Vol. 6 No. 2, 2025, pp. 154-164 **Jurnal Pepadun**

Implementation of Swin Transformer for Web-Based Identification of Rhizome Shaped Spices Using Streamlit Framework

***[1]Muhammad Arkan Nibrastama, [2]Didik Kurniawan, [3]Admi Syarif, and [4]Rico Andrian**

[1,2,3,4]Department of Computer Science, Universitas Lampung,
Jl. Prof. Dr. Ir. Sumantri Brojonegoro No.1, Lampung, Indonesia 35145
[2]Department Computer Science and Electronics, Universitas Gadjah Mada, Bulaksumur, Yogyakarta, Indonesia 55281
e-mail: *[1]**markan.nibrastama21@students.unila.ac.id**, [2]didik.kurniawan@fmipa.unila.ac.id,
[3]admi.syarif@fmipa.unila.ac.id, [4]rico.andrian@fmipa.unila.ac.id

Abstract *— Indonesia is a country rich in natural resources, particularly in various types of spices. Among these, rhizome-shaped spices such as ginger, turmeric, galangal, and aromatic ginger often present classification challenges due to their similar visual appearances. This study aims to address this issue by developing a web-based identification system utilizing the Swin Transformer—an advanced Vision Transformer architecture known for its effectiveness in image classification tasks. The Swin Transformer model demonstrated superior performance, achieving an accuracy of 99.11%, precision of 98.24%, recall of 98.21%, and F1-score of 98.21%. These results significantly outperform the Xception convolutional neural network (CNN) model, which was previously considered state-of-the-art, with 95.00% accuracy, 90.14% precision, 90.00% recall, and 90.01% F1-score. To ensure practical usability, the final Swin Transformer model was deployed as a web application using the Streamlit framework, allowing users to classify rhizome spices through image uploads. These findings highlight the effectiveness of Swin Transformer for practical image-based spice classification.*

Keywords: *Vision Transformer; Swin Transformer; Rhizome Spices; Streamlit.*

1. INTRODUCTION

Indonesia is a country with abundant natural resources, one of which is spices. These spices, which can be derived from flowers, leaves, seeds, stems, rhizomes, and roots, have long been known as high-value commodities not only for culinary purposes but also for their medicinal properties, particularly in maintaining the immune system [1][2]. Among the most common spices are rhizomes like ginger (*Zingiber officinale*), turmeric (*Curcuma longa*), aromatic ginger (*Kaempferia galanga)*, and galangal (*Alpinia galanga*). However, the visual similarities in shape, color, and texture make it difficult for many people to distinguish between them [1]. This difficulty is supported by a survey conducted by previous research [3], where only 31% of 100 respondents could correctly identify more than three out of five native Indonesian spices, indicating a low level of public knowledge.

To address this classification challenge, Machine Learning and Deep Learning techniques have been widely applied. These methods can automatically process data to find specific patterns and predict outcomes based on learned information [4][5]. Several studies have explored the identification of spices and herbs using these approaches. For instance, Riska and Farokhah [6] achieved 73% accuracy in classifying rhizomes using the K-Nearest Neighbors (K-NN) method. More advanced studies employing Convolutional Neural Network (CNN) architectures have shown significantly better results, with accuracies of 93% and 90% in herbal and spice identification [2][7]. Other implementations of CNN variants, such as VGGNet and VGG16, achieved accuracies of 70% and 85%, respectively [1][8].

While CNNs have become a common approach, recent advancements in deep learning have introduced the Transformer architecture, originally designed for Natural Language Processing (NLP), to the field of computer vision. The Vision Transformer (ViT) model works by dividing an image into patches and processing them as a sequence of tokens [9]. To address the computational intensity and limitations in capturing fine-grained details found in the original ViT, the Swin Transformer was introduced as an improved hierarchical vision architecture [10]. This architecture utilizes a hierarchical feature mapping approach and a novel shifted windows technique, demonstrating high efficiency and achieving 83.7% accuracy on the ImageNet-1K dataset. The Swin Transformer has also shown competitive or even superior performance compared to CNNs in other specialized domains, such as detecting Diabetic Retinopathy with a higher AUC score of 95.7% [11].

Despite its proven potential, the application of the Swin Transformer for identifying rhizome-type spices remains an unexplored area. Therefore, this study aims to address this research gap by implementing the Swin Transformer architecture for the image-based identification of rhizomes. The main objectives are to: (1) develop a model using a pre-trained Swin Transformer to classify images of ginger, turmeric, aromatic ginger, and galangal; (2) evaluate the model's accuracy; and (3) develop a web-based application using the Streamlit framework to provide a practical tool for spice identification. This research is expected to contribute a potentially more accurate model than existing CNN-based methods for this specific task.

