RESEARCH ARTICLE

Autoencoder Applications for Enhancing Spice Classification Performance Under Limited Dataset Conditions

Evan Tanuwijaya[1,*], **Andreas Lim**[1], **Nathanael Suryanto**[1], and **Joey Wiryawan**[1]

[1]Informatics, School of Information Technology, Universitas Ciputra Surabaya, Surabaya 60219, Indonesia

*Corresponding author: Evan Tanuwijaya, evan.tanuwijaya@ciputra.ac.id

Article Information

Article History:
Received: 18 April 2024
Revised: 12 November 2024
Accepted: 28 November 2024
Published: 19 October 2025

Keywords:
Autoencoder
Convolution Neural Network
Limited Dataset
Spices

Abstract

Indonesia is home to a vast array of spices, known for their health benefits and essential roles in culinary traditions. However, widespread knowledge of these spices remains limited. This study explores the use of autoencoder technology to address the challenges of classifying spices based on their intricate shapes and colors, especially when faced with the limitations of a small dataset. By carefully adjusting hyperparameters such as learning rate, optimization function, input size, and epochs, the autoencoder is optimized to extract distinctive features necessary for accurate classification. A comparison with convolutional neural networks (CNNs) using transfer learning shows that the autoencoder achieves similar accuracy levels, while consistently avoiding issues of overfitting or underfitting. With precision and recall rates nearing 0.97, the autoencoder demonstrates its ability to compress image data and identify key features, making it an effective solution for spice classification in constrained data environments.

1. Introduction

Indonesia is a country renowned for its vast and diverse range of spices, many of which are integral to both the health and culinary sectors. Spices (Nisa and Candra, 2023) such as ginger, turmeric, galangal, and cinnamon have been historically revered for their medicinal properties, serving as key components in traditional remedies and herbal treatments (Jiang, 2019; Li, 2006). In addition to their health benefits, these spices are fundamental to Indonesian cuisine, where they are used as primary ingredients to enhance the flavor, aroma, and preservation of food (Agrahar-Murugkar, 2020; Cardoso-Ugarte and Sosa-Morales, 2021; De-Montijo-Prieto et al., 2021; Mandal et al., 2022). The economic importance of Indonesia's spice industry cannot be understated, with many of these spices being exported to international markets, further highlighting their global demand and significance.

Despite the critical role these spices play in both the health and culinary industries, widespread knowledge and understanding of the diverse types of spices remain limited, particularly within Indonesian society (Khrisne and Suyadnya, 2018). This knowledge gap is likely due to a combination of factors, including the sheer number of spice varieties and the subtle yet significant visual similarities between certain spices. For instance, distinguishing between ginger and turmeric or between galangal and lesser galangal can be challenging due to their similar appearance (Pratondo et al., 2022). This visual ambiguity often creates difficulties not only for consumers but also for those involved in the trade and export of spices, who require accurate identification for quality control and economic purposes.

Given these challenges, recent advancements in artificial intelligence (AI) have opened new avenues for addressing the problem of spice classification. AI-based techniques have shown promise in similar areas, such as herbal leaf recognition and plant classification, where models leverage image data to distinguish between visually similar objects. One approach to spice recognition that has been explored involves using Histogram of Oriented Gradient (HOG) for feature extraction, combined with a modified K-Nearest Neighbor (K-NN) algorithm for classification. In this method, a dataset of 2250 spice images were utilized, with the Manhattan distance metric employed for calculating distances in the K-NN algorithm. The results of this approach yielded an accuracy of 87%, along with favorable precision, recall, and F1 scores, demonstrating its potential (Melisah and Muhathir, 2023). However, this method has its limitations, particularly when dealing with smaller datasets

ISSN 3088-1803 |

or more complex features, as the HOG technique is optimized for datasets with a lower number of samples.

In contrast, Convolutional Neural Networks (CNNs), a more advanced form of deep learning, have gained popularity for their ability to handle complex image data. CNNs have been widely used for tasks like herbal leaf classification, where they can achieve high levels of accuracy by using transfer learning and pre-trained models such as VGG 16 (Atliha and Sesok, 2020; Rismiyati and Luthfiarta, 2021). Studies have shown that CNNs, when trained on large datasets, can achieve training and testing accuracies as high as 97% and 96%, respectively (Arrofiqoh and Harintaka, 2018). While CNNs offer impressive performance, they also present certain drawbacks—primarily the need for large amounts of labeled data to properly train the model, making it challenging to apply CNNs effectively to smaller datasets like those commonly found in spice classification.

