Fusion of CNN Models and Optimizers for Plant Species Identification Using Deep Learning

Tengku Mohd Afendi Zulcaffle* and Dynniellea Rhynnai Wilter
Department of Electrical and Electronics, Faculty of Engineering,
Universiti Malaysia Sarawak, Kota Samarahan, Sarawak, Malaysia

Abstract

Plant species identification plays a crucial role in biological research, ecosystems, and related studies. With recent advancements in artificial intelligence and deep learning, these technologies provide powerful alternatives to manual plant identification. However, previous research lacks comprehensive studies on the combination of CNN models and optimizers for plant species identification. In this study, convolutional neural network (CNN) models, namely ResNet50, VGG16, and EfficientNetB0, were utilized for plant species classification, and their performance was further investigated by integrating different optimizers, including Adam, SGD, and RMSProp. The models were evaluated on two fern species, namely Nephrolepis biserrata and Nephrolepis cordifolia. The dataset comprises 360 images for each class. The experimental results reveal that the fusion of EfficientNetB0 with Adam achieved the highest accuracy of 95.59%. Overall, based on average performance across optimizers, EfficientNetB0 proved to be the most effective model, while RMSProp emerged as the most consistent optimizer. These findings demonstrate that the combination of EfficientNetB0 and Adam is particularly suitable for plant species classification.

Keywords: *Plant species identification, Deep learning, Convolutional neural networks, Optimizers.*

1. Introduction

Plant species taxonomy is a fundamental aspect of ecological research and a cornerstone of biodiversity conservation. It plays a critical role in various fields, including conservation biology and agriculture. Accurate identification of plant species is essential for monitoring biodiversity, maintaining healthy ecosystems, and implementing effective farming practices. Many plant species also serve as valuable sources of therapeutic compounds, making precise classification vital for advancing medical research [1]. Reliable plant taxonomy facilitates the exploration of medicinal properties, potentially leading to the discovery of novel drugs and treatments that enhance public health. In addition, the rise of automated plant classification techniques empowers citizen scientists and local communities to actively participate in biodiversity monitoring and conservation efforts.

Traditional plant identification methods rely heavily on morphological characteristics and expert knowledge, making them often labour-intensive and susceptible to human error. As highlighted by Figueroa-Mata *et al.* [2], the manual and expert-driven approaches can be both time-consuming and subjective. In contrast, recent advances in machine learning (ML) and deep learning (DL) have introduced automated techniques that enhance the accuracy and efficiency of plant species classification.

* Corresponding author. Tel.: +6082-584527; fax: +6082583409
E-mail address: ztmafendi@unimas.my

Manuscript History:
Received 11 September, 2025, Revised 13 April, 2026, Accepted 22 April, 2026, Published 30 April, 2026

https://doi.org/10.33736/jese.10807.2026

e-ISSN:

In the field of deep learning, numerous approaches have been explored for plant species classification. For instance, ResNet-50, a convolutional neural network (CNN) that employs residual learning to train a deeper network, was utilized in [3], achieving an accuracy of 99% with data augmentation and 98.60% without it. This method enhanced classification performance by enabling the training of deep networks. However, despite its high accuracy, ResNet-50 has notable limitations, including high computational demands requiring substantial hardware resources, a risk of overfitting on small datasets, and interpretability challenges that make its decision-making process difficult to understand.

To enhance classification accuracy and leaf segmentation, Rashid *et al.* [4] proposed a hybrid model called the Plant Species Detection Stacking Ensemble Deep Learning Model (PSD-SE-DLM), which integrates MobileNetV2 and UNET, achieving over 90% accuracy. This approach combines lightweight architecture for efficiency with strong image segmentation capabilities. However, its implementation is complex and heavily reliant on high-quality training data, often requiring extensive tuning and adjustments.

