Journal of Applied Informatics and Computing (JAIC)
Vol.8, No.2, December 2024, pp. 413~420
e-ISSN: 2548-6861

A Comparison of Convolutional Neural Network (CNN) and Transfer Learning MobileNetV2 Performance on Spices Images Classification

Khoirizqi Velarati [1]*, Christy Atika Sari [2], Eko Hari Rachmawanto[3]
Study Program in Informatics Engineering, Faculty of Computer Science, Universitas Dian Nuswantoro, Indonesia
khoiriz.velarati@gmail.com[1], atika.sari@dsn.dinus.ac.id [2], eko.hari@dsn.dinus.ac.id [3]

Article Info

Article history:

Received 2024-10-02
Revised 2024-10-11
Accepted 2024-10-14

Keyword:

Classification,
Convolutional Neural Network,
MobileNetV2,
Transfer Learning,
Spices.

ABSTRACT

This research was conducted to analyze the performance of the CNN algorithm without transfer learning in classifying spice images and compare it with the CNN algorithm using transfer learning on the MobileNetV2 architecture. This comparison aims to evaluate both methods' accuracy, efficiency, and overall performance and analyze the impact of transfer learning on classification results in the context of spices. The dataset consists of 1500 spice images divided into 10 classes, with each class of 150 images. In the first experiment, CNN without transfer learning resulted in 93% accuracy performance. For the second experiment using MobileNetV2, there was an increase in accuracy, reaching a value of 99% for all spice classes. The results of this study confirm that MobileNetV2 architecture significantly improves the accuracy and performance of spice classification compared to CNN without transfer learning, which can be recommended for spice image classification.

I. Introduction

Spices are parts of the plant commonly used as seasonings, flavorings, fragrances for foods, and food preservatives with limited use [1]. In addition to their essential function in flavoring food, spices, especially in Indonesia, which is a country that has a wealth of spices [2] also have an important place in Indonesia's culinary culture and traditional medicine as well as international commerce. It is proven by its content, which is rich in various active chemical substances, including sulfur compounds, flavonoids, terpenoids, and polyphenols [3]. Unfortunately, with such vital positions, recognition of spices, especially among the younger population, is declining. Most of them struggle to recognize the difference between one spice and another. This is caused by the similarity of several types of spices at a glance and not knowing their characteristics [4].

In this context, artificial intelligence and growing information technology can provide the urgently required solutions. One of the solutions that can be developed is by applying image processing. Image processing is one of the modern technology areas that can be used to convert image object information into digital information to make it better recognized and processed by computers [5].

Doing image processing on spice objects will require a careful and structured approach. One of the approaches that is commonly used in image processing is the Convolutional Neural Network (CNN). As in the previous study, Ponmozhi et al [6] classified 27 types of spices using the CNN approach with 75% accuracy. However, that study did not use the transfer learning method. The transfer learning method is a learning process that can imitate how humans think to make conclusions about other cases [7]. The transfer learning method is a good method to use and choose when the data used is limited [8], such as in this study. One of the architectures commonly used in transfer learning methods is the MobileNetV2 architecture. In another study, Parjito et al [9] have already classified 1554 types of ornamental plants to test and compare the performance of CNN models and transfer learning methods on MobileNetV2 architecture. That study resulted in an average accuracy score of 67% using the CNN model. However, after performing the transfer learning process using the MobileNetV2 algorithm, their average accuracy has increased to 95%. Based on that study, it can be seen that the use of MobileNetV2 architecture significantly impacts image object classification research.

Considering the positive impact of using MobileNetV2 architecture in previous studies, this study will compare the performance and accuracy between CNN models without the transfer learning method and CNN models with the transfer learning method on MobileNetV2 architecture.

This study aimed to evaluate the performance of the CNN algorithm in categorizing spice images without transfer learning and to compare it with the results achieved using the CNN method with transfer learning using the MobileNetV2 architecture. This comparison seeks to assess the accuracy, efficiency, and overall performance of both techniques and investigate the influence of transfer learning on classification outcomes within the domain of spices.