To overcome these limitations, this study proposes the use of autoencoder technology as an alternative approach to classifying spices (Ahmad et al., 2022; Habaragamuwa et al., 2024; Han et al., 2022; Zilvan et al., 2022). Autoencoders (Zilvan et al., 2022), a type of neural network designed for unsupervised learning, have shown promise in tasks involving data compression and feature extraction, making them well-suited for applications where large datasets are unavailable. The autoencoder works by encoding input data into a lower-dimensional space and then decoding it back to its original form, allowing the model to focus on the most critical features of the image. By optimizing key hyperparameters such as learning rate, optimization function, input size, and the number of training epochs, the autoencoder can be fine-tuned to extract distinctive features from the spice images, facilitating more accurate classification. The primary goal of this research is to explore whether autoencoders, when properly refined, can perform on par with or exceed the performance of CNN architectures that use transfer learning. This study seeks to demonstrate that the autoencoder can effectively handle the challenges posed by small, specialized datasets and the complex visual characteristics of spices. By comparing the performance of the autoencoder model with that of traditional CNN architectures, this research aims to assess the feasibility of using autoencoders as a viable alternative for spice classification, even in data-constrained environments.

Academically, this study contributes to the growing body of research on deep learning and its applications in niche fields where data scarcity is a primary obstacle. While CNNs have been the dominant approach for image recognition tasks, this research highlights the potential of autoencoders to achieve similar levels of accuracy without the need for extensive datasets (Liu et al., 2020; Liu and Lin, 2023; Xia and Xiao, 2020) or transfer learning. Furthermore, the findings of this study could serve as a foundation for future research on AI-based classification systems, particularly in fields where the cost of data collection and labeling is high, and where visual ambiguity presents significant challenges.

2. Proposed Method

2.1 Dataset

The dataset used in this study was collected using a camera, ensuring high-quality images with a resolution of 2364×1773 pixels, stored in JPEG format. Each spice was photographed under controlled lighting conditions to minimize shadows and reflections, ensuring consistent image quality. The spices were placed against a plain white background to maintain a neutral setting, which helped in enhancing the contrast between the spices and their surroundings. The dataset consists of five spice categories: ginger, turmeric, galangal, lesser galangal, and Javanese turmeric, with 20 images per category. The selection of these spices was based on their frequent use in traditional Indonesian culinary and medicinal practices. These spices were chosen due to their widespread availability and importance in various applications. Furthermore, they present a visual challenge for classification, given their subtle variations in shape and color, which adds complexity to the model's learning process. This ensures that the dataset is not only representative of common spices but also provides an appropriate test for the robustness of the autoencoder in distinguishing between similar-looking classes.

Figure 1. Example raw spices dataset.

To further diversify the dataset, data augmentation techniques such as horizontal and vertical flipping, rotation, and brightness adjustments were applied. This process resulted in a more robust dataset, simulating various lighting conditions and orientations, which helps in enhancing the model's ability to generalize across different scenarios. Despite these augmentation techniques, the relatively small size of the dataset presented challenges for building a robust classification model, making the use of autoencoders an attractive approach to optimize feature extraction under these constraints. Figure 1 shows a sample of the dataset before preprocessing, illustrating the uniformity of the background and the variability of the spices in terms of their shapes and colors.

2.2 Preprocessing

In preparation for training the CNN models, the acquired dataset underwent a series of preprocessing steps aimed at enhancing model accuracy. Initially, image augmentation was performed using Roboflow, starting with resizing the images to a 1:1 aspect ratio through center cropping, which prevented object stretching. The images were standardized to a size of 1400×1400 pixels. Following this, each image was duplicated through horizontal and vertical flipping and subjected to a 25% reduction in brightness to simulate different lighting conditions. After the augmentation, backgrounds that were initially imperfect due to various factors were cleaned up using the Otsu method for image thresholding. To avoid eliminating objects with colors like the background, a morphological operation with a 20x20 kernel was applied, preserving the object integrity while removing unwanted background artifacts. This process resulted in a dataset with clean, uniform backgrounds conducive to accurate classification by the CNN models (see Figure 2).

