Innovative Informatics and Artificial Intelligence Research (IIAIR)
Vol. 1, issue 1, pp. 19-25, 2025
Received 18 Dec 2024; accepted 23 March 2025; published 30 Apr 2025
https://doi.org/10.35718/iiair.v1i1.1277

BBG52: A New Dataset for Plant Species Recognition in the Balikpapan Botanical Gardens, Borneo Island

Rahmat Ramadhani [1], Gusti Ahmad Fanshuri Alfarisy [1], and Boby Mugi Pratama[1]

[1]Department of Informatics, Institut Teknologi Kalimantan, Balikpapan 76127, Indonesia

Corresponding author: Gusti Ahmad Fanshuri Alfarisy (gusti.alfarisy@lecturer.itk.ac.id)

To cite this article: R. Ramadhani, G. A. F. Alfarisy, B. M. Pratama, "BBG52: A New Dataset for Plant Species Recognition in the Balikpapan Botanical Gardens, Borneo Island," *Innovative Informatics and Artificial Intelligence Research*, vol. 1, issue 1, 2025. [Online]. Available: https://doi.org/10.35718/iiair.v1i1.127

Gusti Ahmad Fanshuri Alfarisy serves as an Editor of IIAIR but was not involved in the peer-review process of this article

Abstract

The Balikpapan Botanical Garden serves as a conservation area in Indonesia for preserving biodiversity, particularly the endemic species of Kalimantan. Accurate and efficient identification and classification of plant species are crucial for conservation efforts. However, traditional methods are often time-consuming and require expert knowledge, highlighting the need for an automated approach. In this study, we manually collected a dataset of natural images in the Balikpapan Botanical Garden that simulates real-world conditions and contain 5,200 image samples of 52 different plant species named BBG52. We compared the manual train-test data splitting by considering intra-class variants against random splitting to evaluate the performance differences. To construct a classification model, we employed ResNet variants as pre-trained models—ResNet-34, ResNet-50, and ResNet-101— and examined the effect of the hidden layer in the classification part of the model. Our empirical results demonstrate that manual data splitting yields better performance than random splitting. Furthermore, the ResNet-50 model without additional hidden layers achieved the best performance with an accuracy of 96.88% and F1-score of 0.9689. The computational analysis provided empirical evidence that the model runs efficiently, requiring 0.1379 seconds on a CPU and 0.0861 seconds on a GPU, demonstrating the model's efficiency for constrained device. The BBG52 dataset is openly accessible at https://github.com/inidhanii/BBG52

Keywords: computer vision; plant classification; balikpapan botanical gardens; residual network; transfer learning; deep learning; machine learning

1. Introduction

Plant biodiversity is one of the most valuable assets on Earth, playing a crucial role in maintaining ecological balance. It not only provides oxygen and essential resources for humans but also serves as food and shelter for various animal species [1]. Unfortunately, biodiversity loss due to deforestation, climate change, and human activities has threatened the balance of plant species. A recent study has estimated that 45% of flowering plant species are at risk of extinction [2]. One such conservation effort is the Balikpapan Botanical Gardens located in Balikpapan, East Borneo. It has a role of preserving Indonesia's biodiversity, particularly the endemic species of Borneo [3].

Accurate and efficient plant species classification is a crucial aspect of conservation efforts but remains challenging due to the immense number of species and limited available expertise [4]. Traditional classification methods require extensive manual effort, such as botanists manually identifying species from images. This process is not only time-consuming but also prone to human error, highlighting the need for an automated approach [5]. Meanwhile, automated classification systems offer a solution to overcome these limitations and enable large-scale identification with high efficiency.

The rapid development in the deep learning field, particularly Convolutional Neural Networks (CNN) has revolutionized image classification and identification tasks, which include plant species classification [6]. CNN architectures such as Residual Networks (ResNet) introduce skip connections to allow the model to have lots of layers without experiencing vanishing gradient issues [7]. Prior studies often used scanned leaf images for classification [8, 9] . While effective, this approach has challenges, including the need to scan leaves in optimal positions and conditions, which often requires specialized equipment and requires more labor. This becomes impractical for large-scale applications. Meanwhile, some studies have explored the use of natural images captured directly in real-world environments [10, 11]. This approach offers practical advantages for real-world applications as it significantly reduces the complexity and time required for image acquisition by removing the need for complex scanning procedures.