II. METHOD

Several research methods can be used to classifyspice images, such as utilizing image processing techniques with the MobileNetV2 model and without the MobileNetV2 model, which is one of the architectures of the Convolutional Neural Network (CNN) algorithm. The process of identifying spices used in classification with spices is shown in Figure 1.

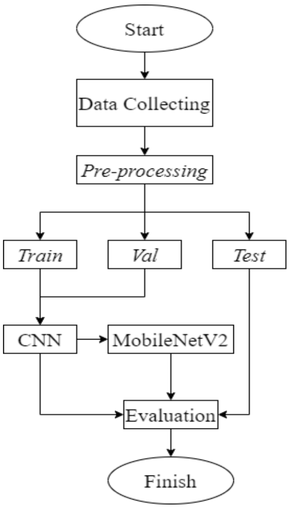


Figure 1: Research Process

A. *Data Collecting*

Research in the classification of spice types takes data form of spice images taken manually by researchers. There are 1500 image data used, which are categorized into 150 image data of clove, 150 image data of ginger, 150 image data of cardamom, 150 image data of aromatic ginger, 150 image data of turmeric, 150 image data of black pepper, 150 image data of white pepper, 150 image data of galangal, 150 image data of nutmeg, and 150 image data of java ginger. While capturing images, researchers used hvs paper as the background of the spices and took a spacing of 20 cm from the camera. Because the method of collecting spice image data is done manually, where each image is taken very carefully to ensure the quality of each image, it can be assured that the collected images are clean from duplication, noise, or other anomalies that could be the reason why the data cleaning process is needed. Considering this fact, the steps that will be carried out in this research after collecting all necessary data is pre-processing without performing data cleaning.

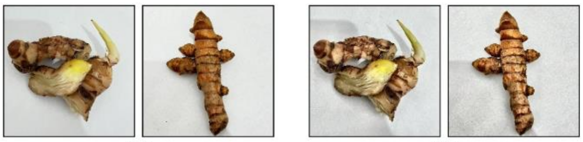
B. *Pre-processing*

After all the image data of spices needed in the study are collected, the next step is data pre-processing. Several steps are carried out during this pre-processing. The first step is format standardization, which is to change the entire image format into the same format. In addition to ensuring the consistency of the format for all image data, the purpose of format standardization is to minimize errors and increase efficiency for the next process because when the format are different, it can give different characteristics to these images [10]. The format used in this study is JPG format. After all data has the same JPG format, the next step is to resize the image to change the image size to 224 x 224 pixels so that the dimensions of each image become consistent. The next step is the image normalization step. Image normalization is a step that can be taken to accelerate convergence, improve accuracy, and ensure model consistency by adjusting the ratio and distribution of image pixel values. Image normalization is a step that can be taken to accelerate convergence, improve accuracy, and ensure model consistency by adjusting the ratio and distribution of image pixel values [11]. In image normalization, various techniques can be useful to normalize images: Min-Max Normalization, Z-Score Normalization, Mean Normalization, Contrast Normalization, and others. In this study, researchers decided to implement the Min-Max Normalization technique. The Min-Max normalization technique is one of the normalization techniques that works by transforming the original data linearly to obtain a balanced comparison result value between the data before and after the process. This technique can be used to normalize images by setting pixel values from 0 to 1 [12]. The functions of the Min-Max normalization technique are:

$$x_{new} = \frac{x_{old} - x_{min}}{x_{max} - x_{min}}$$

Where x_{new} is the pixel value that has been normalized, x_{old} is the original pixel value before normalization, x_{max} is the maximum pixel value in the original image, and x_{min} is the minimum pixel value in the original image [13].

After normalizing using the Min-Max Normalization technique, which was applied to all image data, the next step was performed to improve the contrast value of the images using the Contrast Limited Adaptive Histogram Equalization (CLAHE) technique. This CLAHE technique is an advanced version of the AHE (Adaptive Histogram Equalization) technique, which is a way to improve image contrast by expanding the intensity range or extending the most frequently occurring intensity values. In the CLAHE technique, each image will be divided into several sub-images or blocks, and each block will be subjected to histogram equalization. This process can avoid excessive amplification and redistribution of pixels, which makes the image contrast clearer and more visible [14]. The result of applying the Clahe technique can be seen in Figure 2 below.



