the performance of CNN. The network modernized the concept of ResNet and resulted in architecture and training procedure being redesigned to be closer to ViTs. In the design process, the stage compute ratio was adjusted. Input preprocessing was changed from standard convolution (Stem) to non-overlapping convolution (Patch) (Liu et al., 2021), the idea of new version of ResNet was adopted to ResNeXtify (a deep convolutional network) which is the operation similar to self-attention (Cordonnier et al., 2020) in ViTs. Inverted bottle neck was created in a block of depthwise with larger filter and utilized state-of-the-art parameter in each layer. With these improvements, ConvNeXt surpassed ViTs and outperformed state-of-the-art CNN.

Even though these DNNs have evolved into more advanced networks, they are not suitable for all specific tasks. Therefore, the experiment with target task is required. This research employed thirteen pre-trained models from aforementioned families to compare the performance of the models in order to discover the optimal architecture. Included were VGG16, ResNet50, InceptionV3, MobileNet, Xception, DenseNet201, NASNetLarge, MobileNetV2, EfficientNetB7, RegNetX064, RegNetY064, ResNetRS50 and ConvNeXtBase.

2.4 Data acquisition

Fresh *Hevea* leaves were collected from the budwood and rootstock nursery in Chachoengsao Rubber Regulartory Center. Healthy middle leaflets were taken from the upper mature whorls and images were captured using a digital camera (Nikon COOLPIX B700 and Fujifilm FINEPIX F300EXR). There were 36 images per leaf taken from various viewpoints and orientations on a white background under controlled environment (Figure 3). The balanced dataset contains 28,800 images of 2 classes: the RRIM 600 Clone and the RRIM 600 seedling. The RRIM 600 clone contains images of plants propagated by budding while the RRIM 600 seedling contains images of plants grown from seeds. Clones, unlike seedlings, do not exhibit highly distinct variations. Most of them possess more stable morphological features, which can be used for identification.

Figure 3. Sample images from leaf rotation and camera angle adjustment

2.5 Data understanding

The shape of the leaves is one of the characteristics commonly used to classify *Hevea* clones. The middle leaflet of mature leaves is the ideal leaf for characterization. They often appear in three basic forms (Table 1): elliptical, obovate and diamond, as shown in Figure 4.

Table 1. Leaf shape characteristics of *H. brasiliensis*

Forms	Characteristics
Elliptical	The maximum width is on the middle and tapers equally towards base and apex.
Obovate	The maximum width is found between the middle of leaflet and apex.
Diamond	Both sides of the leaf blade that divides along the midrib are not symmetrical. The widest part of the leaf on both sides is not in the same position.

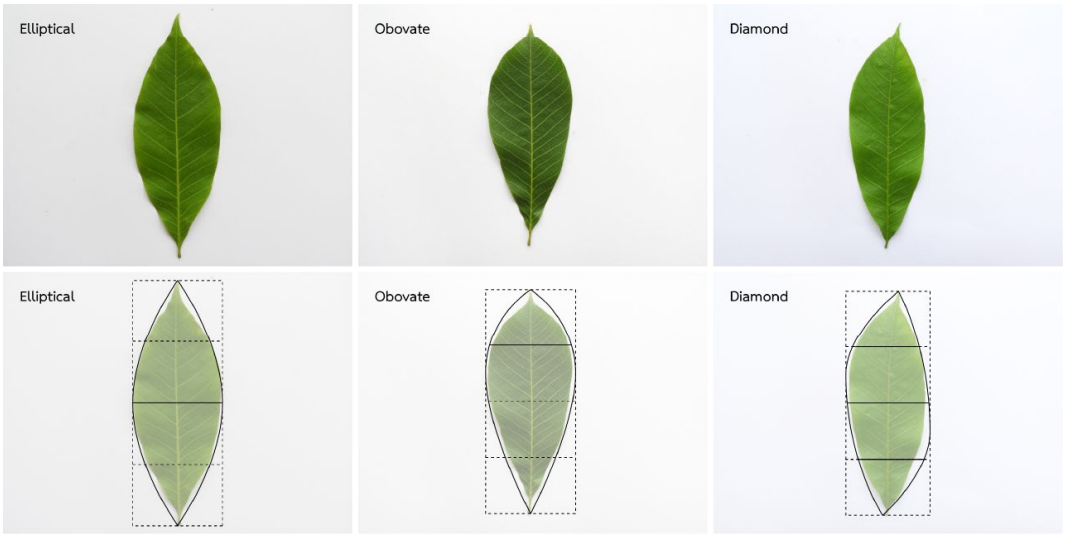

Figure 4. Leaf shapes of *H. brasiliensis*

The middle leaflet of RRIM 600 clone was considered as obovate form. However, the heterozygous nature of the rubber tree affects the leaf shapes of RRIM 600 seedlings, which appear in various forms. Therefore, leaf shape of RRIM 600 seedlings is highly variable compared to RRIM 600 clone as shown in Figure 5.

2.6 Data preparation

Data quality and quantity directly affect the performance of a DNN. During the training process, DNN learns from relationship between features of the dataset. The reliability of model performance often relies on the quality of a given data. Therefore, data preparation becomes a crucial process to make the model more robust. The data preparation process can be describe as follows:

Figure 5. Leaf shape variation of RRIM 600 seedlings

2.6.1 Image resizing

Training with high-resolution images requires high computational power and is time consuming. It was also difficult to find a suitable architecture for *H. brasiliensis*. For faster training, the dimension of images was carefully resized while maintaining their image aspect ratio, as shown in Table 2.

2.6.2 Image augmentation

DNN requires a vast amount of comprehensive data. The number of images in data sets was increased using two data augmentation techniques (Lei et al., 2019; Fonseka & Chrysoulas, 2020) as detailed below.

1) Geometric augmentation: In this research, it consists of two methods:

- Traditional method: This is a simple and high efficiency method that involves manual leaf rotation and flpping when taking a photo (Figures 6 and 7) (Zheng et al., 2016).
- Specific method: In *H. brasiliensis* clone identification, the leaves are captured from various viewpoints. The camera angle is adjusted around 60 degrees from left to right per every 45 degrees of leaf rotation (Figures 8 and 9).