While CNNs like ResNet have shown significant potential in image classification, their effectiveness relies on the availability of sufficient and high-quality datasets. Common issues such as small dataset sizes, class imbalance, and labeling errors can all negatively impact the performance of a model [12]. To overcome this limitation, transfer learning was introduced as an effective solution. By leveraging models pre-trained on large datasets such as ImageNet, transfer learning enables efficient feature extraction for more specific tasks, such as plant image classification, while reducing training time and mitigating challenges that might exist by using smaller datasets [6]. Prior studies have demonstrated how using transfer learning can enhance model performance by utilizing prior knowledge, such as shape and texture recognition [13, 14].

This study aims to propose a new dataset in plant species identification and to investigate the potential of using the Residual Network architecture with transfer learning to classify plant species at the Balikpapan Botanical Gardens using natural images captured using mobile devices. In summary, the contributions of this study are as follows:

1. Proposing a new dataset for plant species in the Balikpapan Botanical Gardens that can be utilized for future research in the relevant industry named *BBG52*.
2. Exploring how data splitting methods affect model performance: manual vs. random.
3. Exploring the performance of Residual Network architecture using transfer learning for the classification of plant species within our new proposed dataset.
4. Exploring how additional hidden layers affect model performance.

2. Related Works

Prior studies in the field of automatic plant classification have demonstrated the potential of various deep learning models. The most widely used approach is Convolutional Neural Networks (CNN), which are particularly effective in processing image data to classify plant species. Pujiati and Rochmawati [9] investigated the classification of herbal plant leaves using CNN. Their model, consisting of three convolutional layers and two fully connected layers, was utilized to classify 33 species of herbal plants using a dataset consisting of 21,450 leaf images. The results showed an accuracy of 84% during testing, demonstrating the capability of CNN to handle complex classification problems, particularly for species with morphologically similar features.

A study by Falahkhi et al. [13] compared the performance of two pre-trained models utilizing transfer learning, ResNet and AlexNet for flower classification. They utilized the Flower102 dataset with 8,189 images across 102 categories. The study found that ResNet achieved an accuracy of 97.6%, outperforming AlexNet, which achieved 90.2%. This result highlights the advantages of transfer learning. Pre-trained models on large datasets, such as ImageNet, can adapt effectively to smaller, domain-specific datasets. Moreover, the capability of ResNet to address gradient vanishing problems in deep architectures contributed to its superior performance.

Musyaffa et al. [14] conducted a study on the classification of Indonesian herbal plants using transfer learning. Five CNN models—ResNet-34, DenseNet, VGG11, ConvNeXt, and Swin Transformer—were evaluated using a dataset derived from the Vietnam Medicinal Plant dataset [15] consisting of 20,000 images across 200 categories. They also introduced a new dataset of 100 Indonesian herbal plant categories collected via Google Images, comprising 10,000 images. They employed augmentation techniques to address the imbalance in certain classes. The best performance was achieved by ConvNeXt, with an accuracy of 92.5%. This study highlights the efficacy of transfer learning and data augmentation in addressing the limitations of small datasets.

Atique et al. [8] focused on species classification using leaf venation patterns. They applied ResNet and DenseNet models to the MalayaKew dataset, which contains 44 tropical tree species with 64 images per category. Using Canny edge detection to extract venation features, the study found that DenseNet-169 achieved the highest accuracy of 95.72% using Adam optimizer, outperforming ResNet-101, which achieved 89.50%. This result highlights DenseNet's effectiveness in handling venation-based plant classification tasks.

Sun et al. [10] explored plant classification in natural environments using a deep learning model with 26 layers and 8 residual blocks to classify ornamental plant species at Beijing Forestry University (BJFU). Using the BJFU100 dataset, which comprises 10,000 mobile-captured images of 100 species, their ResNet-26 model achieved an accuracy of 91.78%. This study demonstrated the suitability of ResNet architectures for handling natural images and mobile-acquired datasets. Meanwhile, Bodhwani et al. [11] applied ResNet-50 for plant species classification using the LeafSnap dataset, which includes 30,866 leaf images from 185 tree species in the northeastern United States. Their model achieved an accuracy of 93.09%, demonstrating the robustness of ResNet in classifying plant species, especially under natural environmental conditions.