2. RESEARCH METHOD

This research was conducted through a systematic workflow encompassing several key stages: dataset collection, preprocessing, model development, web application development, model and web performance evaluation. The overall research workflow is illustrated in Figure 1.

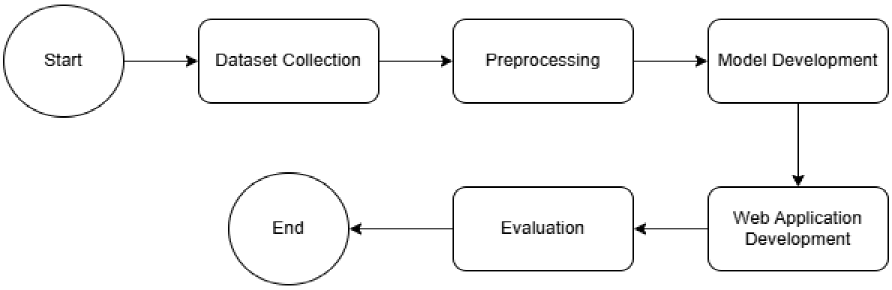

Figure 1. Research workflow.

2.1. Dataset Collection

The initial stage involved the acquisition of the dataset used for this study. A total of 1,400 images of rhizome-type spices were collected, ensuring a balanced distribution across four distinct classes: ginger, turmeric, aromatic ginger, and galangal. Each class contained 350 images.

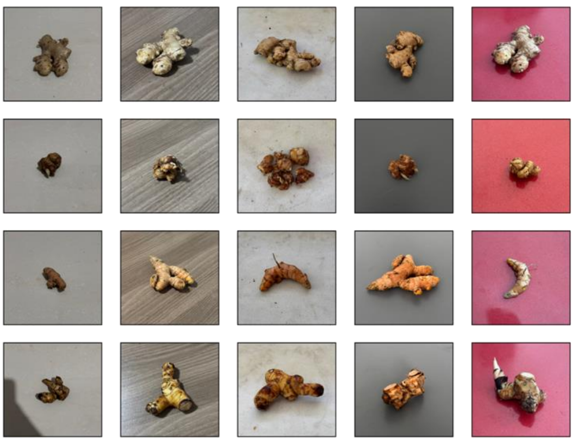

Figure 2. Example of dataset from each class.

The dataset was aggregated from two primary sources: a publicly available collection from the Roboflow platform and original photographs captured specifically for this research using a mobile phone camera. The example of the dataset from each class is illustrated in Figure 2.

2.2. Data Preprocessing

To prepare the raw data for model training, a multi-step preprocessing workflow was implemented. Data preprocessing is a vital stage for transforming raw data into a clean, suitable format to enhance data quality and facilitate more effective pattern recognition by the model [12]. First, datasets from both sources were merged into a single directory. A Python script utilizing the OS, Shutil, and PIL libraries was used to automate this process, ensuring all images were converted to a uniform JPEG format and systematically renamed to prevent file duplication. Following the merge, the dataset was split into training, validation, and testing subsets with a 60:20:20 ratio. This division, which is a common practice to ensure robust evaluation and prevent overfitting, resulted in 840 images for training, 280 for validation, and 280 for testing [13].

Table 1. Augmentation techniques.

No	Augmentation Type	Value
1	Random Horizontal Flip	
2	Random Vertical Flip	
3	Random Rotation	degrees = (0,180)
4	Random Perspective	distortion_scale= 0.3, p = 0.5
5	Resize	size = (224, 224)
6	ToTensor	
7	Normalize	mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225]

Figure 3. Example of augmented dataset.

Data augmentation was then applied exclusively to the training set to enhance its diversity and improve the model's generalization capabilities. The augmentation techniques included random horizontal flips, random vertical flips, random rotations, and random perspective transformations are listed on Table 1. Finally, all images across the three subsets were resized to a uniform dimension of 224 × 224 pixels. This step is crucial as it standardizes the input size to match the specific input requirement of the Swin Transformer architecture. The example result of augmented dataset is illustrated in Figure 3.

2.3. Model Development

The model development process centered on a transfer learning approach using the Swin Transformer architecture to enhance performance and reduce training time [14]. The process began by initializing the model and loading weights from a pre-trained Swin Transformer, previously trained on the large-scale ImageNet-1K dataset [10]. Following this, parameter freezing was selectively applied, if a layer's parameters were successfully transferred from the pre-trained model, they were frozen to preserve their robust, learned feature representations. However, if a layer's parameters were not transferred, such as in a newly created layer, they

 | doi: 10.23960/pepadun.v6i3.276

were left unfrozen and trainable. Specifically, the model's architecture was adapted by replacing the final classification layer with a new fully-connected layer configured with four neurons to match the rhizome classes in this study. This modified model was then trained for several hyperparameters that were listed on Table 2. Upon completion of training, the final fine-tuned model was saved as a PyTorch Tensors (.pt) file for subsequent evaluation and deployment.