Table 2. Images resizing

Digital Camera	Image Resizing			
	Orginal		Resized	
	Width	Height	Width	Height
Nikon COOLPIX B700	5184 pixel	3888 pixel	324 pixel	243 pixel
Fujifilm FINEPIX F300EXR	2816 pixel	2112 pixel	352 pixel	264 pixel

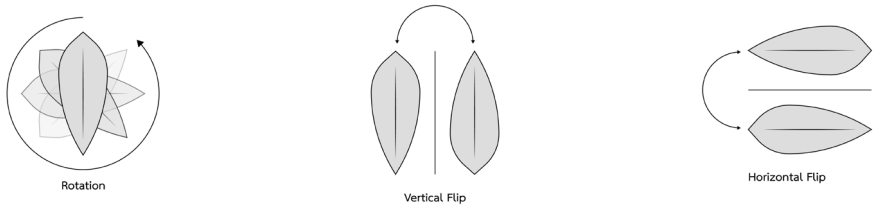

Figure 6. Geometric augmentation (Traditional method)

Figure 7. Sample images from geometric augmentation (Traditional method)

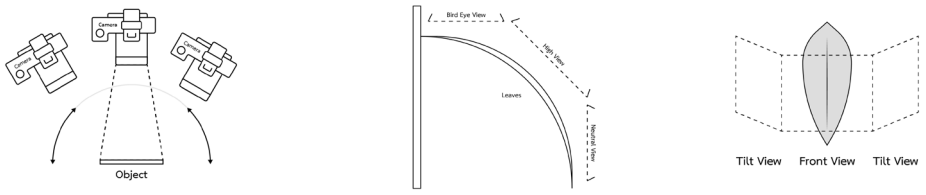

Figure 8. Geometric augmentation (Specific method)

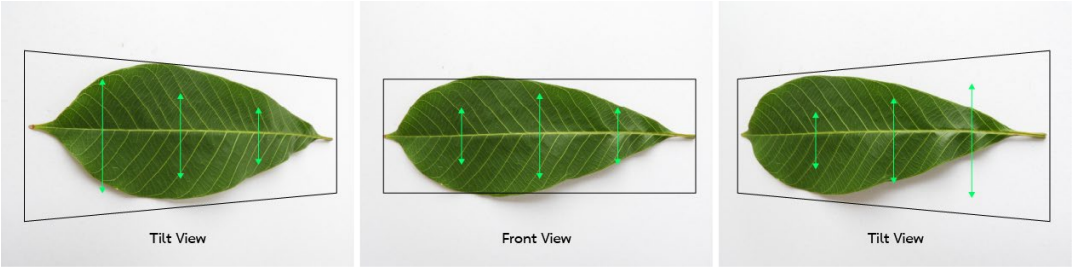

Figure 9. Sample images from geometric augmentation (Specific method)

2) Photometric augmentation: This technique is artificially generated on a computer. Guassian noise and radial distortion (Figure 10) are used to improve model performance and prevent distortion effects from wide-angle lenses (Buquet et al., 2021), illumination and temperature (Zhou et al., 2017) to accommodate environmental changes (Dodge & Karam, 2016; Xiao et al., 2020) in real situations.

2.7 Modeling

The models were trained by using transfer learning from pre-trained models due to the different resource constraints of each architecture (Bianco et al., 2018). The technique can address the data shortage problem in DNN and reduce time consuming on training phase. The concept of transfer learning is a training process by initializing weights from a pre-trained model (Yosinski et al., 2014; Yin et al., 2017). There are two approaches:

1) Feature extraction approach (FEA): The pre-trained models were used as a feature extractor. Related generic features were transferred and the specific features were extract to train the custom classifier. This approach requires a similar domain problem and a large amount of data.

2) Fine-tuning approach (FTA): The custom classifier was trained for introducing to the specific task, then a few top layers of a frozen (convolutional layers) in the pre-trained model were unfrozen and fine-tuned the higher order feature representation. This approach takes more time than FEA.

This research explores suitable architecture using transfer learning techniques based on experimental study. Hyperparameter tuning and the sufficient amount of data were the key factors. Therefore, FEA was implemented for more efficient.

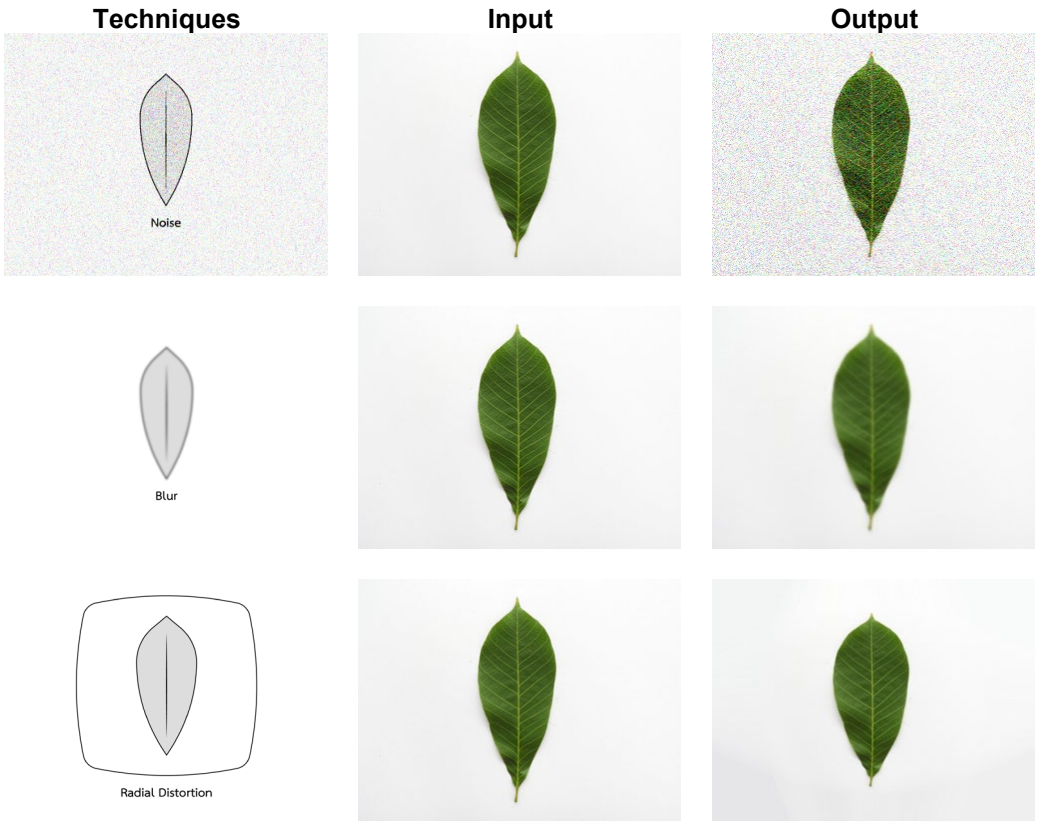

Figure 10. Sample images from photometric augmentation

2.8 Evaluation metrics

A confusion matrix is a table displaying the number of predictions of a model and it was utilized for the performance evaluations. Table 3 describes the performance of classifiers in four terms:

- True positive (TP): the model correctly predicted the positive class. RRIM 600 clones were identified.
- True negative (TN): the model correctly predicted the negative class. RRIM 600 seedlings were identified.
- False positive (FP): the model incorrectly predicted the negative class. RRIM 600 seedlings were misidentified.
- False negative (FN): the model incorrectly predicted the negative class. RRIM 600 clones were misidentified.