Using hyperspectral images, Liu *et al.* [5] introduced a Lightweight Convolutional Neural Network (LtCNN), achieving an accuracy of over 85%. This model is optimized for identifying leaf characteristics using hyperspectral data and is less complex than traditional CNNs. Nevertheless, its performance is highly dependent on the quality of hyperspectral images and may struggle to generalize to other datasets.

Similarly, Barhate *et al.* [6] employed a Hybrid Classification Method that achieved over 97% accuracy on the Flavia dataset and 98.85% on the Swedish dataset. This method combined Enhanced Spearman's Principal Component Analysis (ESPCA) for dimensionality reduction with Deep CNN and VGG-16 for feature extraction. It required multiple steps, including dataset loading, noise removal, and data augmentation, but its complexity and heavy data dependency may affect practical implementation and performance.

In another study, Sardogan *et al.* [7] classified tomato leaf diseases using a CNN combined with Learning Vector Quantization (LVQ), which processed RGB channels to capture colour information essential for plant leaf disease detection.

In order to address the “black box” issue in plant identification using CNN, Lee *et al.* [8] employed Deconvolutional Networks (DN) to visualize learned features and enhance interpretability. This approach improved the understanding of leaf venation patterns; overall interpretability challenges persisted. However, the study did not specify the achieved accuracy, and the approach faced challenges with computational efficiency and generalization.

LeafNet, a CNN-based plant identification system developed by Barre *et al.* [9], demonstrated the ability to identify multiple plant species. However, the algorithm depends on comprehensive training datasets.

Other CNN models that were utilized by Ambarwari *et al.* [10] for plant species identification were InceptionV3 and Xception. The models achieved an accuracy of nearly 90%. However, the models required high computational requirements and reliance on large and high-quality datasets.

For essential oil plant classification, Carnagie *et al.* [11] developed a customized CNN with transfer learning using the Xception model, which achieved high accuracy (unspecified in the study) but required fine-tuning to generalize effectively.

The methods that discussed above are recently developed using deep CNN algorithms to identify plant species. This shows how important CNN in plant species classification. One of the crucial components in training deep neural network models is the optimizer. An optimizer determines how the model's parameters are updated. The performance of optimizers depends on various factors, including the dataset, model architecture, hyperparameters, and loss function [12] - [14]. Consequently, selecting the optimal optimizer to achieve the best results is often challenging.

Although the promise of deep learning for plant species identification has been demonstrated in numerous studies, a thorough study into the combination of various deep learning models and optimisers is lacking. Hence, this study is important to find the best combination of deep learning models and optimisers in terms of accuracy and efficiency in plant species identification.

In comparison with existing studies in this area, the contributions of the research presented in this paper are:

- Development of algorithms for plant species identification, specifically for fern species, using combinations of CNN models and optimizers
- Rigorous experimental study and analysis on combinations of CNN models and deep learning optimizers for plant species identification, specifically for fern species classification.

The remainder of this paper is organized as follows: Section 2 describes the development of the proposed study. Experimental results are presented in Section 3. Section 4 concludes the paper.

2. Methodology

Figure 1 illustrates the process flow diagram of the method employed in this study. The first step in training a deep learning model is data collection or acquisition. Once the data is gathered, augmentation is applied to expand the dataset and enhance training performance. Next, several models are selected and implemented using Python, TensorFlow, Keras, and other deep learning libraries. Optimizers are then integrated into each model, followed by training with separate training and validation sets. Finally, inference evaluation is conducted to assess model performance.

2.1. Data Collection and Augmentation

In this paper, we use fern species to investigate the best combination of models and optimizers for plant species identification. The fern species that involved in this research are Nephrolepis Biserrata and Nephrolepis Cordifolia, which were collected from websites in [15] and [16]. These species are not yet used for plant species identification. These websites offer information related to the species, such as ecological information, taxonomy, and distribution. The images collected from those websites are 360-degree images per species. The images come in various sizes; hence, they are resized to 240 × 240 before applying the augmentation process.