In summary, the mentioned studies collectively emphasize the significant contributions of CNN and transfer learning techniques in advancing plant species classification, especially when addressing challenges posed by limited data and using images captured in a natural environment.

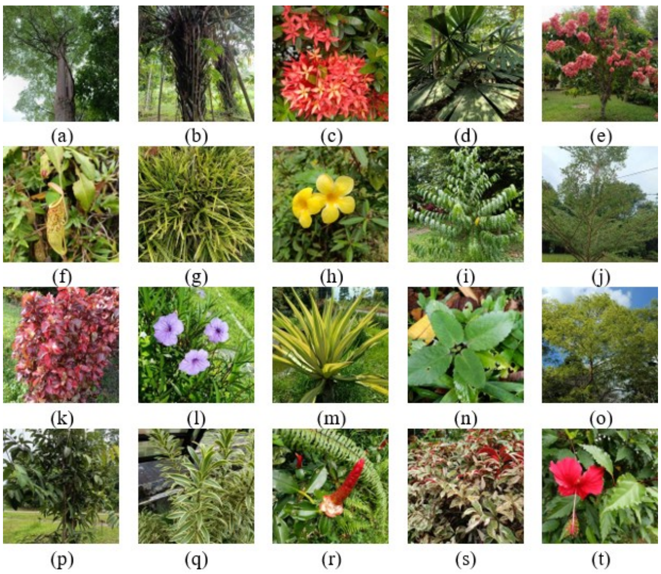


Figure 1: Sample images from the BBG52 dataset.

3. Proposed BBG52 Dataset

The BBG52 dataset consists of images collected from 52 different species located in the Balikpapan Botanical Gardens, Balikpapan City. The dataset is captured using a mobile device under various lighting conditions, angles, and distances. Each class contains 100 images, which makes a total of 5,200 images in the dataset.

Sample images from the BBG52 dataset are presented in Figure 1. The sample includes (a) *Alstonia angustiloba Miq,* (b) *Borassodendron borneensis J. Dransf*, (c) *Ixora javanica*, (d) *Licuola spinosa*, (e) *Mussaenda philippica*, (f) *Nepenthes mirabilis Druce*, (g) *Pandanus pygmaeus Thouars*, (h) *Allamanda cathartica L*, (i) *Strombosia ceylanica Gardn*, (j) *Terminalia mantaly H.Perrier*, (k) *Acalypha wilkesiana Müll.Arg*, (l) *Ruellia simplex C.Wright*, (m) *Furcraea foetida (L.) Haw*, (n) *Kalanchoe pinnata*, (o) *Koompassia excelsa (Becc.) Taub*, (p) *Vatica umbonata Burck*, (q) *Dracaena reflexa Lam*, (r) *Costus woodsonii Maas*, (s) *Excoecaria cochinchinensis Lour*, and (t) *Hibiscus rosa-sinensis L*.

4. Methods

4.1 Data Preparation

The dataset will be split with a 60-40 ratio, with 60% of the data used for training and 40% for validation/testing. A 60:40 split is chosen based on the approach used in the Vietnam Medicinal Plant dataset [15]. With 5,200 images, the training set will contain 3,120 images, while the validation will contain 2,080 images. The data splitting process will be performed both manually and randomly, resulting in two dataset variations.

In the manual split, images are selected into either the training or validation set while ensuring intra-class variations such as differences in lighting, physical conditions of the plants, or growth stages are represented in both the training and validation sets. By maintaining this representation, the model is expected to recognize patterns more effectively and produce better performance on both the training and validation data. Meanwhile, the random split will be carried out using the *train_test_split* function in Scikit-learn python library, ensuring a random division of the data with the predetermined ratio.

Image augmentation using RandAugment [16] will be undertaken to add variation to the training set and ensure model robustness, resulting in better generalization of the model. All images will be resized to a dimension of 224x224 pixels to reduce computational load during the training process and normalized using ImageNet mean and standard deviation values to ensure effective knowledge transfer.

