Jurnal Pertanian Presisi, 10 (1):79-97
https://doi.org/10.35760/jpp.2026.v10i1.262

COMPARATIVE PERFORMANCE AND GENERALIZATION ANALYSIS OF MOBILENETV1 AND MOBILENETV2 FOR RHIZOME SPICE CLASSIFICATION

Najmah Femalea[1], Guntur Eka Saputra[2*], Delianti[3], Hilmi A'ini Nurthoyibah[4]

[1]Informatics Study Program, Faculty of Industrial Technology, Gunadarma University. najmahfemalea15@gmail.com

[2]Informatics Study Program (North Penajam Paser Regency Campus), Faculty of Industrial Technology, Gunadarma University. guntur@staff.gunadarma.ac.id

[3]Information Systems Study Program (North Penajam Paser Regency Campus), Faculty of Computer Science and Information Technology, Gunadarma University. delianti93@gmail.com

[4]Agrotechnology Study Program, Faculty of Industrial Technology, Gunadarma University. hilmitoyibah@gmail.com

Diterima 01 Mei 2026; Disetujui 04 Juni 2026

ABSTRACT

Indonesia's rich biodiversity includes rhizome spices that are often difficult to distinguish manually due to their similar visual characteristics. This study developed and compared MobileNetV1 and MobileNetV2 for classifying four rhizome spice classes, namely ginger, turmeric, galangal, and aromatic ginger, using a dataset of 1,120 images. MobileNetV1 achieved a training accuracy of 0.9611 at a learning rate of 0.001 in 1,522.65 seconds, whereas MobileNetV2 achieved a higher training accuracy of 0.9823 at a learning rate of 0.0002 in 1,444.20 seconds. While MobileNetV2 demonstrated superior classification performance and faster convergence, MobileNetV1 demonstrated stronger generalization capability, indicated by a smaller train-validation accuracy gap (1.21% vs. 1.80%) and more stable validation performance. Both no-dropout models achieved an accuracy, precision, recall, and F1-score of 0.9642 on the 112-image test set. The two best-performing models were deployed in a Streamlit-based web application. The results demonstrate that MobileNetV2 is preferable when maximizing predictive performance, whereas MobileNetV1 offers greater robustness for relatively small datasets. This study contributes to the development of practical AI-based tools for agricultural and spice-identification applications.

Keywords: Rhizome Spice Classification, Deep Learning, MobileNetV1, MobileNetV2, Convolutional Neural Networks

INTRODUCTION

Indonesia is widely recognized for its rich biodiversity, particularly in spices belonging to the Zingiberaceae family, such as ginger, turmeric, galangal, and aromatic ginger. These rhizome spices play important roles in Indonesian cuisine, traditional medicine, and the herbal industry. However, identifying rhizome spices remains challenging because many species possess highly similar visual characteristics, including color, texture, and shape. Misidentification frequently occurs among the general public, especially among younger generations and urban communities with limited agricultural experience (Batubara et al., 2020). Previous studies also reported that many students and respondents experienced difficulties distinguishing common rhizome species accurately (Hikmatulloh et al., 2017; Feberian & Fitriati, 2022). These challenges indicate the need for an automatic and reliable rhizome classification system.

Recent advances in deep learning, particularly Convolutional Neural Networks (CNNs), have demonstrated promising performance in image classification tasks. Several studies have applied CNN-based methods for rhizome spice classification. Indrani et al. (2020) employed SqueezeNet to classify nine rhizome species and reported 81% accuracy. Darmatasia and Syafar (2023) compared MobileNet, InceptionV3, and VGGNet, showing that MobileNet achieved competitive accuracy with lower computational cost. Suandana et al. (2024) used MobileNetV2 and achieved validation accuracy approaching 99.96% for 16 rhizome classes. Chen et al. (2025) use MobileNetV1 and MobileNetV2 architectures leveraging depthwise separable convolutions to reduce computational cost and parameter size without sacrificing accuracy.

Although previous studies reported high classification accuracy, several limitations remain. Most studies focused primarily on reporting model accuracy without analyzing model generalization capability, computational efficiency, or deployment feasibility in practical applications. In addition, many studies evaluated only a single architecture or used different datasets and experimental settings, making fair comparisons between lightweight CNN architectures difficult. Consequently, the relative advantages of MobileNetV1 and MobileNetV2 for rhizome spice classification remain unclear.

The comparison between MobileNetV1 and MobileNetV2 is important because both architectures are specifically designed for resource-constrained environments using depthwise separable convolutions, yet they differ significantly in architectural design. MobileNetV2 introduces inverted residual blocks and linear bottlenecks that potentially improve feature extraction efficiency and classification performance compared to MobileNetV1 (Sandler et al., 2018). However, the additional architectural complexity may also affect model stability and generalization when applied to relatively small agricultural datasets. Therefore, evaluating both architectures under identical experimental conditions is necessary to determine their effectiveness for practical rhizome classification tasks.

This study contributes in three aspects. First, it provides a direct comparison between MobileNetV1 and MobileNetV2 using the same dataset, preprocessing strategy, augmentation methods, and evaluation settings to ensure fair performance analysis. Second, this study evaluates not only classification accuracy but also generalization capability, training efficiency, and deployment suitability. Third, the best-performing model is implemented in a web-based application to demonstrate its practical applicability for real-world rhizome identification.