Small datasets can adversely affect the classification process, as observed in this study. Therefore, data augmentation is employed to enhance the adaptability and generalization capability of machine learning models by artificially expanding the training dataset through various transformations applied to existing images. Common augmentation techniques used in this paper include rotation, width and height changes, shear, zoom, and horizontal flipping. By providing a broader range of training examples, data augmentation helps reduce the risk of overfitting, a common issue with limited datasets. Moreover, it strengthens the model’s ability to distinguish between closely related species and improves feature extraction [17], ultimately leading to significant performance gains.

2.2 CNN Models

In this study, we propose three CNN models to be combined with optimizers: ResNet-50, VGG16, and EfficientNetB0. ResNet [18] is one of the most widely known CNN architectures. It was developed to address challenges in deep learning training, which is often time-consuming and limited by the number of layers that can be effectively trained. ResNet introduces shortcut or skip connections to overcome these limitations. Unlike many deep architectures that suffer from degraded accuracy as complexity increases, ResNet maintains strong performance while improving network training efficiency and simplifying computations. In this work, we employed ResNet-50, which consists of 50 layers, including the input and output layers.

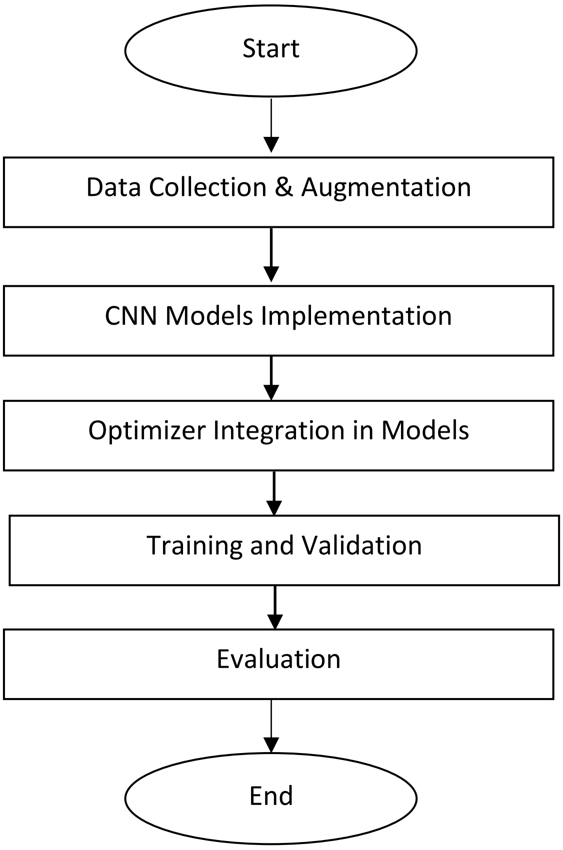

Figure 1. The process flow diagram.

The second CNN model that we propose in this paper is VGG16. The VGG16 model has achieved significant progress in image categorisation [19]. Since its initial presentation at the ILSVRC at the 2014 conference, the model has been a commonly used benchmark for a variety of computer vision applications. The VGG-16 architecture consists of 13 convolutional layers and three fully connected layers. The VGG16 model is based on the non-linear transformations using convolutional filters of smaller sizes (3x3) in each layer. The choice of this design strategy was motivated by the realisation that using several layers with smaller filters provides more expressive possibilities while producing a receptive field that is equivalent to that of a single layer with bigger filters. The simplicity and uniformity of the VGG-16 model's design are notable because it keeps the pooling layers and filter sizes constant throughout the network.

Another model applied in this study is EfficientNetB0, the baseline architecture of the EfficientNet series introduced in [20]. This model is designed to maximize accuracy while maintaining computational efficiency in image classification tasks. It employs a compound scaling strategy that proportionally scales the network's depth, width, and resolution, thereby optimizing performance while reducing computational cost. EfficientNetB0 is built using depthwise separable convolutions, which require fewer parameters and computations compared to standard convolutions. This design allows the model to effectively learn distinctive features from input images. With approximately 5.3 million parameters, the architecture incorporates batch normalization and the Swish activation function to improve training stability and generalization. EfficientNetB0 has been widely adopted in plant disease classification studies and has demonstrated excellent performance, as reported in [21~23].