(a) (b)

Figure 2. Difference between the original image and the image that has been given CLAHE technique (a) original image, (b) image after CLAHE technique

The next step is to split the data into train data, validation data, and test data. In this study, researchers used a ratio of 7:1:2 with 70% for train data, 10% for validation data, and 20% for test data. These ratios were chosen to provide sufficient data for the training and evaluation phases of the model. These splits are randomly generated using the random shuffle method. After successfully splitting the data according to the ratio above, the data will be saved in a new folder that will be used to perform the next step.

Then, before moving on to the modelling stage, it is necessary to perform an image augmentation process. Image augmentation is a technique that can generate new image with various image orientations, such as increasing the size and shape of the image [15]. By performing image augmentation, more data can be generated, and overfitting becomes smaller. This process is performed to increase the size of the dataset by making several changes to the data in the hope that the model will not memorize specific patterns in the data [16]. This research uses several augmentation techniques, including 20-degree rotation, 20-degree shear, 20-degree zoom, and horizontal data reversal. The ImageDataGenerator from Keras will be used to perform these techniques, and the data that has been augmented into the model will be forwarded.

C. Convolutional Neural Network

After augmenting the data, the next step is training the model. In this study, the first model is derived from the Convolutional Neural Network model without the transfer learning method, commonly called a simple CNN. CNN, or Convolutional Neural Network, is one type of neural network known for its high accuracy, especially in classification tasks. The CNN algorithm has several layers, such as an input layer, an output layer, and various other hidden layers. The most common hidden layers in CNN include convolutional layers, fully connected layers, pooling layers, normalization layers, and pooling layers (ReLU)[17]. An example of these layers can be seen in Figure 3 below.

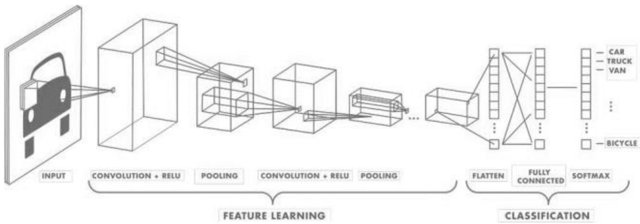


Figure 3. Convolutional Neural Network

In this training, there are three steps performed for the model training process using CNN, which are:

1) Designing the Model: The first step is designing the model. The input shape used is 224 x 224 pixels with three colour channels (RGB). Several main layers are used sequentially in a simple CNN. First, the model starts with a convolution layer using a 3x3 kernel and ReLu activation function followed by a MaxPooling layer. In this research, 4 layers are used, and each layer is given a gradually increasing number of filters, which are 32, 64, until 128. The purpose of this increase is to allow the model to capture more complex patterns. Furthermore, the output of the flattening layer is combined with the fully connected layer using 512 units and ended by using a softmax activation function in the output layer.

2) Perform Model Optimization: In this model design, optimization uses Adam Optimizer with a learning rate of 0.0001 and a loss function categorical cross-entropy.

3) Train the Model: The maximum number of epochs used in training the model with a simple CNN is 25. The training process is carried out by applying 32 batch sizes. At each epoch, performance will be evaluated using the validation set. The program will execute the callback early stop function if the validation loss is not improved for 10 consecutive epochs.

D. MobileNetV2

Furthermore, this research used a second model, the MobileNetV2 model, one of the architectures in the Convolutional Neural Network algorithm. The green and blue boxes in Figure 4 below illustrate the use of a combination of depthwise convolution and pointwise convolution.

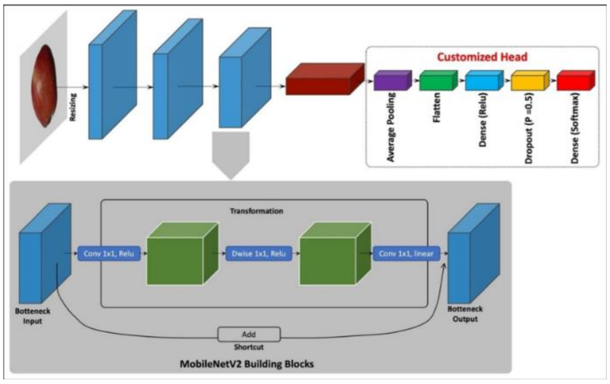


Figure 4. Transfer Learning – MobileNetV2 Model

MobileNetV2 is a transfer learning model previously trained on feature extraction and image classification. This architecture can be used on mobile devices and embedded systems because it is based on pointwise and depthwise convolution layers and is quite lightweight [18]. In MobileNetV2, the input image will be separated into different channels, and each channel will be convolved separately. The results of the convolution will be recombined using 1×1 pointwise convolution [19]. Three steps must be completed in the model training process using MobileNetV2, including:

1) *Designing the Model*: MobileNetV2 was chosen as the model to be used in this model design step. Using pre-trained weights from ImageNet, researchers added several additional layers, including GlobalAveragePooling2d to reduce dimensionality, Dense layer with 1024 units and ReLu activation, BatchNormalization and Droupout to reduce overfitting with a rate of 0.5, and softmax activation as the output layer for multi-class classification.

2) *Perform Model Optimization*: The model will be optimized using Adam Optimizer, which is a combination of AdaGrad and RMSProp optimization [20]. The folllowing below is the function of the Adam Optimizer.

$$\text{Adam Optimizer: } \sigma_{n+1} = \sigma_n - \frac{\delta}{\sqrt{h_n + \tau}} m_n \textit{[21]}$$

For the loss function, this study will employ categorical cross-entropy loss function that can be used to help the model produce more accurate predictions [22].

3) *Train the Model*: In the training step of the initial model, the maximum number of epochs used is 25 epochs where the performance of each epoch will be evaluated using validation loss, then the early stop callback function will be executed.

E. Evaluation

At the model evaluation stage, the data used is test data. The purpose of this evaluation is to find out how the previously built model performs. This stage is carried out using several evaluation metrics as follows:

1) Precision: Also known as the positive predictive value, this is the value of how accurate the model is when making positive predictions by calculating the proportion of datasets classified as error-prone [23]. The following calculation can determine the precision value:

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

True positive is the amount of data that is actually positive and has been correctly classified as positive by the model. In contrast, a false positive is the amount of data that is actually negative and has been wrongly classified as positive by the model.

2) Recall: This value represents the proportion of correct results compared to the total expected results [24]. The calculation to determine the recall value is:

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

3) F1-Score: It is an ideal metric to assess a classification's performance because it can consider the model's accuracy and predictive ability. It combines precision and gain by calculating the harmonic mean [25]. The calculation to get the F1-Score is:

$$\text{F1-Score} = \frac{2 \cdot (\text{Precision} \cdot \text{Recall})}{\text{Precision} + \text{Recall}}$$

4) Accuracy: This is the value of evaluating the accuracy of a whole classification model [26]. The following bellow is a calculation that can be used to determine the accuracy value:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Predictions}}$$

III. Results and Discussion

Based on Figure 5, datasets used in this study are image data obtained from field studies by taking pictures directly using a smartphone camera with a 12mp resolution.

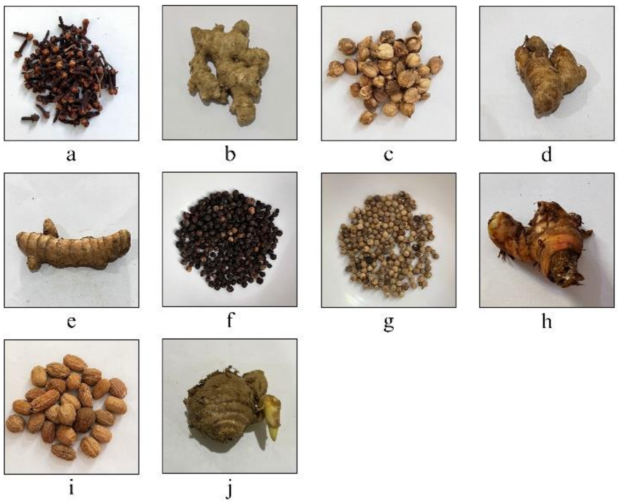


Figure 5. Spice image samples (a) clove image, (b) ginger image, (c) cardamom image, (d) aromatic ginger image, (e) turmeric image, (f) black pepper image, (g) white pepper image, (h) galangal image, (i) nutmeg image, and (j) java ginger image

In the process of taking images, researchers used white paper as a background for spices and took a distance of 20 cm from the camera. The data used are 1500 images, divided equally by clove, ginger, cardamom, aromatic ginger, turmeric, black pepper, white pepper, galangal, nutmeg, and java ginger.