Therefore, this study aims to compare the performance of MobileNetV1 and MobileNetV2 in classifying four commonly used Indonesian rhizome spices, namely ginger, turmeric, galangal, and aromatic ginger. The study further analyzes their computational efficiency and generalization capability while implementing the best-performing model in a web-based classification system suitable for practical agricultural applications.

MATERIALS AND METHODS

Research Workflow

The overall research workflow is presented in Figure 1. The study began with dataset collection and manual annotation of four rhizome spice classes, namely ginger, turmeric, galangal, and aromatic ginger. The annotated dataset was then divided into training, validation, and testing subsets, followed by preprocessing and data augmentation to improve data quality and model robustness. Subsequently, transfer learning was applied using MobileNetV1 and MobileNetV2 architectures to develop classification models. The trained models were evaluated using accuracy, precision, recall, F1-score, and generalization gap metrics. Finally, the best-performing models were deployed in a Streamlit-based web application and further evaluated through functional and usability testing to assess their practical applicability.

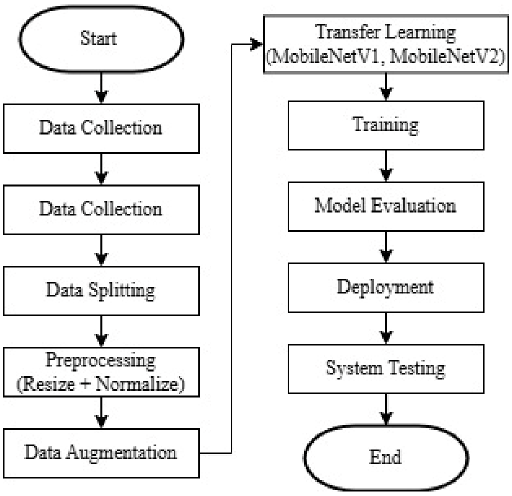


Figure 1. Research Workflow

Table 1. Primary Dataset

Ginger	Aromatic Ginger	Turmeric	Galangal
			

Dataset

The dataset used in this study consists of primary data collected using a 12 MP smartphone camera, resulting in images with a resolution of 3456 × 3456 pixels. The objects were placed on white paper at a distance of 10–15 cm and saved in JPG format (Alfarizi & Sela, 2024). The dataset includes four rhizome spice classes, namely ginger, turmeric, galangal, and aromatic ginger, with 280 images per class. All images were captured under varying lighting conditions and backgrounds to enhance the model's generalization capability, as shown in Table 1.

Data Preparation

Data preparation includes merging the images, manual annotation (each image placed into its corresponding class folder), dataset splitting, preprocessing, and augmentation. Manual annotation was conducted based on the morphological characteristics of each rhizome species, including shape, skin texture, and color (Sager et al., 2021). Since this study involved an interdisciplinary collaboration between information technology and agricultural fields, the labeling process was reviewed using agricultural knowledge related to rhizome identification to reduce labeling inconsistencies. The dataset was split into training (80%), validation (10%), and test (10%) sets. All images were resized to 224×224 pixels and normalized to the range [0,1] to improve training stability and convergence. Data augmentation was applied to the training set to reduce overfitting and enhance robustness to variations in viewpoint, lighting, and scale. The augmentation techniques used are summarized in Table 2.

Transfer Learning Framework

Transfer learning is a machine learning approach that utilizes knowledge acquired from a model previously trained on a source task and applies it to a related target task. By reusing previously learned feature representations, transfer learning can reduce training time and improve model performance, particularly when the target dataset is relatively limited. However, the effectiveness of transfer learning depends on the similarity between the source and target domains, as transferring knowledge between unrelated tasks may lead to performance degradation, commonly referred to as negative transfer (Hosna et al., 2022).

Table 2. Augmentation Techniques

Augmentation Technique	Description
Rotation range 20°	Randomly rotate images by up to ±20°
Brightness range (0.5, 1.5)	Scale image brightness between 50% and 150% of the original
Shear range 20°	Apply a shear (slant) transformation up to 20°
Width & height shift range 0.2	Translate images horizontally or vertically by up to ±20% of their size
Vertical flip	Randomly flip images top-to-bottom
Zoom range 0.3	Randomly scale images between 70% and 130% of the original size