All the models, ResNet-50, VGG16, and EfficientNetB0, have dropout layers to avoid overfitting. In addition, EfficientNetB0 has the DropConnect and squeeze and excitation (SE) blocks to further overcome the overfitting problems.

2.3 Optimizers

In this study, we utilize three optimizers, namely Stochastic Gradient Descent (SGD), Root Mean Square Propagation (RMSProp), and Adaptive Moment Estimation (Adam). SGD is widely recognized for its simplicity and effectiveness. It updates model parameters using a small subset of the training data, allowing for faster iterations and reduced memory usage [24]. However, despite its straightforward approach, SGD can sometimes struggle with convergence due to its sensitivity to the learning rate. This issue is especially critical in deep networks with many layers and complex connections. The purpose of using SGD in this work is twofold: to establish a baseline for comparison with more advanced optimizers, and to evaluate its effectiveness in plant species classification. According to [25], the update rule for Stochastic Gradient Descent (SGD) is expressed as follows:

$$\theta_t = \theta_{t-1} - \alpha \,.\, g(t) \tag{1}$$

where:

- α is the learning rate
- $g(t)$ represents the gradient at time t.
- θ_t is the updated parameter at time t.

In this study, we set α = 0.01, which is the default value in TensorFlow and Keras, and a common practice when using SGD.

The second optimizer that we employ in this work is the RMSProp. It was introduced to overcome the limitations of standard gradient descent by adapting the learning rate for each parameter individually. It achieves this by maintaining a moving average of the squared gradients, which helps normalize the parameter updates. This mechanism prevents the learning rate from becoming too large and causing divergence, while also avoiding excessively small updates that slow down training. As a result, RMSProp is particularly effective for handling non-stationary objectives and deep neural networks with complex architectures. The use of RMSProp in this work is intended to improve stability and convergence speed in plant species classification tasks. The update rule for RMSProp is given in [26] as follows:

$$v_t = \beta \cdot v_{t-1} + (1 + \beta) \cdot g_t^2 \tag{2}$$

where:

- $v(t)$ is the accumulated moving average of squared gradients at time t.
- β is a decay rate
- $g(t)$ represents the gradient at time

This results in the following update rule for each parameter θ_t:

$$\theta_t = \theta_{t-1} - \frac{\alpha}{\sqrt{v_t + \epsilon}} \cdot g_t \tag{3}$$

where:

- α is the learning rate.
- ϵ is a small constant added to prevent division by zero.
- θ_t is the updated parameter at time t.

In this work, the hyperparameters for RMSProp are set to: α = 0.001, β = 0.9, ϵ = 10^{-7}. The hyperparameter settings are based on TensorFlow and Keras default values, and a common practice when using RMSProp.

Adaptive Moment Estimation (Adam) is another key optimizer employed in this study. By maintaining both the squared gradients and a moving average of past gradients, Adam combines the strengths of momentum and RMSProp. This allows it to adaptively adjust learning rates for different parameters, which accelerates convergence and improves overall performance [24]. In this study, Adam is applied to optimize model training and enhance classification accuracy on the target dataset. The Adam algorithm is illustrated in Figure 2. In this study, the hyperparameters for Adam are set to: α = 0.001, β_1 = 0.9, β_2 = 0.999, ϵ = 10^{-7}. The hyperparameter settings are based on TensorFlow and Keras default values, and on common practice when using Adam.