Table 2. Hyperparameter for training model.

No	Augmentation Type	Value
1	Epoch	30
2	Learning Rate	0.0001
3	Batch Size	16
4	Optimizer	AdamW
5	Weight Decay	0.0001

2.4. Web Application Development

A user-facing web application was developed to provide a practical interface for the classification model, built using Streamlit, an open-source Python framework that simplifies the creation of interactive applications for machine learning projects without requiring traditional web development expertise [15]. The application allows a user to upload an image of a rhizome. The system then automatically preprocesses the uploaded image by resizing it to 224 × 224 pixels and converting it into a PyTorch tensor. This tensor is fed into the saved Swin Transformer model, which performs the classification. The predicted class (e.g., "Ginger," "Turmeric") is then displayed to the user on the web interface that is illustrated on Figure 4.

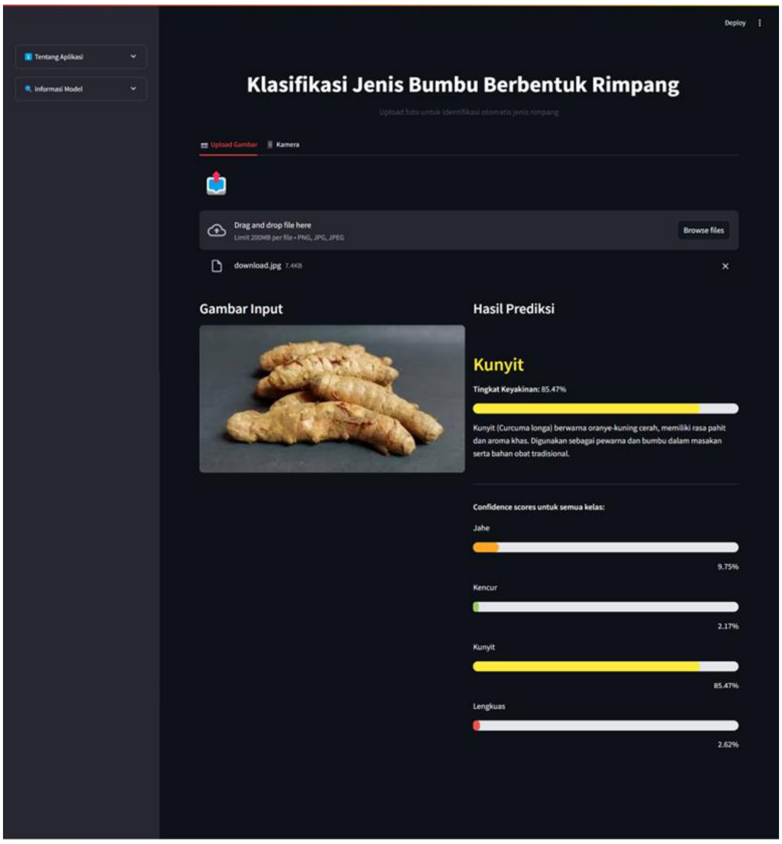

Figure 4. Web application interface.

2.5. Evaluation Methods

The performance of the developed system was evaluated through two distinct testing phases: model performance testing and web application functional testing.

2.5.1. Model Performance Evaluation

After training, all models were evaluated using the test set to assess classification performance on previously unseen data. The performance evaluation was conducted based on several training scenarios to identify the optimal model configuration. These scenarios involved systematically varying key parameters as follows:

i. Training with and without data augmentation.
ii. Applying different patch sizes and window sizes for the Swin Transformer.
iii. Applying various drop rates and drop path rates.

For each model resulting from these scenarios, performance was quantified using a confusion matrix [16] to calculate final test accuracy, precision, recall, and F1-score [17]. Furthermore, the model's ability to generalize and handle Out-of-Distribution (OOD) data was assessed by analyzing the Area Under the Receiver Operating Characteristic (AUROC) curve [18]. An optimal confidence threshold for OOD rejection was determined using the 'Closest to (0,1) Criteria' [19]. This method identifies the threshold corresponding to the point on the ROC curve with the minimum Euclidean distance in Equation (1) to the ideal classifier point (FPR=0, TPR=1), thus providing the best balance between sensitivity and specificity.

$$\mathrm{D} = \sqrt{(1-\mathrm{TPR})^2 + \mathrm{FPR}^2} \quad (1)$$

In addition to these quantitative metrics, visual analyses of the training and validation performance trends (learning curves) were conducted to assess model convergence and detect potential overfitting. The objective of this multi-scenario evaluation was to identify the combination of techniques and hyperparameters that yielded the highest and most stable classification performance. Accuracy measures the proportion of total correct predictions among all cases evaluated [17].

$$\text{Accuracy} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}} \times 100\% \quad (2)$$