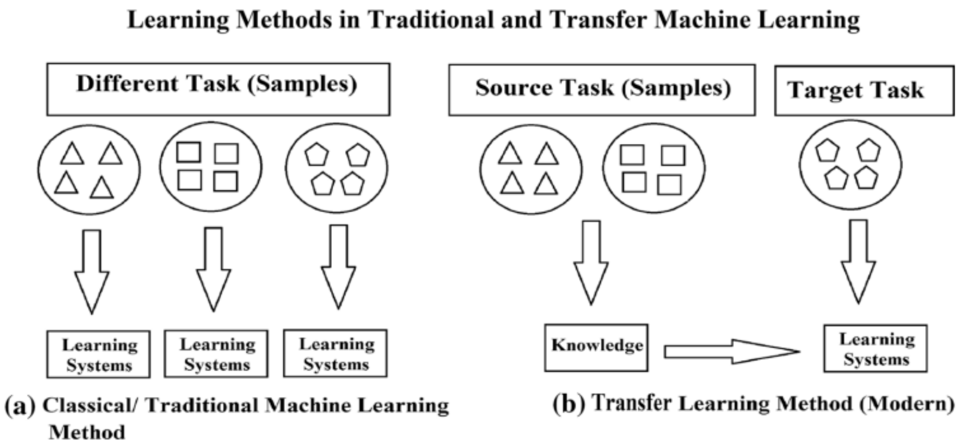


Figure 2. Traditional Machine Learning vs Transfer Learning

https://doi.org/10.35760/jpp.2026.v10i1.262

As illustrated in Figure 2, transfer learning enables a model to leverage previously acquired knowledge instead of learning all visual features from scratch. In this study, MobileNetV1 and MobileNetV2 pretrained on the ImageNet dataset were employed as the backbone architectures. The pretrained weights provide generalized visual representations learned from millions of images, including edges, textures, shapes, and object patterns, which can be effectively transferred to the rhizome spice classification task.

MobileNetV1

MobileNetV1 is a lightweight convolutional neural network architecture introduced by Howard et al. (2017) for mobile and embedded applications. The architecture employs depthwise separable convolutions to significantly reduce the number of parameters and computational cost while maintaining competitive classification performance. Due to its efficiency and compact model size, MobileNetV1 is widely adopted as a backbone architecture in transfer learning-based image classification tasks. As shown in Figure 3, MobileNetV1 pretrained on ImageNet was used as a frozen feature extraction backbone (trainable=False) to reduce computational cost and overfitting while preserving learned visual representations.

The extracted features were processed through a Global Average Pooling layer and fully connected layers with 128 and 64 neurons, followed by a Softmax output layer with four neurons for rhizome spice classification. Three variants were evaluated: without dropout, with dropout 0.2, and with dropout 0.3.

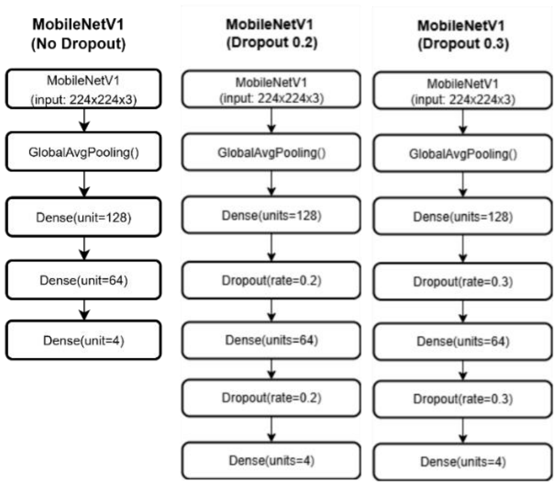


Figure 3. MobileNetV1 Architecture in this study

MobileNetV2

MobileNetV2 was proposed by Sandler et al. (2018) as an improved version of MobileNetV1. The architecture introduces inverted residual blocks and linear bottlenecks, enabling more efficient feature representation while maintaining low computational requirements.

These improvements allow MobileNetV2 to achieve higher accuracy than MobileNetV1 in many image classification tasks while preserving suitability for resource-constrained environments.

As shown in Figure 4, MobileNetV2 pretrained on ImageNet was used as a frozen feature extraction backbone, with the extracted features processed through a Global Average Pooling layer and fully connected layers containing 128 and 64 neurons. Three configurations were evaluated: without dropout, with dropout 0.2, and with dropout 0.3. The final Softmax layer generated classification probabilities for the four rhizome spice categories.

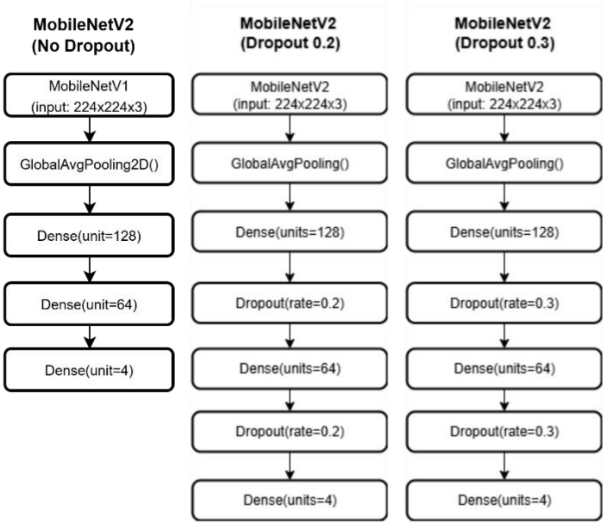


Figure 4. MobileNetV2 Architecture in this study

Evaluation Metrics

The performance of the proposed models was evaluated using a confusion matrix and several classification metrics, including accuracy, precision, recall, F1-score, and generalization gap. A confusion matrix provides a comprehensive representation of classification performance by comparing predicted labels with actual labels. For multi-class classification problems, the confusion matrix summarizes the numbers of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN), which are subsequently used to calculate various evaluation metrics (Kristine et al., 2020).