Require: α: Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0: Initial parameter vector
  $m_0 \leftarrow 0$ (Initialize 1st moment vector)
  $v_0 \leftarrow 0$ (Initialize 2nd moment vector)
  $t \leftarrow 0$ (Initialize timestep)
  while θ_t not converged **do**
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
    $\widehat{m}_t \leftarrow m_t/(1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
    $\widehat{v}_t \leftarrow v_t/(1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
    $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t/(\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
  end while
  return θ_t (Resulting parameters)

Figure 2. The Adam optimizer algorithm [27]

3. Experimental Results and Discussion

All the proposed models have to go through a training process by inputting an image that has 224x224 pixels into a model. The batch size and the epochs were set to 32 and 30, respectively. The batch size was chosen to determine the number of samples that could be processed before adjusting the model's trainable parameters. A lower batch size improves generalisation, whereas a higher batch size speeds up training but risks overfitting. The model was set up for two classes to represent the number of plant species being classified. A categorical cross-entropy loss function was used as it is ideal for multi-class classification tasks.

Data generators were configured using *Keras' ImageDataGenerator* to allow real-time image augmentation. The augmentation process will increase the diversity of the training dataset. The common augmentation techniques, including rotation, width and height changes, shear, zoom, and horizontal flipping, were performed in this experiment. The dataset was split into training and validation with 75% for training, 15% for validation, and 10% for testing. The training validation is necessary for assessing model performance throughout the training process.

A comprehensive comparison of ResNet50, VGG16, and EfficientNetB0 using Adam, RMSProp, and SGD optimizers shows varying performance levels across the reported accuracies in Table 1. Notably, EfficientNetB0 combined with Adam achieved the highest accuracy of 95.59%. Furthermore, the combination of EfficientNetB0 and all the optimizers also achieved the highest average accuracy (93.14%). This shows the suitability of the model for plant species identification. Conversely, VGG16 with Adam yielded the lowest accuracy, and overall, VGG16 performed the worst across all optimizer combinations.

For the optimizer, the RMSProp produced the best average accuracy of the optimizer on all types of models, and its combination with EfficientNetB0 achieved the second-best result with an accuracy 94.12%, which is slightly lower than the combination of EfficientNetB0 and Adam.

Since the combination of EfficientNetB0 with the Adam optimizer achieved the highest accuracy, we further examined its training performance to evaluate the model's ability to generalize to unseen image sets. Figure 3 presents the training and validation loss curves, which demonstrate that the validation loss closely follows the trend of the training loss as the number of epochs increases. This indicates that the model does not suffer from underfitting or overfitting. In the case of underfitting, the validation loss would decrease but fail to follow the downward trajectory of the training loss. Conversely, in overfitting, the validation loss would increase while the training loss continues to decline. For comparison, Figure 4 illustrates a well-fitted model, while Figure 5 presents examples of underfitting and overfitting during training. The trend in Figure 3 is very similar to the ideal well-fitted model shown in Figure 4, indicating that it does not suffer from underfitting or overfitting. This suggests that the generated model achieves a fit that is close to the ideal well-fitted condition.

Table 1. The accuracy comparison between combinations of the models and optimizers

	Adam (%)	**RMSProp (%)**	**SGD (%)**	**Average Accuracy of Model (%)**
ResNet50 [3] (%)	88.24	91.18	92.30	90.57
VGG16 [19] (%)	82.35	86.76	85.55	84.89
EfficientNetB0 [30] (%)	**95.59**	94.12	89.71	**93.14**
Average Accuracy of Optimizer (%)	88.73	**90.69**	89.19	