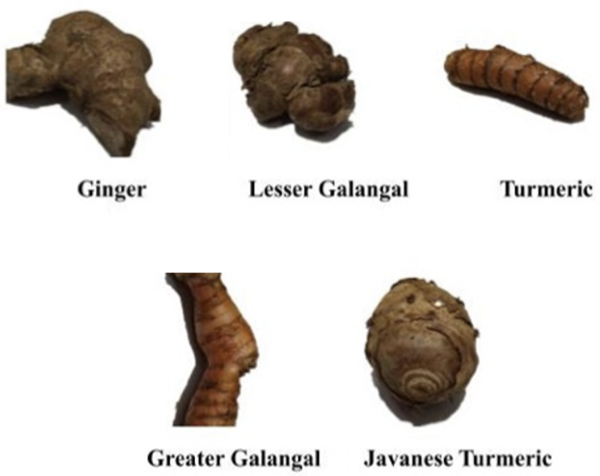


Figure 2. Spices dataset after preprocessing.

For model evaluation, the dataset used for spice classification was divided into training and validation sets. An 80/20 split was employed, with 80% of the dataset allocated for training and 20% for validation. This split was chosen to ensure a sufficiently large portion of the data was available for training while maintaining a reliable set for evaluating the model's performance. Specifically, for the dataset of 100 images (20 images per spice class), 80 images were used for training and the remaining 20 images for validation, ensuring that each spice class was proportionally represented in both sets. Data augmentation was applied to the training set to further diversify the dataset and improve the model's ability to generalize.

2.3 Autoencoder Classifier

The autoencoder model used in this study is designed to handle the challenges of spice classification under limited data conditions. Autoencoders are neural networks that learn to compress input data into a smaller, more efficient representation (latent space) and then reconstruct the original data from this compressed form. This capability makes them particularly useful for feature extraction, especially in situations where the dataset is small or contains noise. The architecture of the autoencoder consists of three main parts: the encoder, the decoder, and the classifier. The encoder takes the input image of size 128×128 pixels with three color channels (RGB), processes it through a series of convolutional layers (Conv2D) with 3×3 filters and ReLU activation functions. The convolution layers extract important features from the input image, while max-pooling layers (2×2) progressively reduce the spatial dimensions, ensuring that only the most critical features are retained. This helps the model focus on distinguishing characteristics, such as the texture and color variations of spices.

Once the image is compressed into a lower-dimensional latent space by the encoder, the decoder takes over. The decoder's role is to reconstruct the original image by upsampling the latent representation. This is done through UpSampling2D layers, followed by additional convolution layers, which gradually restore the image's spatial dimensions to its original 128×128 size. The result is an output that closely resembles the input image, with enhanced focus on the key distinguishing features of each spice.

Finally, the classifier section of the model flattens the output from the decoder and passes it through fully connected (Dense) layers with a softmax activation function. This provides the final classification of the spice. The model was trained using categorical cross-entropy loss and optimized with the Adam optimizer over 50 epochs, with a batch size of 32.

Figure 3 below illustrates the architecture of the autoencoder model, showing the flow from input images through the encoder, the latent space, and back through the decoder to the classifier. The figure highlights how the dimensionality of the images is reduced and restored, allowing the model to focus on the most significant features for classification.

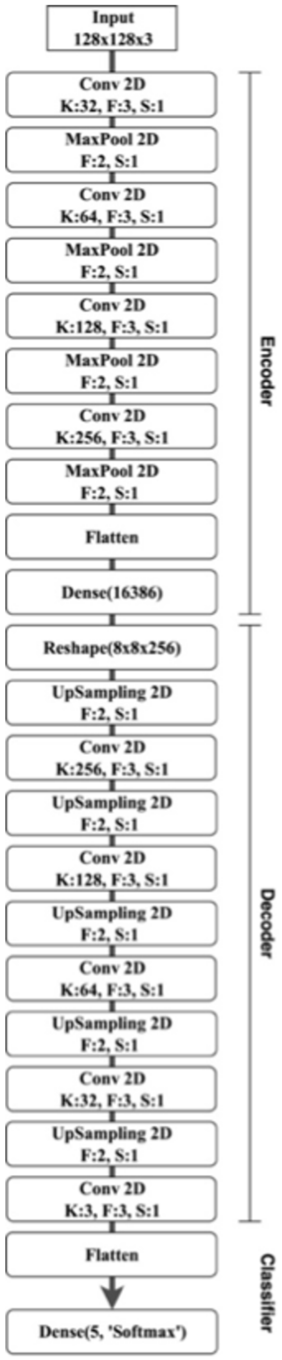


Figure 3. Autoencoder Model.