Accuracy

Accuracy measures the proportion of correctly classified samples relative to the total number of samples and provides an overall indication of model performance. It is calculated as:

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

Precision

Precision evaluates the proportion of correctly predicted positive instances among all instances predicted as positive. A high precision value indicates that the model produces fewer false positive predictions. Precision is calculated as:

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

Recall

Recall measures the ability of the model to identify all relevant positive instances. A higher recall value indicates that fewer positive samples are incorrectly classified as negative. Recall is calculated as:

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

F1-Score

F1-score is the harmonic mean of precision and recall, providing a balanced evaluation when both metrics are equally important. It is calculated as:

$$F1 - score = 2\, x \frac{precision\ x\ recall}{precision + recall} \quad (4)$$

Generalization Gap

Generalization gap measures the difference between model performance on the training dataset and its performance on unseen validation or testing data. This metric is useful for assessing whether a model generalizes well or exhibits signs of overfitting. In this study, the generalization gap was calculated as the difference between training accuracy and validation accuracy:

$$Generalization\ Gap = Training\ Accuracy - Validation\ Accuracy \quad (5)$$

A smaller generalization gap indicates more consistent performance across training and validation datasets, whereas a larger positive gap may indicate overfitting. Negative values occur when validation accuracy exceeds training accuracy, which can arise from regularization techniques such as dropout or the use of data augmentation during training.

RESULTS AND DISCUSSION

The method was implemented in Python 3.10.2 using libraries such as pandas, TensorFlow, and Keras. Development and experiments were performed in a Google Colaboratory environment with Visual Studio Code used as the IDE, while reproducibility tests were run on a Windows 10 machine (Intel® Core™ i5-1135G7 @ 2.40 GHz, 64-bit, 8 GB RAM).

Model Development

The pretrained MobileNetV1 and MobileNetV2 backbones were fine-tuned by attaching additional classification layers, as detailed in Table 3 Hyperparameter.

Each model was trained using the predefined combinations of hyperparameters, yielding six experimental scenarios as summarized in Table 4.

Model Training

Both MobileNetV1 and MobileNetV2 were trained on the training set and evaluated on the validation set using categorical cross-entropy loss, the Adam optimizer, and data augmentation to improve robustness. The training process was monitored through training and validation accuracy and loss to assess convergence and detect overfitting. MobileNetV1 achieved optimal performance with a learning rate of 0.0010, whereas MobileNetV2 performed best with a lower learning rate of 0.0002, reflecting its deeper and more complex architecture. The training results are summarized in Table 5.

Table 3. Hyperparameter

Hyperparameter	Values
Model_type	MobileNetV1, MobileNetV2
Use_dropout	True, False
Dropout_rate	0.2, 0.3
Dense_units1	128
Dense_units2	64
Optimizer	Adam
Learning_rate	0.0001, 0.0002, 0.0010
Class_mode	Categorical
Number of epoch	15
Batch size	32

Table 4. Training Scenario

Scenario	Model	Dropout	Fully connected layer	Epoch
1	MobileNetV1	(0.3)	(128, 64)	15
2	MobileNetV1	(0.2)	(128,64)	15
3	MobileNetV1	-	(128,64)	15
4	MobileNetV2	(0.3)	(128,64)	15
5	MobileNetV2	(0.2)	(128,64)	15
6	MobileNetV2	-	(128,64)	15

Table 5. Training Results

Model	Accuracy	Loss	Val Acc	Val Loss	Learning Rate	Training duration (seconds)
MobileNetV1 dropout 0.3	0.9246	0.2172	0.9554	0.1213	0.0002	1515.88
MobileNetV1 dropout 0.2	0.9472	0.1595	0.9643	0.1030	0.0010	1460.34
MobileNetV1 no dropout	0.9611	0.1164	0.9732	0.0962	0.0010	1522.65
MobileNetV2 dropout 0.3	0.9266	0.2335	0.9732	0.1145	0.0010	1501.08
MobileNetV2 dropout 0.2	0.9607	0.1263	0.8750	0.2481	0.0001	1452.30
MobileNetV2 no dropout	0.9823	0.0683	0.9643	0.0699	0.0002	1444.20

The experimental results show that MobileNetV2 without dropout achieved higher training accuracy (0.9823) and shorter training time than MobileNetV1 without dropout (0.9611). However, MobileNetV1 demonstrated more stable performance, making it potentially more reliable for deployment in limited-data scenarios. Additionally, the inclusion of dropout did not improve performance and generally reduced training efficiency compared to the original architectures.

These results also indicate that adding dropout did not improve performance and in fact reduced training efficiency compared with the original architectures. Thus, explicit regularization such as dropout is not always necessary to prevent overfitting. In this study, data augmentation which increases the diversity of training samples served as a more effective regularizer, making dropout less essential (Hernández-García & König, 2018). This finding suggests that the applied data augmentation strategy already provided sufficient variability in the training data, reducing the need for additional regularization such as dropout. In relatively small datasets, excessive regularization may hinder the model's ability to learn discriminative features effectively, leading to suboptimal performance.