4.2 ResNet Model Implementation

Residual Network (ResNet) is a convolutional neural network architecture designed to overcome the vanishing gradient problem, where the gradient diminishes as layers increase, slowing weight updates and hindering performance. ResNet uses skip connections that allow the input of a layer to bypass the next layer and flow into the next layer, stabilizing the network during training. This technique, known as a residual block, works based on Equation 1.

$$x_{l+1} = x_l + F(x_l, W_l) \tag{1}$$

In Equation 1, x_l is the input to the next layer. $F(x_l, W_l)$ represents the transformation applied at that layer, with W_l as its weights. The transformation $F(x_l, W_l)$ is added to x_l, so the output of the residual block is the combination of the original data and the transformed result, which is x_{l+1}, the residual block ensures that important information is not lost during transformation [7, 17].

This study will evaluate three pre-trained Residual Network models: ResNet-34, ResNet-50, and ResNet-101 acquired from the PyTorch library [18]. These models are chosen to evaluate the dataset because of the different complexities of each model. The implementation begins by loading the pre-trained models, which were initially trained on the ImageNet dataset. Each model will undergo a freezing process where the convolutional layers are frozen, and only the final classification layer will be trained. This approach allows the model to retain the knowledge learned from ImageNet while adjusting the final classification layer to suit the plant species classification task using the collected dataset. To adapt to the number of plant species classes in the dataset, the final classification layer of each ResNet model is replaced with a new fully connected layer with 52 outputs matching the number of classes in the collected dataset.

During the training phase, only the newly added fully connected layer is trained, while the convolutional layers remain frozen to preserve the weights learned from ImageNet. This technique helps prevent overfitting and accelerates model convergence, as the convolutional layers already contain strong feature representations learned from the ImageNet dataset.

4.3 Evaluation Metrics

In this study, the model performance is evaluated based on accuracy and F1-Score. Accuracy is a key metric for evaluating a classification model, representing the proportion of correctly predicted instances (both positive and negative) out of the total cases. Accuracy is calculated as presented in Equation 2.

$$accuracy = \frac{TP + TN}{P + N} \tag{2}$$

In Equation 2, TP is True Positives, TN is True Negatives, FP is False Positives, and FN is False Negatives, which are the different categories of predictions. While accuracy provides a general performance measure, it may be misleading in cases of class imbalance, as it does not account for the model's performance across all classes.

Precision evaluates the quality of positive predictions by calculating the ratio of true positives (TP) to the sum of true positives and false positives (FP) as shown in Equation 3:

$$precision = \frac{TP}{TP + FP} \tag{3}$$

Precision metric is crucial in scenarios where the cost of false positives is high, such as medical diagnostics, ensuring that predicted positives are mostly accurate.

Recall, or sensitivity, measures a model's ability to identify all relevant positive cases, calculated as the ratio of true positives to the total number of actual positives (TP + FN), which is shown in Equation 4.

$$recall = \frac{TP}{TP + FN} \quad (4)$$

Recall is essential in contexts where missing positive cases could have serious consequences, such as disease detection, where high recall ensures most true positives are identified.

The F1-Score combines both precision and recall into a single measure, representing the harmonic mean of the two as presented in Equation 5.

$$F1_{score} = \frac{2 \times precision \times recall}{precision + recall} \quad (5)$$

The F1-Score is particularly useful for imbalanced datasets, as it provides a balanced view of the model's performance, ensuring that both false positives and false negatives are considered.

The mean computation time of each model will also be evaluated to see the efficiency of the three ResNet models after acquiring the best hidden layer configuration.

5. Experimental Settings

Each ResNet model will be tested with three hidden layer configurations: no additional hidden layer, one hidden layer, and two hidden layers. Each additional hidden layer will have 512 neurons and will be followed by a ReLU activation function. We set the hyperparameters of the models including the number of epochs, learning rate, batch size, and optimizer as shown in Table 1. We employed the cross-entropy loss as the loss function to train the model.

Table 1: Hyperparameter settings

Hyperparameter	Value
Epoch	50
Optimizer	Adam
Learning Rate	0.001
Batch Size	32

For the device, we used an NVIDIA GTX 1650 Ti 4GB GPU with CUDA support for faster parallel computations. The training process was conducted on Windows 11 operating system.

The experiment evaluates the impact of manual versus random data splitting on the performance of the baseline ResNet-34 model (without additional hidden layers). This aims to evaluate how data splitting methods influence the performance of the model.