One of the key advantages of the autoencoder is its ability to avoid overfitting by learning an efficient representation of the data, which is crucial when dealing with small datasets. By focusing on the most important features and discarding irrelevant information, the autoencoder achieves high classification accuracy without requiring many training samples. This is particularly useful for spice classification, where obtaining a large, labeled dataset can be challenging.

3. Result and Discussion

In scenarios where models are trained on limited data but tested on more diverse datasets, there is a significant risk of overfitting. This occurs when a model performs well on the training data but fails to generalize to new, unseen data due to its limited exposure during training. With a small dataset, the model might memorize specific patterns rather than generalizing broader features that are important for accurate classification. In this study, the use of autoencoders helps mitigate this risk by learning compressed representations of the data that focus on essential features for classification. However, testing the model on a more diverse dataset would likely reveal some performance drop, particularly in terms of accuracy and recall. This outcome could be attributed to the model's difficulty in handling unseen variations in spice appearance. To address such issues, methods like transfer learning or more extensive data augmentation techniques could be employed to enhance the model's generalization capability.

A comparative analysis of different deep learning models, including VGG16, ResNet50, MobileNet, and InceptionV3, was conducted to benchmark the performance of the autoencoder. VGG16, known for its depth and hierarchical feature extraction, can often suffer from overfitting when applied to smaller datasets. ResNet50, on the other hand, utilizes residual blocks to combat vanishing gradients, allowing for deeper networks without performance degradation, although it requires a longer training process. MobileNet, optimized for mobile and real-time applications, provides fast and efficient computation with fewer parameters while still maintaining high accuracy. InceptionV3 excels in handling complex images by leveraging multiple kernel sizes in its convolutions. Although these models performed well when pre-trained on large datasets, their reliance on transfer learning to achieve high accuracy highlights the challenge of training models from scratch in domains with limited data availability, like spice classification. In contrast, the autoencoder achieved competitive performance without relying on such large-scale pre-training.

To illustrate the benefits of transfer learning, a comparative analysis was conducted between models trained with and without transfer learning. For the models using transfer learning (VGG16, ResNet50, MobileNet, and InceptionV3), we utilized pre-trained weights from ImageNet and fine-tuned them for the spice classification task. In contrast, the same models were also trained from scratch without leveraging pre-trained weights, requiring them to learn all feature representations solely from the dataset.

Table 1. State-of-the-art Transfer Learning vs. Non-Transfer Learning.

Model	Validation Accuracy (with Transfer Learning)	Validation Accuracy (without Transfer Learning)
VGG16	99.5%	97.1%
ResNet50	95.6%	79.1%
MobileNet	100%	99.8%
InceptionV3	100%	99.8%

The results in Table 1 demonstrate significant differences in model performance. Models with transfer learning achieved higher validation accuracy and required fewer epochs to converge. For instance, VGG16 with transfer learning reached a validation accuracy of 99.5% after 25 epochs, while VGG16 trained from scratch only attained 97.1% within the same timeframe. Similarly, ResNet50 with transfer learning achieved 95.6% validation accuracy, compared to 79.5% for the model trained from scratch. This contrast in performance suggests that the pre-trained weights provided a strong starting point, enabling models to achieve better generalization with a limited dataset.

Furthermore, models trained from scratch showed more signs of overfitting, as indicated by larger gaps between training and validation accuracy. For example, ResNet50 without transfer learning exhibited a noticeable drop in validation accuracy (79.5%) compared to its transfer learning counterpart (95.6%), indicating that the model was more prone to memorizing training-specific patterns. On the other hand, the models using transfer learning generally displayed smaller gaps between training and validation accuracy, suggesting that they were better at capturing generalizable features.

Additionally, MobileNet and InceptionV3 achieved 100% validation accuracy with transfer learning, an unusually high result that could indicate potential overfitting to the validation set. This level of accuracy may suggest that these models have learned patterns specific to the validation data, and further evaluation on an independent test set would help determine the robustness of these models.