Figure 5 shows that the no-dropout models (MobileNetV1 and MobileNetV2) achieved faster convergence, maintained training accuracy above 0.95, and exhibited more stable validation performance throughout training. In contrast, the dropout variants converged more slowly, achieved lower final accuracy, and showed greater fluctuations in validation accuracy, indicating less stable learning behavior. These patterns suggest that, while dropout can help reduce overfitting in some cases, it may slow convergence (Salehin & Kang, 2023). In this study, the architectures without dropout achieved the best combination of fast convergence and stable generalization. The difference between training and validation performance was further analyzed using the generalization gap to evaluate the model's ability to generalize to unseen data (Yang & Li, 2021). The result generalization gap for all models is presented in Table 6.

Based on the training and validation curves for the six models, the no-dropout variants generally exhibited smaller generalization gaps than the dropout variants. MobileNetV1 without dropout achieved the smallest positive generalization gap (1.21%), indicating a close correspondence between training and validation performance and suggesting stable learning behaviour.

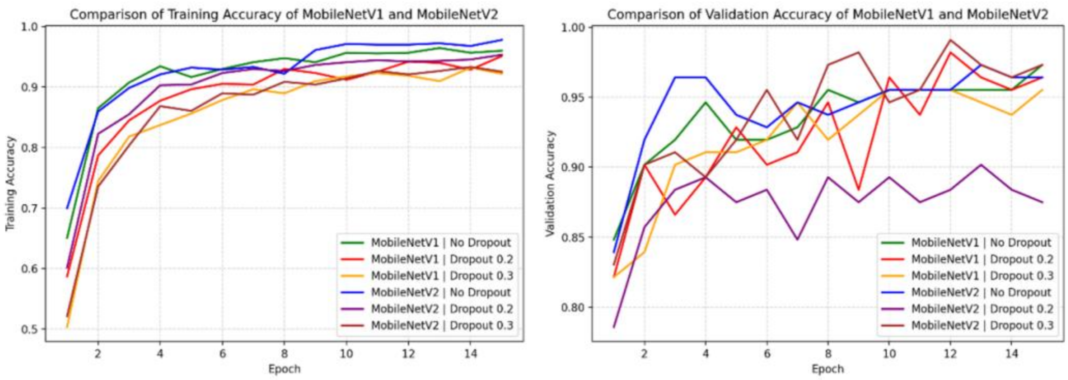


Figure 5. Training Accuracy vs Validation Accuracy graph

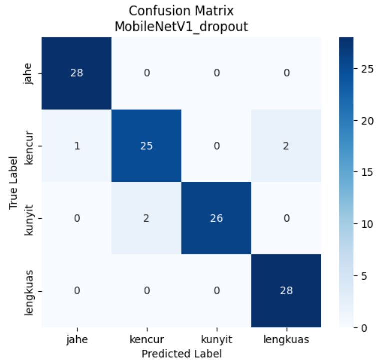
Table 6. Generalization gap for all models

Model	Generalization gap
MobileNetV1 dropout 0.3	3.08%
MobileNetV1 dropout 0.2	1.71%
MobileNetV1 no dropout	1.21%
MobileNetV2 dropout 0.3	4.66%
MobileNetV2 dropout 0.2	-8.57%
MobileNetV2 no dropout	-1.8%

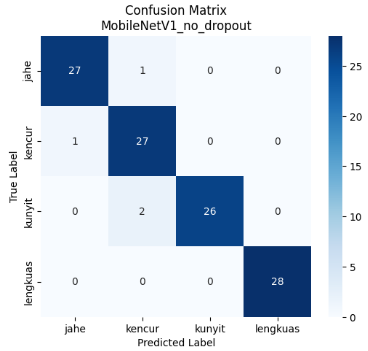
Interestingly, two MobileNetV2 configurations produced negative generalization gaps, namely MobileNetV2 with dropout 0.2 (-8.57%) and MobileNetV2 without dropout (-1.80%). Since the generalization gap is calculated as the difference between training accuracy and validation accuracy, negative values indicate that the validation accuracy exceeded the training accuracy. This phenomenon can occur when the training process involves data augmentation, which introduces additional variability and increases the difficulty of the training samples, while the validation data remain unchanged. Consequently, the model may achieve slightly lower accuracy on the augmented training set than on the validation set. For the dropout-based models, the negative gap can also be attributed to the regularization effect of dropout. During training, randomly deactivated neurons reduce the effective capacity of the network, making the learning task more challenging. In contrast, dropout is disabled during validation, allowing all learned features to contribute to prediction. As a result, validation accuracy may exceed training accuracy, producing a negative generalization gap. These findings suggest that the MobileNetV2 configurations generalized effectively to unseen validation data and did not exhibit evidence of overfitting under the evaluated experimental settings.