Table 3. Confusion matrix

	P' (Predicted)	**N'(Predicted)**
P (Actual)	True Positive	False Negative
N (Actual)	False Positive	True Negative

The evaluation metrics used were accuracy, precision, recall, and F1-score. The calculation of these evaluation metrics according to the values in the confusion matrix was made by equations (1)-(4).

$$Accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + True\ Negative + False\ Positive + False\ Negative} \quad (1)$$

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive} \quad (2)$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (3)$$

$$F1 - Score = \frac{2 * True\ Positive}{2 * True\ Positive + False\ Positive + False\ Negative} \quad (4)$$

A receiver operating characteristics (ROC) curve was computed by plotting the true positive rate (TPR) and false positive rate (FPR). The area under the ROC curve (AUC) was constructed from this plot and denoted as the ability of the model to seperate classes at different thresholds ranging from 0 to 1.

$$TPR(Sensitivity) = \frac{True\ Positive}{True\ Positive + False\ Negative} \quad (5)$$

$$FPR(1 - Specificity) = \frac{False\ Positive}{True\ Negative + False\ Positive} \quad (6)$$

$$AUC = \frac{1}{2}\left(\left(\frac{True\ Positive}{True\ Positive + False\ Negative}\right)\right) + \left(\left(\frac{True\ Negative}{True\ Negative + False\ Positive}\right)\right) \quad (7)$$

2.9 Experimental setup

The experiments were conducted on Intel Xeon E3-1270 v5 3.6 GHz desktop with 16 GB RAM programmed in python 3.11.5 and modeling based on Keras and TensorFlow 2.15.

2.9.1 Dataset splitting

The dataset was devided into training and testing sets in ratio of 85 : 15 for three training experiments : full dataset (800 leaves), half of the dataset (400 leaves), and one-quarter of the dataset (200 leaves), respectively. The quantity of data and optimal size of the dataset are shown in Table 4 and Table 5.

Table 4. Data splitting of main dataset

Main Dataset (Original + Geometric Augmented))			
Experiment	**Class**	**Training Set**	**Testing Set**
Full (E1)	RRIM 600 Clone	12,240	2,160
	RRIM 600 Seedling	12,240	2,160
Half (E2)	RRIM 600 Clone	6,120	1,080
	RRIM 600 Seedling	6,120	1,080
Quarter (E3)	RRIM 600 Clone	3,060	540
	RRIM 600 Seedling	3,060	540

Table 5. Data splitting for K-fold cross validation experiment

K-Fold Cross Validation Dataset (Original + Geometric Augmented)			
Experiment	**Class**	**Training Set**	**Testing Set**
Full (E1)	RRIM 600 Clone	10,080	2,160
	RRIM 600 Seedling	10,080	2,160
Half (E2)	RRIM 600 Clone	5,040	1,080
	RRIM 600 Seedling	5,040	1,080
Quarter (E3)	RRIM 600 Clone	2,520	540
	RRIM 600 Seedling	2,520	540

2.9.2 Architecture design

The architecture of our models (Table 6) were designed as follows:

1) Feature extractor: Thirteen selected pre-trained models were used as feature extractors.

2) Classifier: The custom classifer (Fully-connected layer) was added for *H. brasiliensis* clone identification. Initially, the input data was transformed into one dimensional vector via Flatten. For the hidden layer, Dense was added with a standard activation function named ReLU to achieve lower traning loss (Ding et al., 2018; Javid et al., 2021) followed by BatchNormalization. Each training batch was normalized for speedup in training (Ioffe & Szegedy, 2015). Some units in the networks were randomly dropped by dropout with probability 0.5 to prevent overfitting (Srivastava et al., 2014). After that, dense output was added for the prediction according to the number of classes. Sigmoid was applied to normalize the range of output into a probability value between 0-1 and the threshold was set to 0.5 for binary classification.

Table 6. Model architecture and hyperparameter setup

Hyperparameters	Setting					
	S1	**S2**	**S3**	**S4**	**S5**	**S6**
Input Shape	64 x 64 71 x 71 (Inception) 75 x 75 (Xception)					
Batch Size	32					
Optimizer	Adam		Nadam		RMSProp	
Learning Rate	0.0001					
Patience (Early Stopping)	10					

Model Architecture							
Layers	**Type**	**Structure**					
	Pre-trained	VGG16, ResNet50, InceptionV3, MobileNet, Xception, DenseNet201, NASNetLarge, MobileNetV2, EfficientNetB7, RegNetX064, RegNetY064, ResNetRS50, ConvNeXtBase					
Classifier	Flatten						
	Dense	512	1024	512	1024	512	1024
		256	512	256	512	256	512
			256		256		256
	ReLU						
	BatchNormalization						
	Dropout (0.5)						
	Dense	1					
		Sigmoid					

2.9.3 K-fold cross validation

K-fold cross validation was applied to tune the models (Yadav & Shukla, 2016) during training. The training set was equally splitted into ten folds. In each iteration, one of the ten folds was held out for testing and the remaining folds were used for training. This process was repeated until every fold had been used as testing set. After that, the results of all iterations were averaged to estimate the model's performance.

2.9.4 Hyperparameters

Finally, all images were scaled to the default minimum size of each model. The model weights were updated after feeding each batch of 32 samples for training using Adam, Nadam and RMSProp optimizer (Kingma & Ba, 2017; Dogo et al., 2022) with an effective initial learning rate of 10^{-4} (Jepkoech et al., 2021). The patience of EarlyStopping was set to ten epochs for loss monitoring. Training was forced to stop when a model's performance was not significantly improved for ten epochs.

3. Results and Discussion

3.1 Experimental results

The models were trained with three different sizes of the dataset through k-fold cross validation to find the appropriate hyperparameter and dataset. The results showed that the models trained with S2 on full dataset (E1) achieved highest average accuracy of 96.12% with 126h 21m 35s of training time while the models trained with S1 on half of dataset (E2) and a quarter of dataset (E3) achieved the highest average accuracies of 93.76% with 32h 42m 40s of training time and 95.06% with 15h 14m 24s of training time, respectively (Table 7).