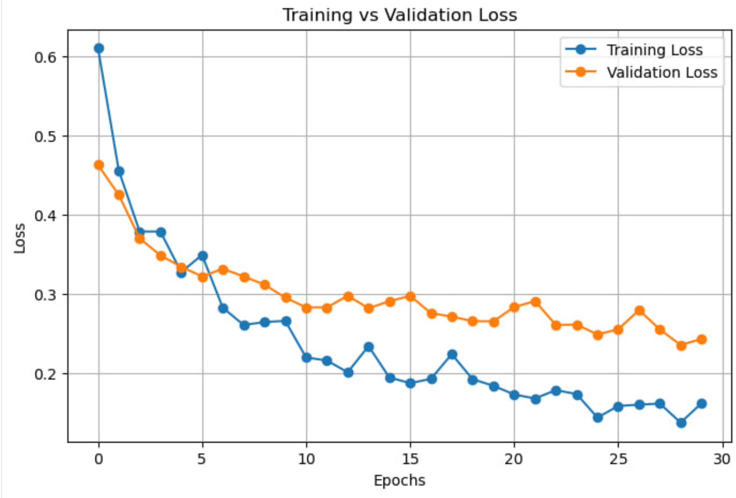

Figure 3. The training and validation loss of EfficiennetB0 with Adam

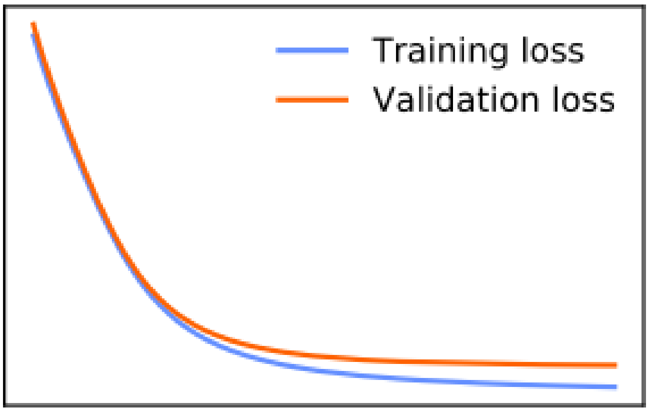

Figure 4. The ideal case of well-fitted training and validation loss.

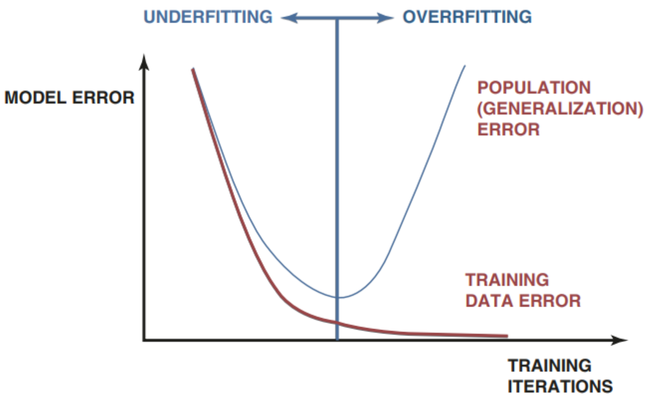

Figure 5. The underfitting and overfitting problems in training a CNN model.

6. Conclusion

In this paper, we examine the fusion of convolutional neural network models with different optimizers for plant species identification. A fern species dataset was employed to evaluate the performance of various model–optimizer combinations. Specifically, EfficientNetB0, VGG16, and ResNet50 were paired with Adam, SGD, and RMSProp.

The experimental results indicate that the combination of EfficientNetB0 with Adam achieved the highest accuracy (95.59%), demonstrating the suitability of this pairing for plant species classification. Furthermore, the loss curve indicates that the model–optimizer combination is well-fitted and suitable for identifying unseen fern species. Moreover, EfficientNetB0 consistently outperformed the other models across all three optimizers, highlighting its robustness. This is due to Mobile Inverted Bottleneck Convolution (MBConv) and Squeeze-and-Excitation (SE) blocks employed in the EfficientNetB0. These two types of layers in EfficientNetB0 help to focus on important features and improve channel-wise attention.

In terms of optimizers, RMSProp emerged as the best overall performer, surpassing both Adam and SGD. This result underscores the strength of RMSProp in effectively updating trainable parameters.

For future work, improvements may be explored by integrating the mechanisms of Adam and RMSProp to leverage their complementary advantages. Additionally, further experiments could be conducted on higher versions of EfficientNet, from B1 to B7, to assess potential performance gains.

Acknowledgements

The authors would like to express their sincere gratitude to Universiti Malaysia Sarawak and the Faculty of Engineering, particularly the Department of Electrical and Electronic Engineering, for their support in completing this research.