Model Evaluation

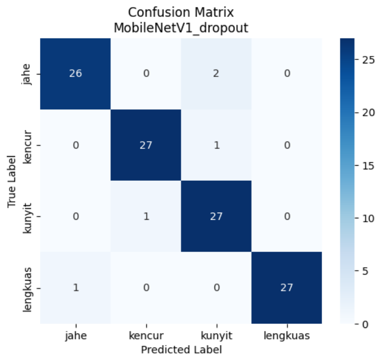
Model performance was evaluated using confusion matrices by comparing predicted labels with the true labels on the test set. MobileNetV1 with dropout (0.2 and 0.3) correctly classified 107 of 112 samples, while the no-dropout variant correctly classified 108 samples. For MobileNetV2, the evaluated variants correctly classified 101, 106, and 108 samples, respectively. MobileNetV1 exhibited a slightly smaller generalization gap (1.21%) than MobileNetV2 exhibited a negative generalization gap of 1.8%, suggesting marginally more consistent performance between training and validation data. However, the difference was relatively small, indicating that both architectures demonstrated similar generalization characteristics under the evaluated experimental conditions. Other performance metrics, such as precision, recall, F1-score can be calculated and the results are shown in Table 7.



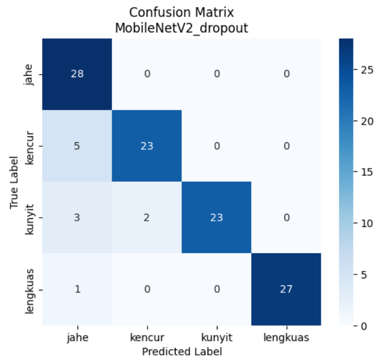
(a) MobileNetV1 dropout 0.3



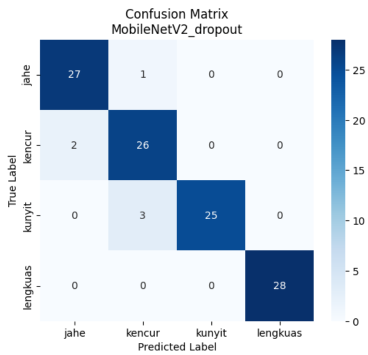
(b) MobileNetV1 no dropout



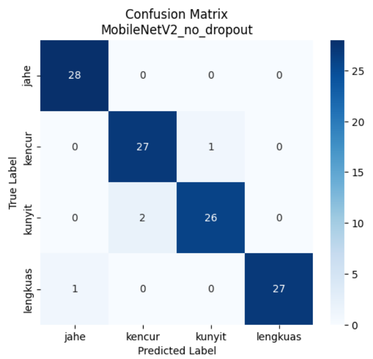
(c) MobileNetV1 dropout 0.2



(d) MobileNetV2 dropout 0.3



(e) MobileNetV2 dropout 0.2



(f) MobileNetV2 no dropout

Figure 6. Confusion matrix for all models

Table 7. Performance metrics for all models

Model	Precision	Recall	F1-score
MobileNetV1 dropout 0.3	95.53%	95.53%	95.53%
MobileNetV1 dropout 0.2	95.53%	95.53%	95.53%
MobileNetV1 no dropout	96.42%	96.42%	96.42%
MobileNetV2 dropout 0.3	90.17%	90.17%	90.17%
MobileNetV2 dropout 0.2	94.64%	94.64%	94.64%
MobileNetV2 no dropout	96.42%	96.42%	96.42%

The use of dropout did not significantly improve model performance. This may indicate that the applied data augmentation techniques already provided sufficient regularization, reducing the need for additional dropout layers. In some cases, excessive regularization can limit the model’s ability to learn important features. The table shows that both models achieved high performance in terms of precision, recall, and F1-score, with an overall score of 0.9642 (96.42%). This indicates that the models are capable of consistently distinguishing between different rhizome spice classes with minimal misclassification.

Computational Efficiency Analysis

In addition to classification performance, the computational efficiency of the proposed models was evaluated because MobileNet architectures are specifically designed for resource-constrained environments. Computational efficiency is an important consideration for real-world deployment, particularly in web-based and mobile applications where inference speed, memory consumption, and model size directly affect user experience and system responsiveness. Therefore, the best-performing configurations of MobileNetV1 and MobileNetV2 were further analyzed in terms of model size, number of parameters, inference time, and deployment response time.

Table 8 shows that MobileNetV2 was more computationally efficient than MobileNetV1, requiring fewer parameters (2.43M vs. 3.37M), a smaller model size (10.9 MB vs. 14.1 MB), and a faster inference time, with the no-dropout variant achieving the lowest average inference time of 76.02 ms per image compared to 80.57 ms for MobileNetV1. All model configurations generated predictions in less than 0.2 seconds after image upload, demonstrating suitability for real-time web-based applications. Although MobileNetV1 without dropout achieved a slightly faster response time (0.1354 s) than MobileNetV2 without dropout (0.1521 s). Overall, the results indicate that both architectures are suitable for practical deployment, with MobileNetV2 offering advantages in model compactness and inference efficiency while maintaining classification performance comparable to MobileNetV1.