Transfer learning provided two clear benefits: first, it allowed the models to learn more effectively from a small dataset by leveraging pre-trained feature extraction layers, and second, it reduced the overall training time needed to achieve optimal performance. This highlights the importance of transfer learning in tasks where data is limited, such as spice classification, as it significantly improves model performance while reducing the risk of overfitting.

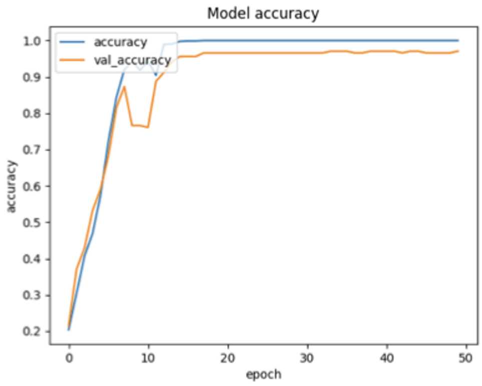
Table 2 provides a detailed comparison of the training and validation performance metrics for each evaluated model, including the autoencoder. Despite its simplicity compared to more complex models like VGG16 and ResNet50, the autoencoder achieved comparable precision and recall without transfer learning, showcasing its ability to adapt to smaller datasets. However, autoencoders are known to have limitations that make them particularly susceptible to overfitting, especially when working with datasets that have high intra-class similarity or limited variety. The autoencoder's training metrics reached near-perfect scores, indicating that it effectively memorized the training data. However, this memorization tendency is also a known drawback of autoencoders. Since autoencoders are designed to compress and then reconstruct input data, they can easily overfit when presented with data that lacks diverse or distinguishing features. In cases where the dataset consists of highly similar visual patterns—such as spices that look alike on the outside—the autoencoder may focus on fine-grained details specific to each training example rather than learning generalized features. This results in a model that performs well on training data but struggles with generalization, as evidenced by the slightly lower validation scores compared to training scores.

Moreover, autoencoders lack the feature extraction capabilities of more complex models, which incorporate multiple convolutional layers and pre-trained weights to capture a broader range of patterns. Unlike models like VGG16 or ResNet50, which can learn hierarchical features and recognize variations within a class, an autoencoder is more prone to overfitting due to its simpler architecture and reliance on direct pixel-to-pixel reconstruction. This weakness becomes apparent when the dataset is small or lacks diversity, as the autoencoder can end up learning the noise or specific quirks of the training set rather than developing robust, transferable representations.

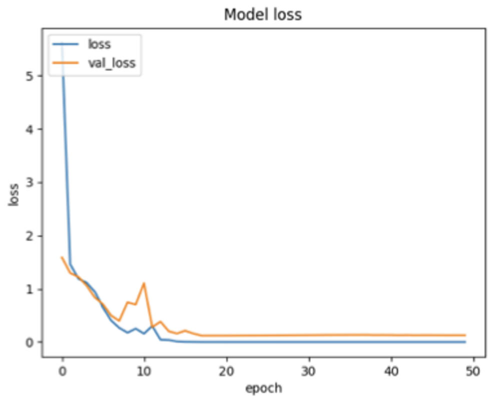
The dataset's inherent limitations—such as the visual similarity among spice classes and the absence of additional sensory information—further exacerbate the overfitting risk. In real-world scenarios, humans rely on attributes like smell or internal texture to differentiate spices that may look visually identical. However, the autoencoder, restricted to visual data alone, may have focused excessively on superficial patterns that do not generalize well, leading to poorer performance on new data. To mitigate this tendency, regularization techniques like dropout and L2 regularization could help the autoencoder learn more generalizable patterns by preventing it from relying too heavily on specific features. Additionally, using data augmentation strategies—such as random rotations, scaling, and brightness adjustments—could introduce greater variability into the training data, helping the model learn more generalized features. If feasible, expanding the dataset or incorporating transfer learning may further help the autoencoder generalize to new data by providing it with a more diverse range of inputs.

Table 2. Model comparison.

Model	Training Loss	Training Accuracy	Training Precision	Training Recall	Validation Loss	Validation Accuracy	Validation Precision	Validation Recall
VGG16	0.008	0.996	0.996	0.996	0.012	0.995	0.995	0.995
ResNet50	0.908	0.772	0.967	0.317	0.899	0.795	0.971	0.326
MobileNet	0.001	1.000	1.000	1.000	0.002	1.000	1.000	1.000
InceptionV3	0.001	1.000	1.000	1.000	0.004	1.000	1.000	1.000
Autoencoder	0.000	1.000	1.000	1.000	0.125	0.970	0.970	0.965



(a)



(b)

Figure 4. Autoencoder (a) accuracy and (b) loss.