Table 7. The average performance of k-fold cross validation experiments

K-Fold Cross Validation Experiments							
Dataset	**Metrics**	**Setting**					
		S1	**S2**	**S3**	**S4**	**S5**	**S6**
E1	Avg. Acc	0.9565	**0.9612**	0.9512	0.9499	0.9464	0.9482
	Avg. Loss	0.3164	**0.2633**	0.2973	0.4006	0.3593	0.5332
	Avg. TT (hh:mm:ss)	65:15:09	**126:21:35**	82:32:31	117:47:39	49:41:42	99:01:12
E2	Avg. Acc	**0.9376**	0.9370	0.9337	0.9292	0.9313	0.9280
	Avg. Loss	**0.3822**	0.5623	0.6856	0.5298	0.7145	0.6981
	Avg. TT (hh:mm:ss)	**32:42:40**	63:14:12	39:25:20	83:18:13	23:13:32	49:38:21
E3	Avg. Acc	**0.9506**	0.9471	0.9518	0.9439	0.9415	0.9387
	Avg. Loss	**0.3457**	0.3509	0.2619	0.6609	0.5272	0.8778
	Avg. TT (hh:mm:ss)	**15:14:24**	30:36:48	19:51:04	40:32:32	11:43:37	23:49:10

Abbreviation: Avg. (Average), Acc (Accuracy), TT (Training Time)

Considering the trade-off between accuracy and training time, it seems training with S1 on E2 and E3 datasets was an effective approach for retraining with train-test splitting because it consumed less training time and smaller dataset size. Thus, in the retraining phase, the models were trained with hyperparameters settings based on these findings.

Retraining the models with S1 on E2 dataset, it was found that MobileNetV2 achieved the highest accuracy of 89.12% with 21m 04s of training time followed by ConvNeXtBase and DenseNet201 with accuracies of 87.96% and 87.82% with 55m 22s and 21m 46s of training time, respectively. NASNetLarge gave the lowest accuracy of 80.05% with 57m 22s of training time (Table 8). However, retraining the models with S1 on the E3 dataset revealed that ConvNeXtBase achieved the highest accuracy of 98.61% at

27m 57s of training time followed by RegNetY064 and ResNetRS50 at accuracies of 95.83% and 94.91%, respectively, at the same 6m 54s of training time. EfficientNetB7 gave the lowest accuracy of 60.74% at 17m 41s of training time (Table 9).

Interestingly, the model that built up based on a simple DNN, VGG16, achieved satisfactory performance that was the same as a complex DNN. Moreover, it performed better than some state-of-the-art architecture with only a few minutes of training time.

With sufficient amount of data, ConvNeXtBase achieved the highest accuracy and performed significantly better overall. NASNetLarge consumed the longest training time but it was not the best model in any experiments, while MobileNet and MobileNetV2 achieved satisfactory performance under shorter training times compared to larger models.

Most of the trained models were able to learn this task but displayed varying performance due to their architectures. In each experiment, most models performed slightly lower than the best model. On the other hand, NASNetLarge and EfficientNetB7 were relatively poor models, showing significant accuracy drop in some experiments, and it seemed that NASNetLarge and EfficientNetB7 could not generalize well on this dataset.

The ROC curve (Figure 11) of retrained models with S1 on E2 dataset (a) showed that ConvNeXtBase achieved the highest AUROC at 0.9762 followed by VGG16 and MobileNetV2 at 0.9726 and 0.9704, respectively. The lowest curve was NASNetLarge at 0.8815.

For the retrained model with S1 on E3 dataset (b), ConvNeXtBase outperformed classification performance with AUROC at 0.9993, resulting ROC curve approaches the ideal top-left corner demonstrating the model's effectiveness, followed by ResNet50 and RegNetY064 at 0.9927 and 0.9923, respectively. The lowest curve was NASNetLarge at 0.8992.

Table 8. The performance of the retrained models with half dataset (E2)

Retrained : Half (E2)						
Model	**Accuracy**	**Loss**	**Precision**	**Recall**	**F1 - Score**	**Training Time (hh:mm:ss)**
VGG16	0.8648	0.8286	0.8862	0.8648	0.8754	00:08:29
ResNet50	0.8745	0.7093	0.8885	0.8745	0.8814	00:22:02
InceptionV3	0.8495	0.8782	0.8596	0.8495	0.8545	00:06:07
MobileNet	0.8745	0.7804	0.8949	0.8745	0.8846	00:13:06
Xception	0.8287	0.9201	0.8626	0.8287	0.8453	00:33:55
DenseNet201	0.8782	0.6825	0.8844	0.8782	0.8813	00:21:46
NASNetLarge	0.8005	0.7999	0.8031	0.8005	0.8018	00:57:22
MobileNetV2	**0.8912**	**0.7173**	**0.8963**	**0.8912**	**0.8937**	**00:21:04**
EfficientNetB7	0.8333	0.6107	0.8641	0.8333	0.8484	00:38:52
RegNetX064	0.8287	0.9860	0.8570	0.8287	0.8426	00:18:35
RegNetY064	0.8333	0.8914	0.8457	0.8333	0.8395	00:16:10
ResNetRS50	0.8708	0.8478	0.8924	0.8708	0.8815	00:24:01
ConvNeXtBase	0.8796	0.6632	0.8934	0.8796	0.8864	00:55:22

Table 9. The performance of the retrained models with quarter dataset (E3)

Retrained : Quarter (E3)						
Model	**Accuracy**	**Loss**	**Precision**	**Recall**	**F1 - Score**	**Training Time (hh:mm:ss)**
VGG16	0.9361	0.2271	0.9365	0.9361	0.9363	00:03:18
ResNet50	0.9398	0.2486	0.9438	0.9398	0.9418	00:13:14
InceptionV3	0.9037	0.4736	0.9037	0.9037	0.9037	00:03:30
MobileNet	0.9315	0.2330	0.9322	0.9315	0.9319	00:03:14
Xception	0.9213	0.2901	0.9227	0.9213	0.9220	00:13:32
DenseNet201	0.9296	0.3406	0.9298	0.9296	0.9297	00:10:05
NASNetLarge	0.8231	0.6686	0.8276	0.8231	0.8253	00:30:02
MobileNetV2	0.9204	0.3000	0.9206	0.9204	0.9205	00:04:53
EfficientNetB7	0.6074	2.6170	0.7801	0.6074	0.6830	00:17:41
RegNetX064	0.8685	0.4488	0.8919	0.8685	0.8800	00:08:25
RegNetY064	0.9583	0.1863	0.9588	0.9583	0.9586	00:06:54
ResNetRS50	0.9491	0.1869	0.9491	0.9491	0.9491	00:06:54
ConvNeXtBase	**0.9861**	**0.0416**	**0.9862**	**0.9861**	**0.9862**	**00:27:57**