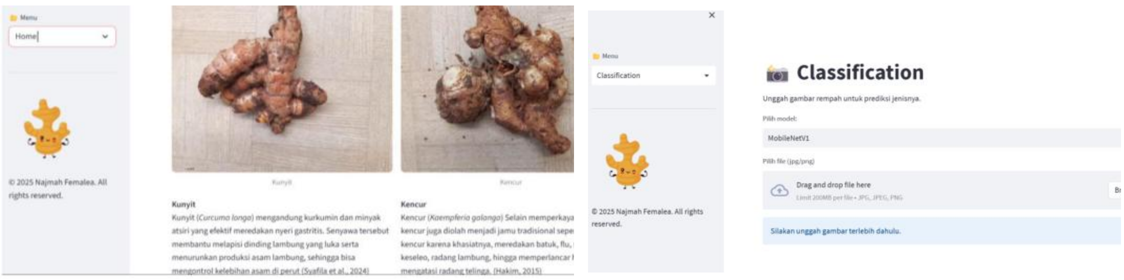
Model Deployment and System Development

The best-performing models were stored in HDF5 format, enabling the trained architectures and their corresponding weights to be reloaded efficiently for inference and deployment purposes (Pham et al., 2022). The best-performing models, MobileNetV1 without dropout and MobileNetV2 without dropout, were deployed in a Streamlit v1.22.0 web application featuring Home and Classification menus.

The Home menu provides an overview of the application and sample rhizome images, while the Classification menu enables real-time classification using both models. MobileNetV2 offers higher accuracy and faster convergence, whereas MobileNetV1 demonstrates a smaller generalization gap and slightly more stable learning behavior. Including both models allows users to compare their predictions and performance in practical deployment scenarios, as illustrated in Figure 7.

Table 8. Computational Efficiency

Metric	MobileNetV1			MobileNetV2		
	Dropout 0.3	Dropout 0.2	No Dropout	Dropout 0.3	Dropout 0.2	No Dropout
Model Size (MB)	14.1	14.1	14.1	10.9	10.9	10.9
Total Parameters	3,368,580	3,368,580	3,368,580	2,430,468	2,430,468	2,430,468
Average Inference Time (ms)	86.02	82.59	80.57	83.01	90.09	76.02
Average Response Time (s)	0.1589	0.1352	0.1354	0.1566	0.1540	0.1521



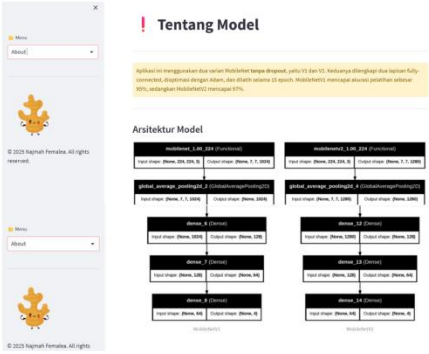


Figure **7**. Web application

Table 9. Results System Usability Scale

Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Sum	Score (sum*2.5)
2	3	4	3	3	2	4	3	4	2	30	75
2	4	4	3	4	3	3	3	4	3	33	82.5
3	4	4	4	4	4	3	4	4	4	38	95
3	3	3	3	4	3	4	3	3	3	32	80
3	3	2	3	4	2	4	3	2	2	28	70
4	4	4	2	4	1	4	4	4	4	35	87.5
3	4	4	4	4	2	4	4	4	4	37	92.5
3	3	3	3	3	2	3	3	3	1	27	67.5
3	3	4	3	4	3	3	3	4	2	32	80
4	4	4	4	4	4	4	4	4	4	40	100
Total SUS Score											830
Average SUS Score											83

System Testing and Evaluation

The best-performing models were successfully deployed in a Streamlit-based web application for real-time rhizome spice classification. Furthermore, usability was evaluated using the System Usability Scale (SUS), where 10 respondents rated 10 statements on a 5-point Likert scale to assess the application's usability, ease of use, and user satisfaction. The SUS score was calculated by subtracting 1 from the scores of odd-numbered items, subtracting the scores of even-numbered items from 5, summing all adjusted scores, and multiplying the total by 2.5 (Kortum & Bangor, 2020).

The system usability evaluation resulted in a SUS score of 83, which falls into Grade A and is categorized as Excellent. This result complements the functional testing results and indicates that the application is not only technically functional but also well accepted by users. Although the evaluation involved only a limited number of participants, the findings indicate the potential usability of the system for rhizome spice classification tasks (Yani et al., 2025). Furthermore, the proposed system shows potential to support agricultural activities, particularly in assisting users in identifying rhizome spices more efficiently and reducing misidentification.

CONCLUSION AND FUTURE WORK

This study compared the performance of MobileNetV1 and MobileNetV2 for rhizome spice image classification using transfer learning. The results showed that MobileNetV2 achieved the best overall classification performance and computational efficiency, characterized by higher accuracy, a smaller model size, fewer parameters, and faster inference time. In contrast, MobileNetV1 exhibited a smaller generalization gap, indicating slightly more stable learning behavior on the evaluated dataset. Both models achieved high classification performance on the test set and were successfully deployed in a Streamlit-based web application for real-time rhizome spice identification. These findings demonstrate that MobileNetV2 is the preferred choice when prioritizing predictive performance and computational efficiency, whereas MobileNetV1 may be considered when greater training stability is desired on relatively small datasets.

Future work may focus on expanding the dataset with additional rhizome spice classes and more diverse image conditions, as well as investigating other lightweight deep learning architectures. Further development may also include mobile-based deployment and real-time image acquisition to improve usability in practical agricultural applications.