The results of this study demonstrate that, despite these limitations, the autoencoder model has significant potential for spice classification, especially in contexts where data is limited. Its high precision and recall make it a valuable tool for industrial applications such as spice sorting in culinary industries, where accuracy is crucial for quality control. As shown in Figure 4, the model achieved stable accuracy and loss values after epoch 20, highlighting its robustness in extended training sessions. Figure 5 further illustrates its consistently high precision and recall values, around 0.97, indicating its capability to extract key discriminative features from the input data. Future work could focus on increasing the dataset's diversity or leveraging transfer learning to further enhance the model's robustness and generalization.

The comparison of computation times and model complexity for the five models—Autoencoder, VGG16, ResNet50, MobileNet, and InceptionV3—is summarized in Table 3. The autoencoder model, with its relatively low parameter count of 1.5 million, was the most efficient in terms of computation time, completing 50 epochs in just 12 minutes. This makes it an attractive option for tasks where rapid training is important, especially in scenarios with limited computational resources.

Table 3. Model Training Time and Parameter Comparison.

Model	Computation Time (50 Epochs)	Number of Parameters
Autoencoder	12 mins	1.5 M
MobileNet	15 mins	4.2 M
InceptionV3	18 mins	23 M
VGG16	25 mins	138 M
ResNet50	30 mins	25 M

Note. M denotes million.

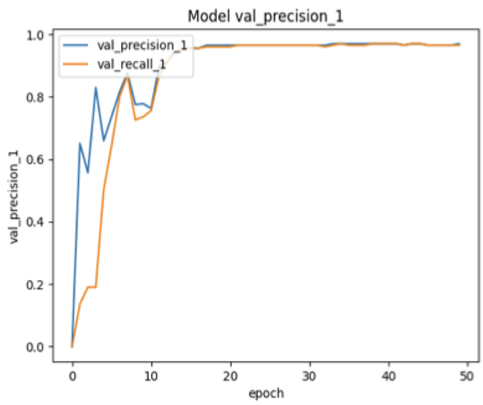


Figure 5. Autoencoder precision and recall.

MobileNet, designed for efficiency, also performed well, completing the training process in 15 minutes with a parameter count of 4.2 million. MobileNet's ability to balance computation time and performance makes it particularly suitable for real-time applications. InceptionV3, with 23 million parameters, took 18 minutes to train, offering a middle ground between speed and model complexity.

On the other hand, VGG16 and ResNet50, with their significantly larger parameter sizes of 138 million and 25 million, respectively, required much longer computation times. VGG16 took 25 minutes to complete 50 epochs, while ResNet50, due to its deeper architecture and residual connections, took the longest at 30 minutes. While these models may provide advantages in accuracy when trained on large datasets, their lengthy training times may not be practical in scenarios where quick results or limited data are required.

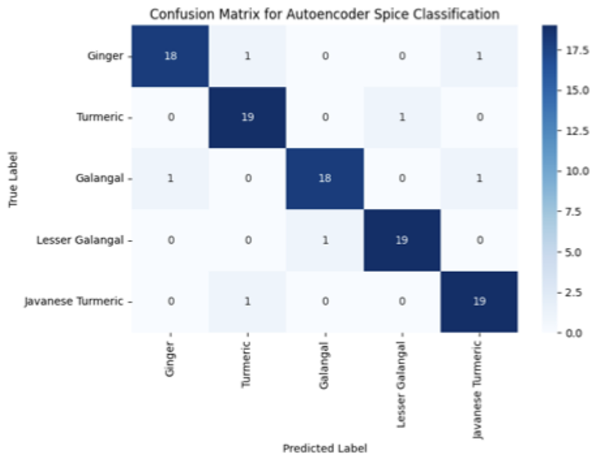


Figure 6. Autoencoder confusion matrix.