Precision measures the proportion of true positive predictions among all instances predicted as positive, indicating the model's exactness [16]. Recall, or sensitivity, measures the proportion of true positive predictions among all actual positive instances, indicating the model's completeness [16]. The F1-Score is the harmonic mean of Precision and Recall, providing a single metric that balances both concerns, which is particularly useful for imbalanced datasets [16].

$$\text{Precision} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}} \times 100\% \quad (3)$$

$$\text{Recall} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} \times 100\% \quad (4)$$

$$\text{F1-Score} = 2 \times \frac{\text{Precision x Recall}}{\text{Precision + Recall}} \times 100\% \quad (5)$$

2.5.2. Web Application Functional Testing

The functionality of the Streamlit web application was verified using black box testing. This method assesses the system's input-output behavior without considering its internal structure [20]. A set of predefined test scenarios, as detailed in Table 3, was executed to ensure the application behaves as expected under various conditions, including valid input, invalid file formats, and empty submissions.

 | doi: 10.23960/pepadun.v6i3.276

Table 3. Web application black-box testing.

ID Test	Test Scenario	Input	Expected Output	Status
1	Upload a valid format file	Rhizome image (jpg/png)	Image is displayed, and a prediction appears.	Valid
2	Upload an invalid format file	PDF or TXT file	An error message is displayed.	Valid
3	No file uploaded, predict button clicked	No file uploaded	Message appears: "No image selected!"	Valid
4	Upload a non-rhizome image	Image of a non-rhizome object	Message appears: "Image is not part of a class"	Valid

3. RESULT AND DISCUSSION

This section presents the results of a series of experiments conducted to determine the optimal configuration for the Swin Transformer model. The performance of various hyperparameter settings is analyzed, followed by a comparative evaluation against a baseline CNN architecture and a final test on real-world data via the web application.

3.1. Comparison of Patch Size and Window Size Configurations

An initial experiment was conducted to evaluate the impact of different patch size and window size configurations on the Swin Transformer's performance. As shown in Table 4, three viable configurations were tested. The results clearly indicate that the model with a patch size of 4 and a window size of 7 achieved the best performance across all metrics, yielding an accuracy of 99.11% and an F1-Score of 98.21%. The configuration with a smaller patch size (2×7) performed significantly worse (61.71% F1-Score), suggesting that excessively small patches failed to capture sufficiently distinct visual features, leading to high confusion between visually similar classes like ginger, turmeric, and galangal.

Table 4. Performance comparison of patch size and windows size configurations.

Patch Size	Windows Size	Accuracy	Precision	Recall	F1-Score
2	7	80.89%	63.31%	61.79%	61.71%
4	7	99.11%	98.24%	98.21%	98.21%
4	14	96.61%	93.48%	93.21%	93.23%

Conversely, the model with a larger window size (4×14) also showed a slight performance degradation (93.23% F1-Score). This implies that an overly large attention window may dilute the model's focus on crucial local features necessary for differentiating the rhizomes. Therefore, the 4×7 configuration provided the optimal balance between capturing detailed local features and maintaining a broad enough receptive field for this specific task.

3.2. Comparison of Drop Rate and Drop Path Rate Configurations

To assess the impact of regularization, a series of experiments were performed by varying the drop rate and drop path rate. The results, summarized in Table 5, show that increasing these rates generally tended to decrease model performance, with the highest rates (e.g., drop rate 0.2, drop path rate 0.3) causing a significant drop in F1-Score to 49.86%. While several configurations with a drop rate of 0 achieved high metric scores, a

 | doi: 10.23960/pepadun.v6i3.276

deeper analysis was required to evaluate their generalization capability and risk of overfitting. By analyzing the Area Under the Receiver Operating Characteristic (AUROC) for Out-of-Distribution (OOD) detection, it was determined that the configuration with drop rate 0 and drop path rate 0.1 was the most robust.

Table 5. Performance comparison of drop rate and drop path rate configurations.

Drop Rate	Drop Path Rate	Accuracy	Precision	Recall	F1-Score
0	0.1	99.11%	98.24%	98.21%	98.21%
0	0.2	99.11%	98.24%	98.21%	98.21%
0	0.3	98.75%	97.54%	97.75%	97.75%
0.1	0.1	96.07%	92.63%	92.14%	92.11%
0.1	0.2	95.36%	91.75%	90.71%	90.75%
0.1	0.3	94.46%	90.42%	88.92%	88.90%
0.2	0.1	81.61%	72.73%	63.22%	62.62%
0.2	0.2	86.07%	73.07%	72.14%	71.49%
0.2	0.3	77.32%	56.64%	54.64%	49.86%

This model not only achieved a high F1-Score of 98.21% but also yielded the highest AUROC of 95.09%. Other top-performing configurations showed slightly lower AUROC values (92.71% for drop path rate 0.2 and 90.88% for drop path rate 0.3), indicating a marginally weaker ability to generalize to unseen data. This confirms that the model with drop rate 0 and drop path rate 0.1 provided the best trade-off between high classification accuracy and strong generalization.