(a) **(b)**

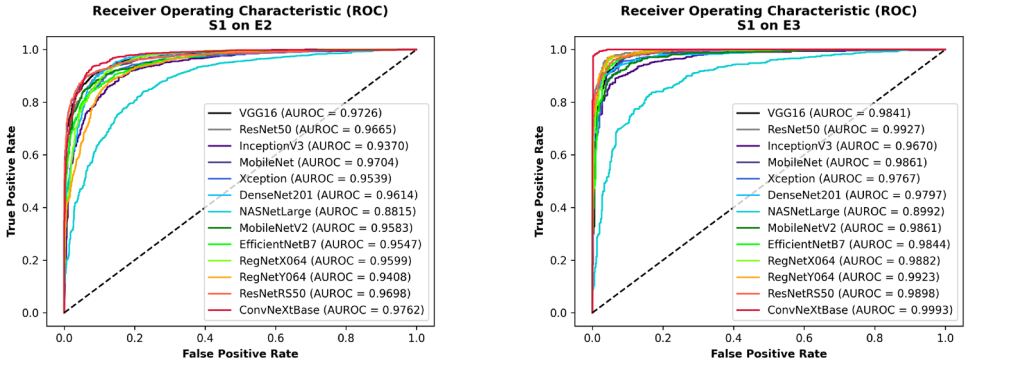

Figure 11. ROC Curve of S1 on E2 (a) and E3 (b) datasets

The model built up by implementing ConvNeXtBase as feature extractor needed longer training time than the other models except for NASNetLarge. Although ConvNeXtBase did not achieve the highest accuracy on the E2 dataset, the model outperformed others in identifying Hevea clones at different thresholds. For this reason, ConvNeXtBase was considered to improve the performance by photometric augmentation. Noise and radial distortion were applied to the dataset, so the quantity of data was increased from the original dataset, as shown in Table 10. After improved retraining on the E2 dataset, the accuracy of ConvNeXtBase increased from 87.96% to 89.19% and the loss dropped from 0.6632 to 0.4941 at 1h 10m 15s of training time, as shown in Table 11. The

learning curves (Figure 12) showed that the model was able to generalize well. The training process steadily improved and eventually converged. The curve of training accuracy fluctuated close to 1.00 while the curve of training loss fluctuated close to 0.00 before early cut off. For the improved retraining on the E3 dataset, the accuracy of ConvNeXtBase slightly dropped from 98.61% to 97.82% and the loss slighlty increased from 0.0416 to 0.0950 at 54m 20s of training time due to applying photometric augmentation (Table 12), the dataset was more complex than the original dataset. The learning curves (Figure 13) showed behavior similar to the E2 dataset with more epochs of training.

Table 10. The quantity of data for improving the experiment

Experiment	Class	Dataset			
		Training Set		Testing Set	
		Original + Geometric Augmented	Photometric Augmented	Original + Geometric Augmented	Photometric Augmented
E2	RRIM 600	6,120	1,530	1,080	270
	RRIM 600 Seedling	6,120	1,530	1,080	270
E3	RRIM 600	3,060	3,060	540	540
	RRIM 600 Seedling	3,060	3,060	540	540

Table 11. The performance of the improved retrained models with half dataset (E2)

Improved Retrained : E2						
Model	**Accuracy**	**Loss**	**Precision**	**Recall**	**F1 - Score**	**Training Time (hh:mm:ss)**
ConvNeXtBase	0.8919	0.4941	0.9058	0.8919	0.8988	01:10:15

(a) (b)

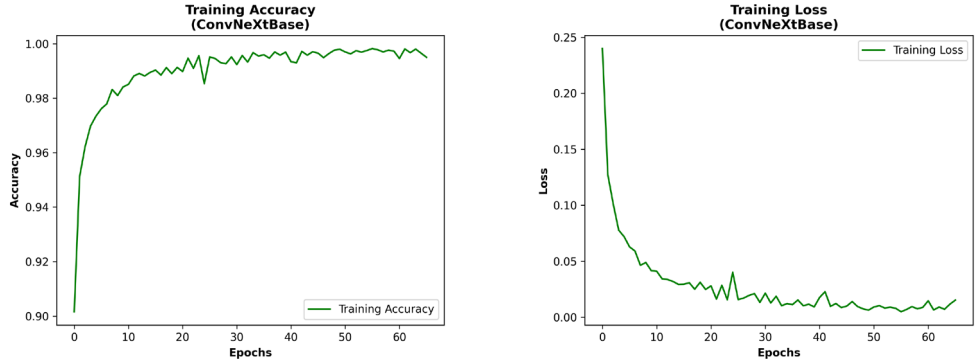

Figure 12. Training accuracy (a) and loss (b) of improved ConvNeXtBase on E2 dataset

Table 12. The performance of the improved retrained models with quarter dataset (E3)

Improved Retrained : E3						
Model	**Accuracy**	**Loss**	**Precision**	**Recall**	**F1 - Score**	**Training Time (hh:mm:ss)**
ConvNeXtBase	0.9782	0.0950	0.9786	0.9782	0.9784	00:54:20

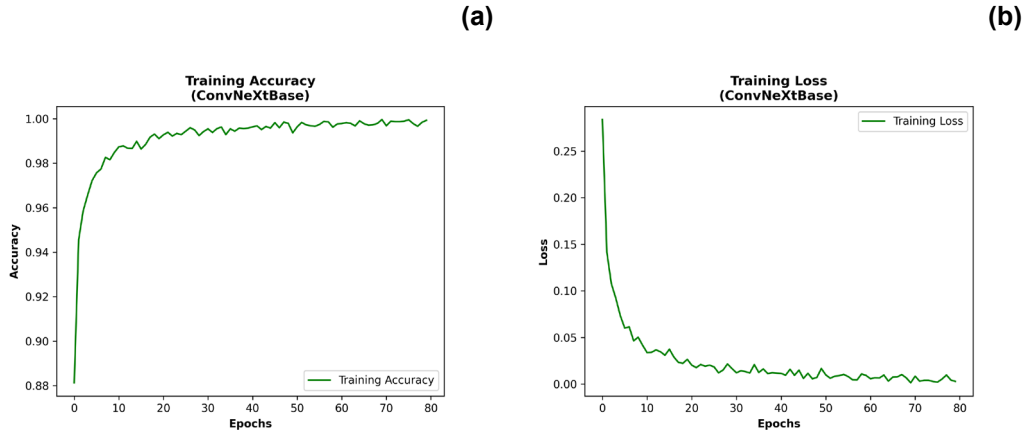

Figure 13. Training accuracy (a) and loss (b) of improved ConvNeXtBase on E3 dataset

The confusion matrix (Figure 14) clearly explained that most of the unseen testing images were correctly identified. The performance of improved ConvNeXtBase on the E3 dataset (b) was better than the E2 dataset (a). RRIM 600 seedlings were misidentified more than RRIM 600 clones due to the leaf shape variation, and some leaves were closely similar to RRIM 600 clones.