REFERENCES

Alfarizi, A. Z., and Sela, E. I. 2024. Klasifikasi Rimpang Menggunakan Metode K-Nearest Neighbor dan Ekstraksi Ciri Gray Level Co-occurrence Matrix. *JURNAL FASILKOM (Teknologi Informasi dan Ilmu Komputer)*, Vol. 14(1), pp. 88-94. doi: https://doi.org/10.37859/jf.v14i1.6832

Batubara, N. P., Widiyanto, D., and Chamidah, N. 2020. Klasifikasi Rempah Rimpang Berdasarkan Ciri Warna Rgb dan Tekstur Glcm Menggunakan Algoritma Naive Bayes. *Informatik: Jurnal Ilmu Komputer*, Vol. 16(3), pp. 156-163. doi: https://doi.org/10.52958/iftk.v16i3.2196.

Chen, Y., Li, Y., Li, S., Lv, S., and Lin, F. 2025. DualCascadeTSF-MobileNetV2: A Lightweight Violence Behavior Recognition Model. *Applied Sciences*, Vol. 15(7) pp. 3862. doi: https://doi.org/10.3390/app15073862.

Darmatasia, D. and Syafar, A. M. 2023.

Implementasi Convolutional Neural Network Untuk Klasifikasi Tanaman Rimpang Secara Virtual Darmatasia. *INSTEK*, Vol. 8(1), pp. 122-131. doi: https://doi.org/10.24252/instek.v8i1.37255

Feberian, Y., and Fitriati, D. 2022. Klasifikasi Rimpang Menggunakan Convolutional Neural Network. *Journal of Informatics and Advanced Computing*, Vol. 3(1), pp. 10-14. doi: https://doi.org/10.35814/jiac.v3i1.3849

Hernández-García, A., and König, P. 2018. Do deep nets really need weight decay and dropout?. pp. 1-5. [Online]. Available: http://arxiv.org/abs/1802.07042

Hikmatulloh, E., Lasmanawati, E., and Setiawati, T. 2017. Manfaat Pengetahuan Bumbu Dan Rempah Pada Pengolahan Makanan Indonesia Siswa SMKN 9 Bandung. *Jurnal Media, Pendidikan, Gizi dan Kuliner*, Vol. 9 (1), pp. 42-50. doi: https://doi.org/10.17509/boga.v6i1.8844.

Hosna, A., Merry, E., Gyalmo, J., Alom, Z., Aung, Z., and Azim, M. A. 2022. Transfer learning: a friendly introduction. *Journal of Big Data*, Vol. 9 (102). doi: https://doi.org/10.1186/s40537-022-00652-w

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., and Adam, H. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv preprint arXiv*:1704.04861

Indrani, K. H., Khrisne, D. C., and Suyadnya, I. M. A. 2020. Android Based Application for Rhizome Medicinal Plant Recognition Using SqueezeNet. *Journal of Electrical, Electronics and Informatics*, 4(1), pp. 10-14. doi: https://doi.org/10.24843/JEEI.2020.v04.i01.p02

Kortum, P., and Bangor, A. 2020. Usability Ratings for Everyday Products Measured With the System Usability Scale. *International Journal of Human–Computer Interaction*, Vol. 36(2), pp. 192-200. doi: https://doi.org/10.1080/10447318.2019.1664705

Pham, Q. H., Nguyen, T. T., and Kim, C. H. 2022. Deep Learning Model Deployment for Web-Based Intelligent Systems: A Lightweight Framework Approach. *Applied Sciences*, Vol. 12(15), pp. 7612. doi: https://doi.org/10.3390/app12157612

Sager, C., Janiesch, C. and Zschech, P. 2021. A survey of image labelling for computer vision applications. *Journal of Business Analytics*, Vol. 4(2), pp. 91-110. doi: https://doi.org/10.1080/2573234X.2021.1908861

Salehin, I., and Kang, D.K. 2023. A Review on Dropout Regularization Approaches for Deep Neural Networks within the Scholarly Domain. *Electronics*, Vol. 12(14), pp. 1-23. doi: https://doi.org/10.3390/electronics12143106

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L. C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 4510-4520)

Suandana, I. N., Asriyanik, A., and Apriandari, W. 2024. Pemanfaatan CNN (Convolutional Neural Network) Dan Mobilenet V2 Dalam

Klasifikasi Rempah-Rempah Lokal Di Indonesia. *JATI (Jurnal Mahasiswa Teknik Informatika)*, Vol. 8(5), pp. 10109–10116. doi: https://doi.org/10.36040/jati.v8i5.10873

Yang, Y., and Li, C. 2021. Quantitative Analysis of the Generalization Ability of Deep Feedforward Neural Networks. *Journal of Intelligent & Fuzzy Systems*, Vol. 40(3), pp. 5719-5731. doi: https://doi.org/10.3233/JIFS-201679

Yani, R., Nazhifah, I., and Pradika, M. I. 2025. System Usability Scale in Information System Application Development Using Systematic Mapping Study. *Indonesian Journal of Applied Technology and Innovation Science (IJATIS)*, Vol. 2(2), pp. 130-140. doi: https://doi.org/10.57152/ijatis.v2i2.2275