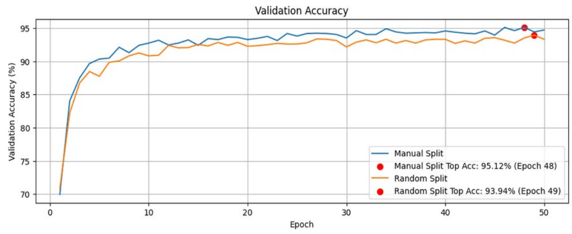
The next experimentation is performed on all three ResNet models, with each tested with three configurations, including no hidden layers, one hidden layer, and two hidden layers utilizing 512 neurons for each hidden layer. Manual splitting ensures consistent data partitioning in this experiment.

After identifying the best hidden layer configuration, we will evaluate the computation time of each model. This involves 100 samples to perform inference through the model using random images from the collected dataset. The computation time will be tested on a AMD Ryzen 5 4600H 3.00 GHz CPU and NVIDIA GTX 1650 Ti 4GB GPU. The mean computation time across all 100 samples is calculated to derive representative computational time.

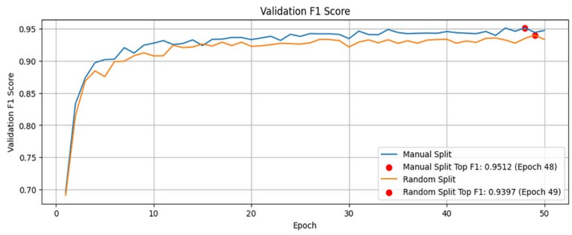
6. Results and Discussions

6.1 Data Splitting Methods Testing

Testing the effect of data splitting was performed using a baseline model ResNet-34 with no additional hidden layers. The testing was conducted twice using two dataset variants: one dataset was manually split with attention to intra-class variation, and the other was automatically split using a random train-test split.



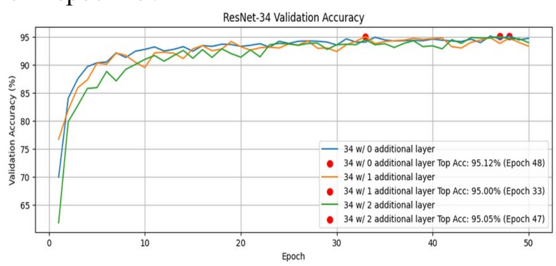
(a)



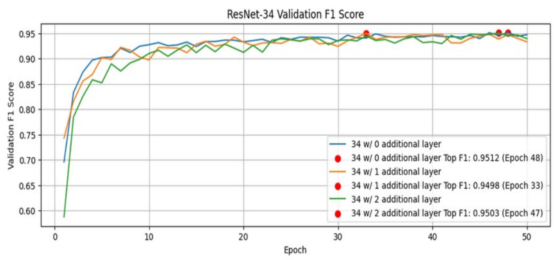
(b)

Figure 2: Graph of Accuracy and F1-Score using two different data splitting methods.

Based on Figure 2, the performance of the ResNet-34 model with manually and randomly split datasets can be observed. The model using the manually split dataset achieved the highest accuracy of 95.12% as shown in Figure 2a and an F1-Score of 0.9512 at epoch 48 as shown in Figure 2b, whereas the model using the randomly split dataset achieved the highest accuracy of 93.94% and an F1-Score of 0.9397 at epoch 49.



(a)



(b)

Figure 3: Graph of Accuracy and F1-Score on ResNet-34 with Various Additional Hidden Layers

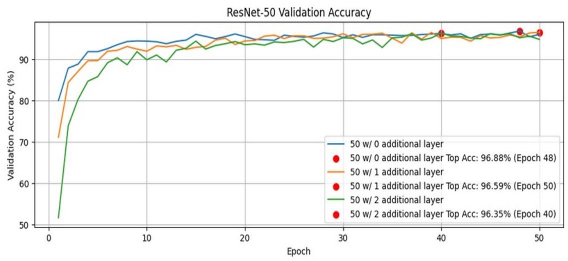
Both models showed improved performance as epochs increased. However, from epoch 5 onward, the manually split dataset consistently outperformed the randomly split dataset in accuracy and F1-score. This difference remained consistent until the end of training, where the manually split dataset yielded 1.18% higher accuracy and 0.0115 higher F1-Score than the randomly split dataset.