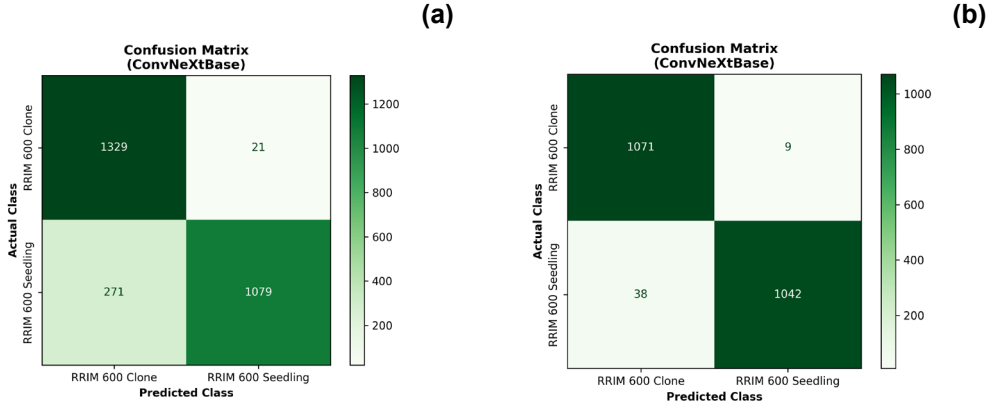

Figure 14. Confusion matrix of improved ConvNeXtBase on E2 (a) and E3 (b) datasets

3.2 Discussion

Numerous deep neural networks have been applied for target image recognition and have good accuracy. However, it is still necessary to compare the models to find the highest performance model for our dataset. Our results indicated that ConvNeXtBase was the most suitable architecture for the feature extraction process. This success in feature extraction is likely attributed to its design, which features the CNN integrated with transformer architecture. The quantity of the E3 dataset was sufficient for training the high performance Hevea clone identification model. The dataset that contained various forms of data from applying image augmentation techniques both performed in manual and computer, when it was automatically extracted by deep neural network. The useful features were comparable to those obtained from traditional feature extraction techniques but required less preprocessing (Anjomshoae et al., 2015; Anjomshoae & Rahim, 2018). After the models were trained, they provided better classification performance than using hand-crafted features as described by Huang et al. (2019), Tiwari (2020) and Kanda et al. (2021) and outperformed traditional algorithms (Thurachon & Sumethawatthanaphong, 2014; Pongsomsong & Ratanaworabhan, 2021). The results revealed that high performance Hevea clone identification model could be achieved under short duration even though it was trained on low resolution images and with less preprocessing. However, the dataset in this research was collected under controlled environment. The model cannot perform on unseen data that is very different from the training set effectively. Therefore, the diversity of the dataset is a key factor for enhancing the model to be more robust and usable in various situations.

4. Conclusions

This research proposed an approach of deep learning for Hevea clone identification to overcome the challenges of heterozygous traits that result in high variation in the dataset. The models were trained by transfer learning technique with FEA on different sizes of the datasets. The appropriate hyperparameters were found through k-fold cross validation. Considering the trade-off between accuracy and training time, the model setup with S1 was the most effective approach. Training the model on half of the dataset (E2) and a quarter of the dataset (E3) gave similar results to training on the full dataset (E1), but required less training time. In the retraining phase, MobileNetV2 achieved the highest accuracy of 89.12% at 21m 4s of training time on the E2 dataset and ConvNeXtBase achieved the highest accuracy of 98.61% at 27m 57s of training time on the E3 dataset. ConvNeXtBase outperformed classification performance at different thresholds, and it was selected to improve the performance by photometric augmentation. The accuracy was improved from 87.96% to 89.19% on the E2 dataset and slightly dropped from 98.61% to 97.82% on E3 dataset due to the larger quantity and higher complexity of the dataset. Moreover, most of unseen testing images were correctly identified. The performance of ConvNeXtBase on the E3 dataset was better than the E2 dataset in overall terms.

In conclusion, using transfer learning technique with FEA can overcome the challenge in identifying between clones and seedlings, and especially the implementation of ConvNeXtBase as feature extractor. Although ConvNeXtBase consumed a large amount of training time, it significantly outperformed other models. However, the proposed method is only an attempt to classify between RRIM600 clones and RRIM600 seedlings on a white background under controlled environment. It can further be improved by adding the environmental background and additional *Hevea* clones along with expanding the age

range of the leaves used for training. Finally, a deep learninig approach, namely the transfer learning (FEA) technique, is recommended for futher studies on specific tasks with high variation.

5. Acknowledgements

The authors would like to thank Chachoengsao Rubber Regulatory Center, Rubber Division, Department of Agriculture (DOA) for supporting data collection process in this research.

6. Authors' Contributions

All authors made significant contributions to this research and approved the final version of the manuscript.

7. Conflicts of Interest

The authors declare that they have no conflicts of interest.

ORCID
Sarayut Nonsiri https://orcid.org/0009-0007-9438-8577

References

Anjomshoae, S. T., Rahim, M. S. M., & Javanmardi, A. (2015). Hevea leaf boundary identification based on morphological transformation and edge detection. *Journal of Pattern Recognition and Image Analysis*, 25(2), 291-294. https://doi.org/10.1134/S1054661815020029

Anjomshoae, S. T., & Rahim, M. S. M. (2018). Feature extraction of overlapping Hevea leaves: a comparative study. *Journal of Information Processing in Agriculture*, 5(2), 234-245. https://doi.org/10.1016/j.inpa.2018.02.001

Arias, M., & van Dijk, P.J. (2019). What is natural rubber and why are we searching for new sources? *Journal of Frontiers for Young Minds*, 7(100), 1-9. https://doi.org/10.3389/frym.2019.00100

Balaga, O. N. R., & Patayon, U. B. (2024). Effectiveness of background segmentation algorithm and deep learning technique for detecting *anthracnose* (leaf blight) and *golovinomyces cichoracearum* (powdery mildew) in rubber plant. *Procedia Computer Science*, 234, 294-301. https://doi.org/10.1016/j.procs.2024.03.013

Bello, I., Fedus, W., Du, X., Cubuk, E. D., Srinivas, A., Lin, T.-Y., Shlens, J., & Zoph, B. (2021). Revisiting ResNets: improved training and scaling strategies. In *Proceedings of the 35th conference on neural information processing system* (pp. 22614-22627). Neural Information Processing Systems Foundation.

Bengio, Y., LeCun, Y., & Hinton, G. (2021). Deep learning for AI. *Journal of Communications of the ACM*, 64(7), 58-65. https://doi.org/10.1145/3448250

Bianco, S., Cadene, R., Celona, L., & Napoletano, P. (2018). Benchmark analysis of representative deep neural network architectures. *Journal of IEEE Access*, 6, 64270-64277. https://doi.org/10.1109/ACCESS.2018.2877890

Buquet, J., Zhang, J., Roulet, P., Thibault, S., & Lalonde J.-F. (2021). Evaluating the impact of wide-angle lens distortion on learning-based depth estimation. In *Proceedings of IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 3688-3696). IEEE. https://doi.org/10.1109/CVPRW53098.2021.00409

Chang, C.-Y., & Lai, C.-C. (2024). Potato leaf disease detection based on a lightweight deep learning model. *Machine Learning and Knowledge Extraction*, 6(4), 2321-2335. https://doi.org/10.3390/make6040114

Chollet, F. (2015). *Keras applications*. https://keras.io/api/applications

Chollet, F. (2017). Xception: deep learning with depthwise separable convolutions. In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 1800-1807). IEEE. https://doi.org/10.1109/CVPR.2017.195