Conflict of Interest

We declare no conflict regarding the publication of the study.

References

[1] Süntar, I. (2020). Importance of ethnopharmacological studies in drug discovery: role of medicinal plants, *Phytochem Rev,* 19, 1199–1209. https://doi.org/10.1007/s11101-019-09629-9

[2] Figueroa-Mata, G., Mata-Montero, E., Valverde-Otarola, J. C., & Arias-Aguilar, D. (2018). Using deep convolutional networks for species identification of Xylotheque samples. In *Proceedings of the 2018 IEEE International Work Conference on Bioinspired Intelligence (IWOBI)*, 1-9. https://doi.org/10.1109/IWOBI.2018.8464216.

[3] Hama, H. M., Taib, S. A., & Omer, S. M. (2024). Houseplant leaf classification system based on deep learning algorithms. *Journal of Electrical Systems and Information Technology*, 11(18). https://doi.org/10.1186/s43067-024-00141-5

[4] Rashid, J., Khan, I., Abbas, I. A., Saeed, M. R., Saddique, M., & Abbas, M. (2023). A hybrid deep learning approach to classify the plant leaf species. *Computers, Materials & Continua*, 76(3), 3897–3920. https://doi.org/10.32604/cmc.2023.040356

[5] Liu, K.H., Yang, M.H.., Huang, S.T., & Lin, C. (2022). Plant species classification based on hyperspectral imaging via a lightweight convolutional neural network model. *Frontiers in Plant Science*, 13. https://doi.org/10.3389/fpls.2022.855660

[6] Barhate, D., Pathak, S., & Dubey, A.K. (2023). Hyperparameter-tuned batch-updated stochastic gradient descent: Plant species identification by using hybrid deep learning. *Ecological Informatics*, 75. https://doi.org/10.1016/j.ecoinf.2023.102094

[7] Sardogan, M., Tuncer, A., & Ozen, Y. (2018). Plant leaf disease detection and classification based on CNN with LVQ algorithm. In *Proceedings of the 2018 3rd International Conference on Computer Science and Engineering (UBMK)*. 382-385. https://doi.org/10.1109/ubmk.2018.8566635

[8] Lee, S.H., Chan, C. S., Wilkin, P., & Remagnino P. (2015). Deep-plant: Plant identification with convolutional neural networks. In *Proceedings of the* 2015 *IEEE International Conference on Image Processing (ICIP)*, 452-456. https://doi.org/10.1109/ICIP.2015.7350839

[9] Barré, P., Stöver, B. C., Müller, K. F., & Steinhage, V. (2017). LeafNet: A computer vision system for automatic plant species identification. *Ecological Informatics*, 40, 50–56. https://doi.org/10.1016/j.ecoinf.2017.05.005

[10] Ambarwari, A., Adrian, Q. J., Herdiyeni, Y., & Hermadi, I. (2020). Plant species identification based on leaf venation features using SVM. *TELKOMNIKA Telecommunication Computing Electronics and Control*, 18(2). https://doi.org/10.12928/telkomnika.v18i2.14062

[11] Carnagie, J. O., Prabowo, A. R., Budiana, E. P., & Singgih, I. K. (2022). Essential oil plants image classification using Xception model. *Procedia Computer Science*, 204, 395–402. https://doi.org/10.1016/j.procs.2022.08.048

[12] Soydaner D., (2020). A comparison of optimization algorithms for deep learning. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(13). https://doi.org/10.1142/S0218001420520138

[13] Chen, C.H., Lai, J.P., Chang, Y.M., Lai, C.J., & Pai, P.F. (2023). A study of optimization in deep neural networks for regression. *Electronics*. 12(14). https://doi.org/10.3390/electronics12143071

[14] Chen, X, Liang, C., Huang, D., Real, E., Wang, K., Pham, H., Dong, X., Luong, T., Hsieh, C.J., Lu, Y., & Le, Q.V. (2024). Symbolic discovery of optimization algorithms. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NIPS '23)*, 49205 - 49233. https://dl.acm.org/doi/10.5555/3666122.3668262