3.3. Comparison With and Without Data Augmentation

The effect of data augmentation on model performance was evaluated. As detailed in Table 6, the results show that the model trained without data augmentation consistently outperformed the model trained with it. The non-augmented model achieved an accuracy of 99.11% and an F1-Score of 98.21%, compared to the augmented model's 97.86% accuracy and 95.71% F1-Score.

Table 6. Performance comparison with and without augmentation.

Augmentation	Accuracy	Precision	Recall	F1-Score
No	99.11%	98.24%	98.21%	98.21%
Yes	97.86%	95.82%	95.72%	95.71%

An analysis of the misclassifications revealed that the augmented model made more errors, particularly between classes with high visual similarity. This counter-intuitive finding suggests that for this specific dataset, where subtle texture and shape differences are key, the transformations applied during augmentation may have introduced noise or distortions that obscured these distinguishing features, thereby hindering the learning process rather than aiding it.

3.4. Comparison of Transformer and CNN Architectures

A key objective of this study was to compare the performance of the Vision Transformer architecture against a state-of-the-art CNN. The best-performing Swin Transformer model was compared with an Xception model, a high-performing CNN from previous research. The results in Table 7 conclusively demonstrate the superiority of the Swin Transformer for this task.

 | doi: 10.23960/pepadun.v6i3.276

Table 7. Performance comparison transformer and CNN architecture.

Architecture	Accuracy	Precision	Recall	F1-Score
Swin Transformer	99.11%	98.24%	98.21%	98.21%
Xception	95.00%	90.14%	90.00%	90.01%

The Swin Transformer achieved an accuracy of 99.11%, significantly higher than Xception's 95.00%. This advantage was consistent across all other metrics, with the Swin Transformer yielding an F1-Score of 98.21% compared to Xception's 90.01%. The performance gap is explained by a substantially lower number of misclassifications for the Swin Transformer, indicating its enhanced ability to capture and differentiate the complex visual features of the rhizomes more effectively than the CNN-based architecture.

3.5. Prediction Results with Internet Data via the Web Application

To determine the model's optimal decision threshold, an analysis of the Receiver Operating Characteristic (ROC) curve was conducted. As shown in Figure 5, the curve plots the true positive rate (sensitivity) against the false positive rate, enabling an assessment of the trade-off between correctly identifying in-distribution samples and avoiding false positives.

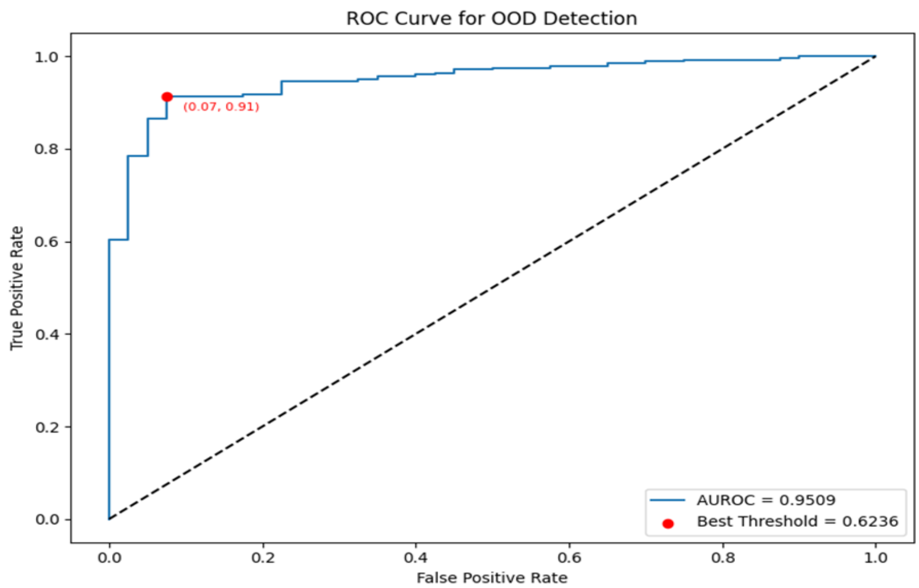

Figure 5. Optimal threshold from ROC curve.

The optimal threshold was selected using the 'Closest to (0,1) Criteria' method, which identifies the point on the ROC curve nearest to the top-left corner—representing the best balance of sensitivity and specificity. This method yielded an optimal confidence threshold of 62.36%, which was used during inference to help the model effectively distinguish between in-distribution (ID) and out-of-distribution (OOD) inputs.