One scenario explaining this result is that manually splitting the data, while more time-consuming due to the need to select each image individually, ensures a more balanced distribution between training and testing data, reflecting the actual distribution in the dataset and real-world conditions. In contrast, random splitting may create an imbalanced class distribution, reducing the model's ability to generalize.

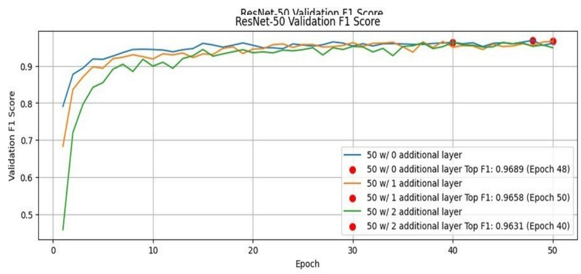
6.2 ResNet Models Testing

Testing was conducted on three ResNet models: ResNet-34, ResNet-50, and ResNet-101, with three variations for each model: no additional hidden layers, one additional hidden layer, and two additional hidden layers. Each additional hidden layer consisted of 512 neurons, and the ReLU activation function was used. Evaluation results for ResNet-34, ResNet-50, and ResNet-101 are shown in Figure 3, Figure 4 and Figure 5 respectively.

Based on Figure 3, the performance of three variants of the ResNet-34 models can be observed. The ResNet-34 model without additional hidden layers achieved the highest accuracy of 95.12% as shown in Figure 3a and an F1-Score of 0.9512 at epoch 48 as shown in Figure 3b. Meanwhile, the ResNet-34 model with one additional hidden layer achieved the highest accuracy of 95% and an F1-Score of 0.9498 at epoch 33, and the model with two additional hidden layers achieved the highest accuracy of 95.05% and an F1-Score of 0.9503 at epoch 47.



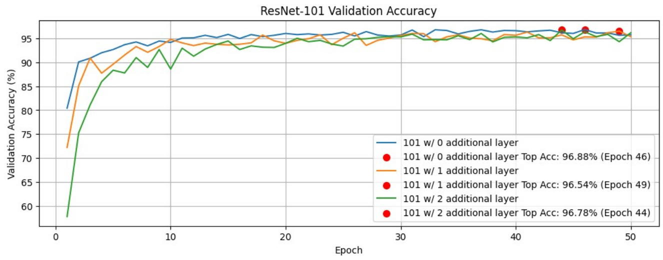
(a)



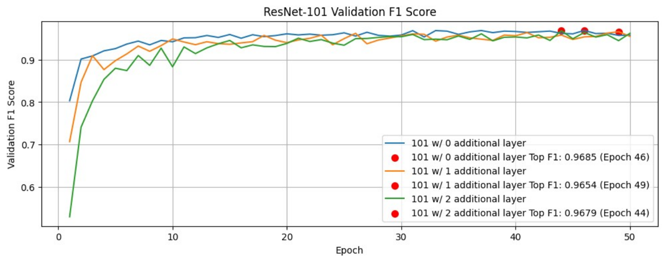
(b)

Figure 4: Graph of Accuracy and F1-Score on ResNet-50 with Various Additional Hidden Layers

Based on Figure 4, the performance of the three variants of the ResNet-50 models can be observed. The ResNet-50 model without additional hidden layers achieved the highest accuracy of 96.88% as shown in Figure 4a and an F1-Score of 0.9689 at epoch 48 as shown in Figure 4b. Meanwhile, the ResNet-50 model with one additional hidden layer achieved the highest accuracy of 96.59% and an F1-Score of 0.9658 at epoch 50, and the model with two additional hidden layers achieved the highest accuracy of 96.35% and an F1-Score of 0.9631 at epoch 40.



(a)



(b)

Figure 5: Graph of Accuracy and F1-Score on ResNet-101 with Various Additional Hidden Layers

Based on Figure 5, the performance of the three variants of the ResNet-101 models can be observed. The ResNet-101 model without additional hidden layers achieved the highest accuracy of 96.88% as shown in Figure 5a and an F1-Score of 0.9685 at epoch 46 as shown in Figure 5b. Meanwhile, the ResNet-101 model with one additional hidden layer achieved the highest accuracy of 96.54% and an F1-Score of 0.9654 at epoch 49, and the model with two additional hidden layers achieved the highest accuracy of 96.78% and an F1-Score of 0.9679 at epoch 44.