[15] Schott L. (2025). Nephrolepis Biserrata (Sw.) Schott. *Global Biodiversity Information Facility*. www.gbif.org/species/2650936

[16] Presl, C. (2025). Nephrolepis Cordifolia (L.) C.Presl. *Global Biodiversity Information Facility*. www.gbif.org/species/2650930

[17] Shorten, C., & Khoshgoftaar, T.M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(60). https://doi.org/10.1186/s40537-019-0197-0

[18] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770-778, https://doi.org/10.1109/CVPR.2016.90

[19] Sahithi, B., & Vigneshwari S. (2023). Exploring the performance of VGG16 and EfficientNet models for plant disease classification: A comparative approach. In *Proceedings of 2023 First International Conference on Cyber Physical Systems, Power Electronics and Electric Vehicles (ICPEEV)*, 1-6. https://doi.org/10.1109/ICPEEV58650.2023.10391936

[20] Tan, M., & Le, Q.V. (2019). EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of 36th International Conference on Machine Learning (ICML 2019)*, 6105-6114. http://proceedings.mlr.press/v97/tan19a.html

[21] Fenu, G., & Malloci, F.M. (2021). DiaMOS Plant: A dataset for diagnosis and monitoring plant disease. *Agronomy*, 11(11). https://doi.org/10.3390/agronomy11112107

[22] Fenu, G., & Malloci F. M. (2023). Classification of pear leaf diseases based on ensemble convolutional neural networks. *AgriEngineering*, 5(1),141-152. https://doi.org/10.3390/agriengineering5010009

[23] Alirezazadeh, P., Schirrmann, M., & Stolzenburg, F. (2023). Improving deep learning-based plant disease classification with attention mechanism. *Gesunde Pflanzen*, 75:49–59. https://doi.org/10.1007/s10343-022-00796-y

[24] Singh, N., Srivastava, R., Mahdi, H. F., Sati S., Joshi P., & Sar A. (2024). A comparative study of different optimisers on convolutional neural networks for bird classification. In *Proceedings of 2024 International Conference on Artificial Intelligence and Emerging Technology (Global AI Summit)*, 852-857. https://doi.org/10.1109/GlobalAISummit62156.2024.10947947

[25] Kartowisastro, I. H., & Latupapua, J. (2023). A comparison of Adaptive Moment Estimation (Adam) and RMSProp optimisation techniques for wildlife animal classification using convolutional neural networks. *Revue d'Intelligence Artificielle*, 37(4), 1023-1030. https://doi.org/10.18280/ria.370424

[26] Cheng, W., Pu, R., & Wang, B. (2025). AMC: Adaptive learning rate adjustment based on model complexity. *Mathematics*, 13(4), 650. https://doi.org/10.3390/math13040650

[27] Kingma, D.P. & Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. http://arxiv.org/abs/1412.6980

[28] Li, H., Rajbahadur, G. K., Lin, D. , Bezemer, C. P., & Jiang, Z. M. (2024). Keeping deep learning models in check: A history-based approach to mitigate overfitting. *IEEE Access*, 12, 70676-70689. https://doi.org/10.1109/ACCESS.2024.3402543

[29] Aliferis, C., & Simon G. (2024). Overfitting, underfitting and general model overconfidence and under-performance pitfalls and best practices in machine learning and AI. In Simon, G.J., & Aliferis C. (Eds.). *Artificial intelligence and machine learning in health care and medical sciences: Best practices and pitfalls*. Springer, Cham, CH. https://doi.org/10.1007/978-3-031-39355-6

[30] K. Kiran and A. Kaur, "Deep Learning-Based Plant Species Identification using VGG16 and Transfer Learning: Enhancing Accuracy in Biodiversity and Agricultural Applications," 2025 3rd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), Bengaluru, India, 2025, pp. 1538-1543, doi: 10.1109/IDCIOT64235.2025.10915124.