To evaluate the model's generalization in a real-world setting, the final model was tested using the deployed web application with new, unseen images sourced from the internet. As illustrated in Figure 6, the model successfully demonstrated its OOD detection capability by refraining from classifying visually similar but out-of-class rhizomes such as java turmeric, thanks to the previously determined confidence threshold. On the in-distribution test, the model achieved an overall accuracy of 93.75%, with per-class accuracies of 100% for galangal, 95% for ginger, and 90% for both aromatic ginger and turmeric. However, this performance reflects a slight drop compared to the 99.11% accuracy on the original test set.

 | doi: 10.23960/pepadun.v6i3.276

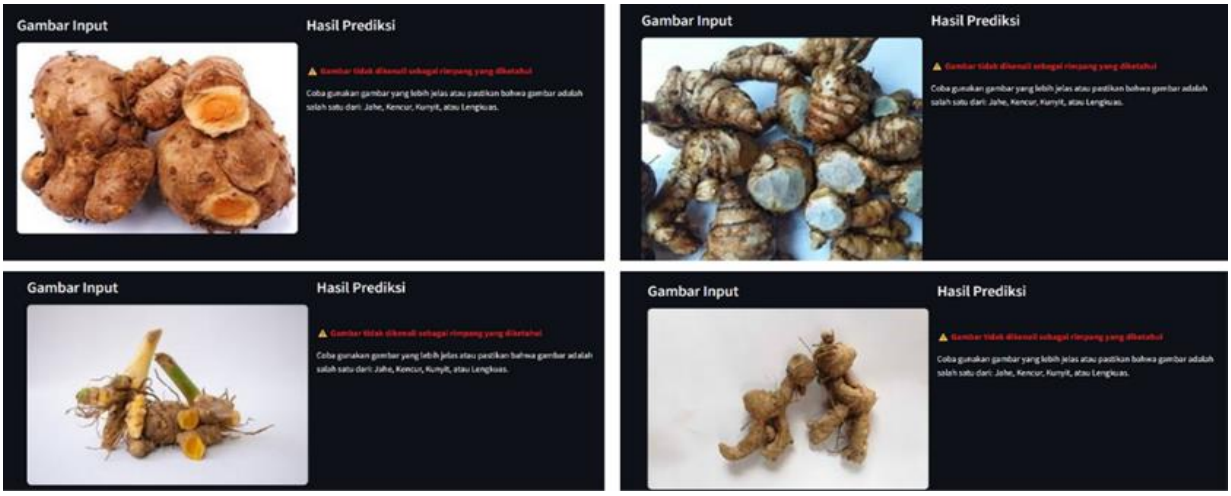

Figure 6. Example of successful Out-of-Distribution (OOD) data rejection.

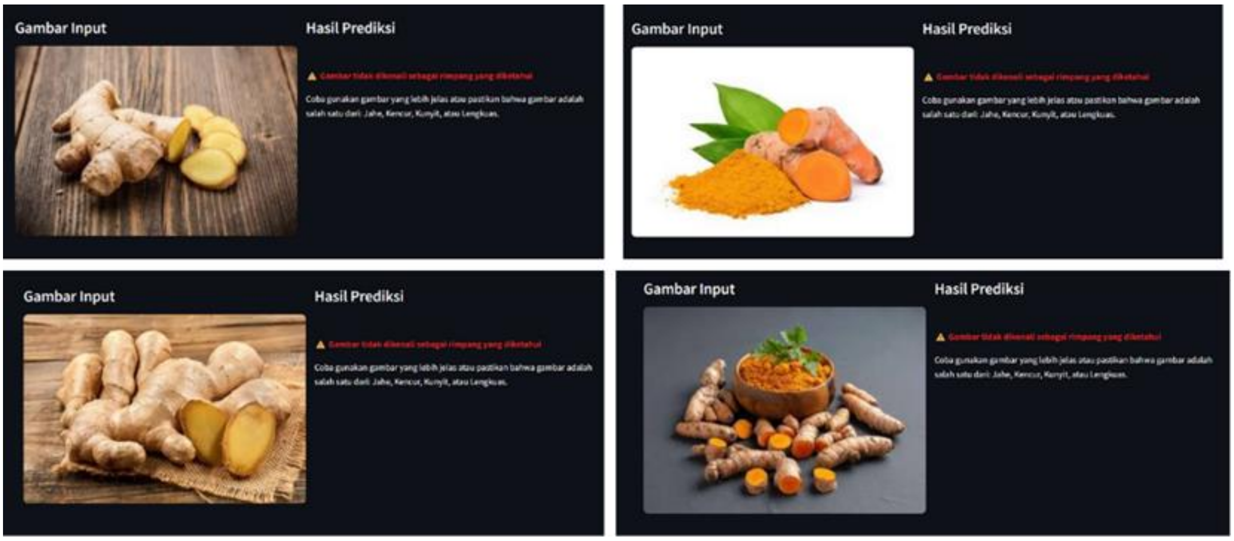

Figure 7. Example of In-Distribution (ID) data incorrectly rejected as OOD.