The performance patterns across all three ResNet models were consistent: adding hidden layers resulted in a slight decline in performance. For ResNet-34, the model without additional hidden layers achieved the best accuracy and F1-Score, with declines of 0.12% in accuracy and 0.0014 in F1-Score for one additional hidden layer and declines of 0.7% in accuracy and 0.0009 in F1-Score for two additional hidden layers. Similar trends were observed in ResNet-50 and ResNet-101, where the highest-performing configurations were those without additional hidden layers. The results indicate that for this dataset, simpler configurations yielded better results, likely due to the small dataset size or limited feature space, where additional layers may introduce unnecessary noise instead of improving performance. Among all tested models, ResNet-50 achieved the best overall results with an accuracy of 96.88% and an F1-Score of 0.9689.

6.3 Time Complexity Evaluation

Time complexity evaluation was conducted on the best-performing configurations of ResNet-34, ResNet-50, and ResNet-101, which are the models without additional hidden layers. The devices used were a CPU and a GPU with CUDA support. Computation time was measured starting from the image transformation process, including resizing the image to

224x224 pixels, converting the image to a tensor, normalizing the image, and the image inference itself. The results are shown in Figure 6.

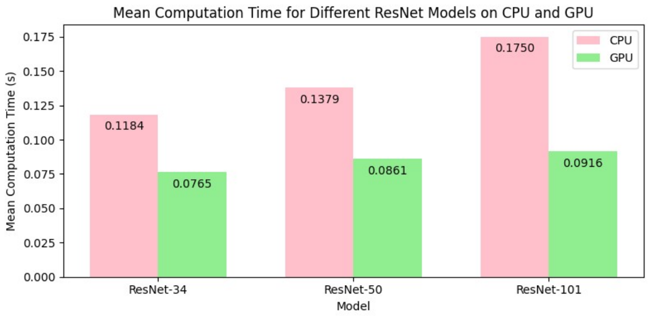


Figure 6: Graph of Computation Time Testing

All models demonstrated rapid computation times, requiring less than one second per image. However, differences in processing times were more pronounced on the CPU than the GPU. For instance, ResNet-34 was 0.0195 seconds faster than ResNet-50 and 0.0566 seconds faster than ResNet-101 on the CPU. On the GPU, ResNet-34 was only 0.0096 seconds faster than ResNet-50 and 0.0151 seconds faster than ResNet-101.

These differences most likely arise from the complexity and parameter counts of each model. ResNet-34, being a simpler model, required less processing time, whereas ResNet-101, with its more complex architecture, had the longest processing time. While smaller models like ResNet-34 process images faster, the differences in computation time are relatively small and unlikely to have a significant impact on most real-time applications.

7. Conclusions and Future Study

This study introduces BBG52, a new dataset for plant species classification containing 5200 natural images of 52 different plant species acquired using a mobile device. Additionally, the experiment results demonstrate that manually splitting the dataset, with attention to intra-class variation, may result in better model performance compared to random splitting, as shown by the ResNet-34 achieving 95.12% accuracy and an F1-Score of 0.9512 with the manually split dataset. Among the tested models, ResNet-50 achieved the highest accuracy of 96.88%, an F1-Score of 0.9689, and computation times of 0.1379 seconds on a CPU and 0.0861 seconds on a GPU. The findings highlight the potential of deep learning models like ResNet in automating plant species classification, which could be further applied in real-world scenarios such as mobile or web applications.

For future studies, the dataset can be expanded by adding new species or increasing the number of images per species. The dataset also provides opportunities to explore other CNN models like MobileNet and EfficientNet, which may offer better performance or faster computation. More complex architectures such as Vision Transformer (ViT) could also be investigated. Additionally, the dataset could support research on advanced machine learning topics like anomaly detection, zero-shot learning, few-shot learning, self-supervised learning, continual learning, and open-set recognition.

Authorship Contribution Statement

Rahmat Ramadhani: Responsible for writing the first draft, collecting data, conducting experiments, and analyzing the results.

Gusti Ahmad Fanshuri Alfarisy: Contributed to conceptualizing the research, refining the research questions and objectives, revising the experimental design, analyzing the results, and revising the manuscript.

Boby Mugi Pratama: Contributed to analyzing the results, revising the manuscript, and providing additional insights for the discussion.