Cordonnier, J.-B., Loukas, A., & Jaggi, M. (2020). On the relationship between self- attention and convolutional layers. In *Proceedings of the 8th international conference on learning representations* (pp. 1-18). ICLR.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: a large - scale hierarchical image database. In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 248-255). IEEE. https://doi.org/10.1109/CVPR.2009.5206848

Ding, B., Qian, H., & Zhou, J. (2018). Activation functions and their characteristics in deep neural networks. In *Proceedings of Chinese control and decision conference* (pp. 1836-1841). IEEE. https://doi.org/10.1109/CCDC.2018.8407425

Dodge, S., & Karam, L. (2016). Understanding how image quality affects deep neural networks. In *Proceedings of the 8th international conference on quality of multimedia Experience* (pp. 1-6). IEEE. https://doi.org/10.1109/QoMEX.2016.7498955

Dogo, E. M., Afolabi, O. J., & Twala, B. (2022). On the relative impact of optimizers on convolutional neural networks with varying depth and width for image classification. *International Journal of Applied Sciences*, 12(23), Article 11976. https://doi.org/10.3390/app122311976

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, D., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: transformers for image recognition at scale. In *Proceedings of international conference on learning representations* (pp. 1-21). ICLR.

Dumoulin, V., & Visin, F. (2018). *A guide to convolution arithmetic for deep learning*. https://arxiv.org/pdf/1603.07285

Fonseka, D., & Chrysoulas, C. (2020). Data augmentation to improve the performance of a convolutional neural network on image classification. In *Proceedings of international conference on decision aid sciences and application* (pp. 515-518). IEEE. https://doi.org/10.1109/DASA51403.2020.9317249

Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36, 193-202. https://doi.org/10.1007/BF00344251

Hassan, D. P., Fajardo, A. C., & Medina, R. P. (2022). Categorization of rubber tree seedling based in leaves using neural network. *Journal of Engineering Science and Technology*, 2022(special issue 4), 89-96.

He, K., Zhang, X., Ren S., & Sun J. (2016). Deep residual learning for image recognition. In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 770-778). IEEE. https://doi.org/10.1109/CVPR.2016.90

Hendrycks, D., Kee, K., & Mazeika, M. (2019). Using pre-training can improve model robustness and uncertainty. *Proceedings of the 36th international conference on machine learning* (pp. 2712-2721). PMLR. https://proceedings.mlr.press/v97/hendrycks19a.html

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., Le, Q. V., & Adam, H. (2019). Searching for MobileNetV3. In *Proceedings of IEEE/CVF international conference on computer vision* (pp. 1314-1324). IEEE. https://doi.org/10.1109/ICCV.2019.00140

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). *MobileNets: Efficient convolutional neural networks for mobile vision applications*. https://arxiv.org/pdf/1704.04861

Huang, G., Lui, Z., Van Der Maaten L., & Weinberger K. Q. (2017). Densely connected convolutional network. In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 2261-2269). IEEE. https://doi.org/10.1109/CVPR.2017.243

Huang, Z.-K., He, C.-Q., Wang, Z.-N. Xi, Wang H., & Hou, L.-Y. (2019). *Cinnamomum camphora* classification based on leaf image using transfer learning. In *Proceeding of the 4th advanced information technology, electronic and automation control conference* (pp. 1426-1429). IEEE. https://doi.org/10.1109/IAEAC47372.2019.8997791

Ioffe, S., & Szegedy, C. (2015). Batch normalization: accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd international conference on machine learning* (pp. 448-456). PMLR. https://proceedings.mlr.press/v37/ioffe15.html

Javid, A. M., Das, S., Skoglund, M., & Chatterjee, S. (2021). A ReLU dense layer to improve the performance of neural networks. In *Proceedings of IEEE international conference of acoustics, speech and signal processing* (pp. 2810-2814). IEEE. https://doi.org/10.1109/ICASSP39728.2021.9414269

Jepkoech, J., Mugo, D. M., Kendujywo, B. K., & Too, E. C. (2021). The effect of adaptive learning rate on the accuracy of neural networks. *International Journal of Advanced Computer Science and Applications*, 12(8), 736-751. https://doi.org/10.14569/IJACSA.2021.0120885

Kaewboonna, N., Chanakot, B., Lertkrai, J., & Lertkrai, P. (2023). Thai rubber leaf disease classification using deep learning techniques. In *Proceedings of the 6th artificial intelligence and cloud computing conference* (pp. 84-91). Association for Computing Machinery. https://doi.org/10.1145/3639592.3639605

Kanda, P. S., Xia, K., & Sanusi, O. H. (2021). A deep learning - based recognition technique for plant leaf classification. *Journal of IEEE Access*, 9, 162590-162613. https://doi.org/10.1109/ACCESS.2021.3131726

Kelleher, J. D. (2019). *Deep learning*. The MIT Press.

Kingma, D. P., & Ba, J. L. (2017). *ADAM: a method for stochastic optimization*. https://arxiv.org/pdf/1412.6980

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *Proceedings of the 26th conference on neural information processing systems* (pp. 1-9). Neural Information Processing Systems Foundation.

LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech and time series. In M. A. Arbib (Ed.). *The handbook of brain theory and neural networks* (pp. 255-258). The MIT Press.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521, 436-444. https://doi.org/10.1036/nature14539

Lei, C., Hu, B., Wang, D., Zhang, S., & Chen, Z. (2019). A preliminary study on data augmentation of deep learning for image classification. In *Proceedings of the 11th Asia-Pacific symposium in internetware* (pp. 1-6). Association for Computing Machinery. https://doi.org/10.1145/3361242.3361259

Li, G., Zhang, R., Qi, D., & Ni, H. (2024). Plant-leaf recognition based on sample standardization and transfer learning. *Applied Sciences*, 14(18), Article 8122. https://doi.org/10.3390/app14188122

Lin, M., Chen, Q., & Yan, S. (2014). *Network in network*. https://arxiv.org/pdf/1312.4400

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. & Guo, B. (2021). Swin transformer: hierarchical vision transformer using shifted windows. In *Proceedings of*

IEEE/CVF international conference on computer vision (pp. 9992-10002). IEEE. https://doi.org/10.1109/ICCV48922.2021.00986

Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A ConvNet for the 2020s. In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 11966-11976). IEEE. https://doi.org/10.1109/CVPR52688.2022.01167

Liyanage, K. K. (2021). Clone identification. In V. H. L. Rodrigo & P. Seneviratene (Eds.), *Handbook of rubber Vol. 1: agronomy* (pp. 253-269). Rubber Research Institute of Sri Lanka.