The confusion matrix, illustrated in Figure 6, provides valuable insights into the performance of the autoencoder model in classifying five different spice classes: Ginger, Turmeric, Galangal, Lesser Galangal, and Javanese Turmeric. The model performed well across all categories, with high accuracy and minimal misclassifications. For Ginger, the model correctly classified 18 out of 20 images, with one instance being misclassified as Turmeric and another as Javanese Turmeric. This minor confusion could be attributed to the visual similarities between these spices, particularly in colour and texture. Similarly, Turmeric was correctly identified in 19 out of 20 cases, with a single instance misclassified as Lesser Galangal, showing a strong recall of 0.95 but indicating occasional difficulty in differentiating between these two spices. Galangal was also accurately classified 18 out of 20 times, with misclassifications occurring for Ginger and Javanese Turmeric. Despite the high overall accuracy, the misclassifications suggest that the model occasionally struggles with distinguishing spices that share similar shapes or textures. Lesser Galangal exhibited strong classification results, with 19 out of 20 images correctly identified. The only misclassification occurred with Galangal, indicating that the model can generally differentiate between these two spices, though further improvements may be needed to handle subtle visual differences. Finally, Javanese Turmeric was accurately classified in 19 out of 20 instances, with one misclassified as Turmeric.

The autoencoder model we've developed has a lot of promise for practical applications, especially in the spice processing industry where precision really matters. By using this model in automated sorting systems, companies can improve the speed and accuracy of classifying different spices without needing constant human oversight. This could be particularly helpful for sorting spices that look quite similar, like ginger and galangal, which can often be tricky to tell apart. On top of that, the model's scalability means it could be integrated into cloud-based platforms for broader use, offering real-time classification across multiple processing facilities. As more data becomes available, we could further refine the model using techniques like transfer learning to help it handle even more diverse datasets.

While the autoencoder performed well with the limited data we had, there's room to make it even more robust. In real-world scenarios, spices might be partially hidden, or lighting conditions could vary, so adding more diverse data and applying advanced data augmentation techniques could improve the model's ability to deal with these challenges. Compared to the traditional method of spice classification, which often relies on human expertise and is prone to mistakes, our model offers a faster, more reliable solution. It cuts down on the need for expert involvement and speeds up the sorting process while maintaining high accuracy.

4. Conclusion

This study demonstrates the effectiveness of using autoencoder models for spice classification, particularly in scenarios where datasets are limited. The autoencoder performed on par with more complex convolutional neural network architectures, such as VGG16 and ResNet50, while requiring fewer data and avoiding overfitting. With a validation accuracy of 97%, the model has shown that it can extract essential features even in data-constrained environments, making it a viable solution for real-world applications in the spice industry, including automated sorting and quality control.

The results also indicate that autoencoders are not only suitable for image classification tasks but offer significant advantages in areas where labeled data is scarce or difficult to obtain. By focusing on compressing and reconstructing data, the model efficiently captures critical features that are essential for differentiating visually similar spices.

Future work could explore expanding the dataset to include more spice varieties and using transfer learning to further enhance model performance. Additionally, optimizing the autoencoder for real-time processing would make it more applicable to fast-paced industrial environments. Overall, this research contributes to the growing body of knowledge on deep learning applications in niche fields like spice classification and opens the door for further exploration of autoencoders in other constrained data environments.

References

Agrahar-Murugkar, D. (2020). Food to food fortification of breads and biscuits with herbs, spices, millets and oilseeds on bio-accessibility of calcium, iron and zinc and impact of proteins, fat and phenolics. *LWT*, 130:109703. DOI: 10.1016/j.lwt.2020.109703.

Ahmad, B., Sun, J., You, Q., Palade, V., and Mao, Z. (2022). Brain tumor classification using a combination of variational autoencoders and generative adversarial networks. *Biomedicines*, 10(2):223. DOI: 10.3390/biomedicines10020223.

Arrofiqoh, E. N. and Harintaka, H. (2018). Implementasi metode convolutional neural network untuk klasifikasi tanaman pada citra resolusi tinggi. *GEOMATIKA*, 24(2):61. DOI: 10.24895/jig.2018.24-2.810.

Atliha, V. and Sesok, D. (2020). Comparison of VGG and ResNet used as encoders for image captioning. In *2020 IEEE Open Conference of Electrical, Electronic and Information Sciences (eStream)*. IEEE. DOI: 10.1109/estream50540.2020.9108880.

Cardoso-Ugarte, G. A. and Sosa-Morales, M. E. (2021). Essential oils from herbs and spices as natural antioxidants: Diversity of promising food applications in the past decade. *Food Reviews International*, 38(sup1):403–433. DOI: 10.1080/87559129.2021.1872084.