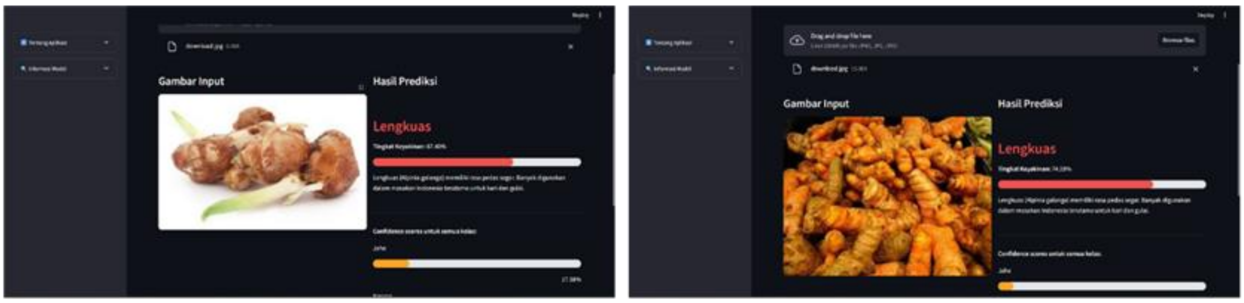

Figure 8. Example of In-Distribution (ID) data misclassification due to ambiguous features.

As shown in Figure 7, several misclassifications occurred due to domain shift, where significant differences in image characteristics—such as lighting conditions or visual edits—caused the model to incorrectly reject ID samples as OOD. Additionally, Figure 8 presents a case where an aromatic ginger image with a sprout was misclassified as galangal, likely due to the model's learned association of sprouts as a dominant feature of galangal during training. These results suggest that while the model is effective, its real-world performance could be further improved by incorporating a more visually diverse training dataset to enhance robustness against variations.

4. CONCLUSION

This research successfully demonstrated that the Swin Transformer architecture significantly outperforms a state-of-the-art CNN (Xception) in the task of classifying visually similar rhizome-type spices, achieving a peak accuracy of 99.11% on the test set. Through a series of comparative experiments, the optimal model configuration was identified, utilizing a patch size of 4, a window size of 7, a drop path rate of 0.1, and notably, no data augmentation, as augmentation was found to degrade performance for this specific dataset. Furthermore, the optimized model was successfully deployed into a functional web application, providing a practical interface for real-time prediction. However, evaluation using new data sourced from the internet revealed a performance drop to 93.75% accuracy, highlighting the model's limitations in generalizing to visual variations not present in the training set, despite its effectiveness in rejecting Out-of-Distribution (OOD) inputs. These findings confirm the potential of Vision Transformers for specialized agricultural image classification while also underscoring the critical need for more diverse training data to improve real-world robustness.

LITERATURE

[1] E. Tanuwijaya and A. Roseanne, "Modifikasi Arsitektur VGG16 untuk Klasifikasi Citra Digital Rempah-Rempah Indonesia," *MATRIK J. Manajemen, Tek. Inform. dan Rekayasa Komput.*, vol. 21, no. 1, pp. 189–196, Nov. 2021, doi: 10.30812/matrik.v21i1.1492.

[2] C. Nisa and F. Candra, "Klasifikasi Jenis Rempah-Rempah Menggunakan Algoritma Convolutional Neural Network," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no. 1, pp. 78–84, Dec. 2023, doi: 10.57152/malcom.v4i1.1018.

[3] A. E. Putra, M. F. Naufal, and V. R. Prasetyo, "Klasifikasi Jenis Rempah Menggunakan Convolutional Neural Network dan Transfer Learning," *J. Edukasi dan Penelit. Inform.*, vol. 9, no. 1, p. 12, Apr. 2023, doi: 10.26418/jp.v9i1.58186.

[4] A. Roihan, P. A. Sunarya, and A. S. Rafika, "Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper," *IJCIT (Indonesian J. Comput. Inf. Technol.*, vol. 5, no. 1, pp. 75–82, 2020, doi: 10.31294/ijcit.v5i1.7951.

[5] S. Ilahiyah and A. Nilogiri, "Implementasi Deep Learning Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan Convolutional Neural Network _ Ilahiyah _ JUSTINDO (Jurnal Sistem dan Teknologi Informasi Indonesia)," *JUSTINDO (Jurnal Sist. Teknol. Inf. Indones.*, vol. 3, no. 2, pp. 49–56, 2018.

[6] S. Y. Riska and L. Farokhah, "Klasifikasi Bumbu Dapur Indonesia Menggunakan Metode K-Nearest Neighbors (K-NN)," *SMATIKA J.*, vol. 11, no. 01, pp. 37–42, Jun. 2021, doi: 10.32664/smatika.v11i01.568.