Ngugi, H. N., Akinyelu, A. A., & Ezugwu, A. E. (2024). Machine learning and deep learning for crop disease diagnosis: performance analysis and review. *Agronomy*, 14(12), Article 3001. https://doi.org/10.3390/agronomy14123001

Nibret, E. A., Mequanenit, A. M., Ayalew, A. M., Kusrini, K., & Martínez-Béjar, R. (2025). Sesame plant disease classification using deep convolution neural networks. *Applied Sciences*, 15(4), Article 2124. https://doi.org/10.3390/app15042124

Office of Agricultural Economics. (2024a). *Agricultural statistics of Thailand 2024*. https://oae.go.th/uploads/files/2025/04/30/fd747711b82231d4.pdf

Office of Agricultural Economics. (20234b). *Information of agricultural economics 2023*. https://oae.go.th/uploads/files/2025/05/06/13e8089a3f69ea96.pdf

O'Shea, K., & Nash, R. (2015). *An introduction to convolutional neural networks*. https://arxiv.org/pdf/1511.08458

Pasaribu, S. A., Basayuni, M., Purba, E., & Hasanah, Y. (2022). Leaf characterizations of IRR 400 series, BPM 24, and RRIC 100 rubber (*Hevea brasiliensis* Muell. Arg.) clone using leafgram method. *International Journal on Advanced Science Engineering Information Technology,* 12(5), 1721-1727. https://doi.org/10.18517/ijaseit.12.5.15512

Pongsomsong, P., & Ratanaworabhan P. (2021). Automatic rubber tree classification. In *Proceeding of the 18th international conference on electrical engineering/electronics, computer, telecommunications and information technology* (pp. 167-170). IEEE. https://doi.org/10.1109/ECTI-CON51831.2021.9454800

Poungchompu, S., & Chantanop, S. (2015), Factor affecting technical efficiency of smallholder rubber farming in northeast Thailand. *American Journal of Agricultural and Biological Sciences*, 10(2), 83-90. https://doi.org/10.3844/ajabssp.2015.83.90

Pratomo, B., Lisnawita, Nisa, T. C., & Basyuni, M. (2021). Short communication: digital identification approach to characterize *Hevea brasiliensis* leaves. *Journal of Biodiversitas*, 22 (2), 1006-1013. https://doi.org/10.13057/biodiv/d220257

Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., & Dollar, P. (2020). Designing network design spaces. In *Proceedings of IEEE/CVF conference on computer vision and pattern recognition* (pp. 10425-10433). IEEE. https://doi.org/10.1109/CVPR42600.2020.01044

Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386-408. https://doi.org/10.1037/h0042519

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). MobileNetV2: inverted residuals and linear bottlenecks, In *Proceedings of IEEE/CVF conference on computer vision and pattern recognition* (pp. 4510-4520). IEEE. https://doi.org/10.1109/CVPR.2018.00474

Saraswathyamma, C. K., Licy, J., & Marattukalem, G. J. (2000). Planting materials. In P. J. George & C. Kuruvilla Jacob, (Eds.). *Natural rubber: agromanagement and crop processing* (pp. 59-74). Anaswara Printing and Publishing Company.

Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. https://arxiv.org/pdf/1409.1556

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1), 1929-1958. https://dl.acm.org/doi/10.5555/2627435.2670313

Szegedy, C., Ioffe, S., Vanhoucke, V., & Alemi, A. (2017). Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Proceedings of the 31th AAAI conference on artificial intelligence* (pp. 4278-4284). Association for the Advancement of Artificial Intelligence. https://doi.org/10.1609/aaai.v31i1.11231

Szegedy, Z., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions, In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 1-9). IEEE. https://doi.org/10.1109/CVPR.2015.7298594

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Woina, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 2818-2826). IEEE. https://doi.org/10.1109/CVPR.2016.308

Tan, M., & Le, Q. V. (2019). EfficientNet: rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th international conference on machine learning* (pp. 6105-6114). ML Research Press.

Thurachon, W., & Sumethawatthanaphong W. (2014). Classification system para rubber varieties using Naïve Bayes. In *Proceedings of the 10th national conference on computing and information technology* (pp. 20-25). King Mongkut's University of Technology North Bangkok.

Tiwari, S. (2020). A comparative study of deep learning models with handcraft features and non-handcraft features for automatic plant species identification. *International Journal of Agricultural and Environmental Information Systems*, 11(2), 44-57. https://doi.org/10.4018/IJAEIS.2020040104

Xiao, K., Engstorm, L., Ilyas, A., & Madry, A., (2020). *Noise or signal: the role of image backgrounds in object recognition*. https://openreview.net/pdf?id=gl3D-xY7wLq

Xie, S., Girshick, R., Dollar, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 5987-5995). IEEE. https://doi.org/10.1109/CVPR.2017.634

Yadav, S., & Shukla, S. (2016). Analysis of k-fold cross validation over hold - out validation on colossal datasets for quality classification. In *Proceedings of IEEE the 6th international conference on advanced computing* (pp. 78-83). IEEE. https://doi.org/10.1109/IACC.2016.25

Yaiprasert, C. (2021). Artificial intelligence for para rubber identification combining five machine learning methods. *Karbala International Journal of Modern Science*. 7(3), 257-267. https://doi.org/10.33640/2405-609X.3154

Yin, X., Chen, W., Wu, X., & Yue, H. (2017). Fine-tuning and visualization of convolutional neural networks. In *Proceedings of the 12th IEEE conference on industrial electronics and applications* (pp. 1310-1315). IEEE. https://doi.org/10.1109/ICIEA.2017.8283041

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks. In *Proceedings of the 27th international conference advances in neural information processing systems* (pp. 3320-3328). Neural Information Processing Systems Foundation.

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *Proceedings of the 13th European conference on computer vision* (pp. 818-833). Springer. https://doi.org/10.1007/978-3-319-10590-1_53

Zeng, T., Li, C., Zhang, B., Wang, R., Fu, W., Wang, J., & Zhang, X. (2022). Rubber leaf disease recognition based on improved deep convolutional neural networks with a

cross-scale attention mechanism. *Frontiers in Plant Science*, 13, Article 829479. https://doi.org/10.3389/fpls.2022.829479

Zheng, S., Song, Y., Leung, T., & Goodfellow, I. (2016). Improving the robustness of deep neural networks via stability training. In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 4480-4488). IEEE. https://doi.org/10.1109/CVPR.2016.485

Zhou, Y., Song, S., & Cheung, N.-M. (2017). On classification of distorted images with deep convolutional neural networks. In *Proceedings of IEEE international conference on acoustics, speech and signal processing* (pp. 1213-1217). IEEE. https://doi.org/10.1109/ICASSP.2017.7952349

Zoph, B., Vasudevan, V., Shlens, J., & Le, Q. V. (2018). Learning transferable architecture for scaling image recognition. In *Proceedings of IEEE conference on computer vision and pattern recognition* (pp. 8697-8710). IEEE. https://doi.org/10.1109/CVPR.2018.00907