De-Montijo-Prieto, S., Razola-Díaz, M. d. C., Gómez-Caravaca, A. M., Guerra-Hernandez, E. J., Jiménez-Valera, M., Garcia-Villanova, B., Ruiz-Bravo, A., and Verardo, V. (2021). Essential oils from fruit and vegetables, aromatic herbs, and spices: Composition, antioxidant, and antimicrobial activities. *Biology*, 10(11):1091. DOI: 10.3390/biology10111091.

Habaragamuwa, H., Oishi, Y., and Tanaka, K. (2024). Achieving explainability for plant disease classification with disentangled variational autoencoders. *Engineering Applications of Artificial Intelligence*, 133:107982. DOI: 10.1016/j.engappai.2024.107982.

Han, D., Tian, M., Gong, C., Zhang, S., Ji, Y., Du, X., Wei, Y., and Chen, L. (2022). Image classification of forage grasses on Etuoke banner using edge autoencoder network. *PLOS ONE*, 17(6):e0259783. DOI: 10.1371/journal.pone.0259783.

Jiang, T. A. (2019). Health benefits of culinary herbs and spices. *Journal of AOAC International*, 102(2):395–411. DOI: 10.5740/jaoacint.18-0418.

Khrisne, D. C. and Suyadnya, I. M. A. (2018). Indonesian herbs and spices recognition using smaller VGGNet-like network. In *2018 International Conference on Smart Green Technology in Electrical and Information Systems (ICSGTEIS)*, page 221–224. IEEE. DOI: 10.1109/icsgteis.2018.8709135.

Li, T. (2006). *The range of medicinal herbs and spices*, page 113–125. Elsevier. DOI: 10.1533/9781845691717.2.113.

Liu, B., Tan, C., Li, S., He, J., and Wang, H. (2020). A data augmentation method based on generative adversarial networks for grape leaf disease identification. *IEEE Access*, 8:102188–102198. DOI: 10.1109/access.2020.2998839.

Liu, J. and Lin, T. H. (2023). A framework for the synthesis of X-ray security inspection images based on generative adversarial networks. *IEEE Access*, 11:63751–63760. DOI: 10.1109/access.2023.3288087.

Mandal, D., Sarkar, T., and Chakraborty, R. (2022). Critical review on nutritional, bioactive, and medicinal potential of spices and herbs and their application in food fortification and nanotechnology. *Applied Biochemistry and Biotechnology*, 195(2):1319–1513. DOI: 10.1007/s12010-022-04132-y.

Melisah, M. and Muhathir, M. (2023). A modification of the distance formula on the K-nearest neighbor method is examined in order to categorize spices from photo using the histogram of oriented gradient. In *2023 International Conference on Computer Science, Information Technology and Engineering (ICCoSITE)*, page 23–28. IEEE. DOI: 10.1109/iccosite57641.2023.10127780.

Nisa, C. and Candra, F. (2023). Klasifikasi jenis rempah-rempah menggunakan algoritma convolutional neural network: Classification of spice types using the convolutional neural network algorithm. *MALCOM: Indonesian Journal of Machine Learning and Computer Science*, 4(1):78–84. DOI: 10.57152/malcom.v4i1.1018.

Pratondo, A., Elfahmi, E., and Novianty, A. (2022). Classification of curcuma longa and curcuma zanthorrhiza using transfer learning. *PeerJ Computer Science*, 8:e1168. DOI: 10.7717/peerj-cs.1168.

Rismiyati, R. and Luthfiarta, A. (2021). VGG16 transfer learning architecture for Salak fruit quality classification. *Telematika*, 18(1):37. DOI: 10.31315/telematika.v18i1.4025.

Xia, H. and Xiao, M. (2020). 3D human pose estimation with generative adversarial networks. *IEEE Access*, 8:206198–206206. DOI: 10.1109/access.2020.3037829.

Zilvan, V., Ramdan, A., Heryana, A., Krisnandi, D., Suryawati, E., Yuwana, R. S., Kusumo, R. B. S., and Pardede, H. F. (2022). Convolutional variational autoencoder-based feature learning for automatic tea clone recognition. *Journal of King Saud University - Computer and Information Sciences*, 34(6):3332–3342. DOI: 10.1016/j.jksuci.2021.01.020.