All authors have reviewed and approved the final version of the manuscript for submission.

Declaration of Conflicting Interests

The authors declare that there are no competing interests that could have influenced the work of our study.

References

[1] T. D. Siboro, “Manfaat Keanekaragaman Hayati Terhadap Lingkungan,” JURNAL ILMIAH SIMANTEK ISSN. 2550-0414, vol. 3, Feb. 2019.

[2] A. Antonelli *et al.*, *State of the World's Plants and Fungi 2023*. 2023. doi: 10.34885/wnwn-6s63.

[3] D. W. Purnomo, M. Siregar, J. R. Witono, and D. Usmadi, “Rencana 10 Tahun (2020-2030) Pengembangan Kebun Raya di Indonesia,” 2020.

[4] J. Wäldchen, M. Rzanny, M. Seeland, and P. Mäder, “Automated plant species identification—Trends and future directions,” Apr. 01, 2018, *Public Library of Science*. doi: 10.1371/journal.pcbi.1005993.

[5] A. Lawrence and W. Hawthorne, “Plant identification: Creating user-friendly field guides for biodiversity management,” *Plant Identification: Creating User-Friendly Field Guides for Biodiversity Management*, pp. 1–276, Oct. 2013, doi: 10.4324/9781849772198.

[6] L. Alzubaidi *et al.*, “Review of deep learning: concepts, CNN architectures, challenges, applications, future directions,” *J Big Data*, vol. 8, no. 1, Dec. 2021, doi: 10.1186/s40537-021-00444-8.

[7] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” 2015. [Online]. Available: http://image-net.org/challenges/LSVRC/2015/

[8] A. Atique, S. Karim, S. Shahid, and Z. Alamgir, “Identification of Plant Species through Leaf Vein Morphometric and Deep Learning,” *Pak J Bot*, vol. 54, no. 6, pp. 2195–2202, 2022, doi: 10.30848/PJB2022-6(38).

[9] R. Pujiati and N. Rochmawati, “Identifikasi Citra Daun Tanaman Herbal Menggunakan Metode Convolutional Neural Network (CNN),” *Journal of Informatics and Computer Science*, vol. 03, 2022.

[10] Y. Sun, Y. Liu, G. Wang, and H. Zhang, “Deep Learning for Plant Identification in Natural Environment,” *Comput Intell Neurosci*, vol. 2017, 2017, doi: 10.1155/2017/7361042.

[11] V. Bodhwani, D. P. Acharjya, and U. Bodhwani, “Deep residual networks for plant identification,” in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 186–194. doi: 10.1016/j.procs.2019.05.042.

[12] Y. Gong, G. Liu, Y. Xue, R. Li, and L. Meng, "A survey on dataset quality in machine learning," *Inf Softw Technol*, vol. 162, p. 107268, Oct. 2023, doi: 10.1016/j.infsof.2023.107268.

[13] B. Falahkhi, E. F. Achmal, M. Rizaldi, R. Rizki, and N. Yudistira, "Perbandingan Model AlexNet dan ResNet dalam Klasifikasi Citra Bunga Memanfaatkan Transfer Learning Comparison of AlexNet and ResNet Models in Flower Image Classification Utilizing Transfer Learning," 2022, [Online]. Available: http://journal.ipb.ac.id/index.php/jika

[14] M. S. I. Musyaffa, N. Yudistira, M. A. Rahman, and J. Batoro, "IndoHerb: Indonesia Medicinal Plants Recognition using Transfer Learning and Deep Learning," Aug. 2023, [Online]. Available: http://arxiv.org/abs/2308.01604

[15] T. Nguyen and V. Truong Hoang, "VNPlant-200 – A Public and Large-Scale of Vietnamese Medicinal Plant Images Dataset," 2021, pp. 406–411. doi: 10.1007/978-3-030-49264-9_37.

[16] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "RandAugment: Practical automated data augmentation with a reduced search space," Sep. 2019, [Online]. Available: http://arxiv.org/abs/1909.13719

[17] K. He, X. Zhang, S. Ren, and J. Sun, "Identity Mappings in Deep Residual Networks," Mar. 2016, [Online]. Available: http://arxiv.org/abs/1603.05027

[18] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32*, Curran Associates, Inc., 2019, pp. 8024–8035.