[7] J. O. Carnagie, A. R. Prabowo, I. Istanto, E. P. Budiana, I. K. Singgih, I. Yaningsih, and F. Mikšík, "Technical review of supervised machine learning studies and potential implementation to identify herbal plant dataset," *Open Eng.*, vol. 13, no. 1, Feb. 2023, doi: 10.1515/eng-2022-0385.

[8] D. C. Khrisne and I. M. A. Suyadnya, "Indonesian Herbs and Spices Recognition using Smaller VGGNet-like Network," *in 2018 International Conference on Smart Green Technology in Electrical and Information Systems (ICSGTEIS)*, IEEE, Oct. 2018, pp. 221–224. doi: 10.1109/ICSGTEIS.2018.8709135.

[9] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *ICLR 2021 - 9th Int. Conf. Learn. Represent.*, Oct. 2020, [Online]. Available: http://arxiv.org/abs/2010.11929

[10] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," *2021 IEEE/CVF Int. Conf. Comput. Vis.*, pp. 9992–10002, Mar. 2021, doi: 10.1109/ICCV48922.2021.00986.

[11] O. Moutik, H. Sekkat, S. Tigani, A. Chehri, R. Saadane, T. A. Tchakoucht, and A. Paul, "Convolutional Neural Networks or Vision Transformers: Who Will Win the Race for Action Recognitions in Visual Data?," *Sensors*, vol. 23, no. 2, p. 734, Jan. 2023, doi: 10.3390/s23020734.

[12] Z. Song, S. Yang, and R. Zhang, "Does Preprocessing Help Training Over-parameterized Neural Networks?," *Adv. Neural Inf. Process. Syst.*, vol. 27, pp. 22890–22904, Oct. 2021, [Online]. Available: http://arxiv.org/abs/2110.04622

[13] Z. Ye, Y. Yang, X. Li, D. Cao, and D. Ouyang, "An Integrated Transfer Learning and Multitask Learning Approach for Pharmacokinetic Parameter Prediction," *Mol. Pharm.*, vol. 16, no. 2, pp. 533–541, 2019, doi: 10.1021/acs.molpharmaceut.8b00816.

[14] G. Morales, S. G. Huam, and J. Telles, Artificial Neural Networks and Machine Learning – ICANN 2018, vol. 11141, no. November. *in Lecture Notes in Computer Science*, vol. 11141. Cham: Springer International Publishing, 2018. doi: 10.1007/978-3-030-01424-7.

[15] I. L. Kharisma, D. A. Septiani, A. Fergina, and Kamdan, "Penerapan Algoritma Decision Tree untuk Ulasan Aplikasi Vidio di Google Play," *J. Nas. Teknol. dan Sist. Inf.*, vol. 9, no. 2, pp. 218–226, Sep. 2023, doi: 10.25077/teknosi.v9i2.2023.218-226.

[16] A. Arias-Duart, E. Mariotti, D. Garcia-Gasulla, and J. M. Alonso-Moral, "A Confusion Matrix for Evaluating Feature Attribution Methods," *in 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, Jun. 2023, pp. 3709–3714. doi: 10.1109/CVPRW59228.2023.00380.

[17] M. Bekkar, H. K. Djemaa, and T. A. Alitouche, "Evaluation Measures for Models Assessment over Imbalanced Data Sets," *J. Inf. Eng. Appl.*, vol. 3, no. 10, pp. 27–38, 2013, [Online]. Available: http://www.iiste.org/Journals/index.php/JIEA/article/view/7633

[18] A. M. Carrington, D. G. Manuel, P. W. Fieguth, T. Ramsay, V. Osmani, B. Wernly, C. Bennett, S. Hawken, O. Magwood, Y. Sheikh, M. McInnes, and A. Holzinger, "Deep ROC Analysis and AUC as Balanced Average Accuracy, for Improved Classifier Selection, Audit and Explanation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 1, pp. 329–341, 2023, doi: 10.1109/TPAMI.2022.3145392.

[19] Kanika, J. Singla, and Nikita, "Comparing ROC curve based thresholding methods in online transactions fraud detection system using deep learning," *Proc. - IEEE 2021 Int. Conf. Comput. Commun. Intell. Syst. ICCCIS 2021*, pp. 119–124, 2021, doi: 10.1109/ICCCIS51004.2021.9397167.

[20] D. Febiharsa, I. M. Sudana, and N. Hudallah, "Uji Fungsionalitas (Blackbox Testing) Sistem Informasi Lembaga Sertifikasi Profesi (SILSP) Batik dengan AppPerfect Web Test dan Uji Pengguna*," Joined J. Journal Informatics Educ.*, vol. 1, no. 2, p. 117, Dec. 2018, doi: 10.31331/joined.v1i2.752.