A. *CNN Classification*

The first test of image data is using the CNN method. The results of these tests are in Table 1.

TABLE I
CNN TESTING RESULTS

Scenario	Precision	Recall	F1-Score	Accuracy
Clove	1.00	0.90	0.95	0.93
Ginger	0.92	0.73	0.81	0.93
Cardamom	0.96	0.87	0.91	0.93
Aromatic Ginger	0.82	0.90	0.86	0.93
Turmeric	0.97	1.00	0.98	0.93
Black Pepper	0.97	1.00	0.98	0.93
White Pepper	1.00	1.00	1.00	0.93
Galangal	0.94	1.00	0.97	0.93
Nutmeg	0.83	1.00	0.91	0.93
Java Ginger	0.97	0.93	0.95	0.93

This study has conducted several experiments using the CNN algorithm to evaluate the model's performance. Based on the results of these experiments, there is a variation in the model's accuracy, which shows that the results of the CNN model are not always consistent in each trial that has been carried out. As a representation of the optimal performance of the CNN model tested, the best results from several experiments can be seen in Table 1 above.

These results shows that the test classification using CNN results varies between the 10 types of spices tested. Table 1 shows the value of evaluation metrics, including precision value, recall value, F1-Score value, and accuracy value for each type of spice tested. Overall, the CNN model produces an average precision of 0.94, recall of 0.93, F1-Score of 0.93, and accuracy of 0.93. The model's performance also looks outstanding in several types of spices, such as the best results obtained for the white pepper image with precision, recall, F1-Score, and accuracy values of 1.00, 1.00, 1.00, and 0.93. In addition, turmeric also shows good performance with values of precision 0.97, recall 1.00, F1-score 0.98, and accuracy 0.93. This indicates that the CNN method without transfer learning can quite classify the white pepper and turmeric classes.

On the other hand, unsatisfactory results were obtained in several types of spices, such as the lowest performance produced in the ginger class with a precision value of 0.92, recall 0.73, F1-score 0.81, and accuracy value of 0.93. This shows that the CNN method without transfer learning is not optimal for recognizing patterns in the ginger class well. From the overall results, the model works quite well but requires improvement to achieve better results, especially in spice types that show poor results, such as nutmeg and curcuma.

Figure 6 is a graph of two metrics resulting from measurements during the model training process: training accuracy and training loss compared to validation accuracy and validation loss. Both graphs showed information about the results of the CNN model performance during training based on the number of epochs used. In the training accuracy graph, it can be seen that training and validation increase as the number of epochs increases. The graph shows that the accuracy of both datasets increases sharply during the initial period, reaching its highest point around the 15th to 25th epoch. The small difference between the training and validation curves indicated that the model learns well without experiencing significant overfitting. Overfitting is a situation where the model cannot properly generalize from training data to validation data [27].

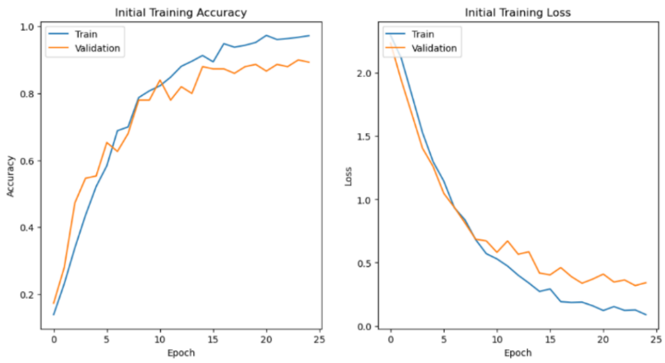


Figure 6. CNN Performance

As for the training loss graph, there is a drastic decrease in the initial epoch. This indicates a considerable improvement in the model's performance. However, there are signs of overfitting in the validation loss values which do not decrease as much as the loss in the training data.

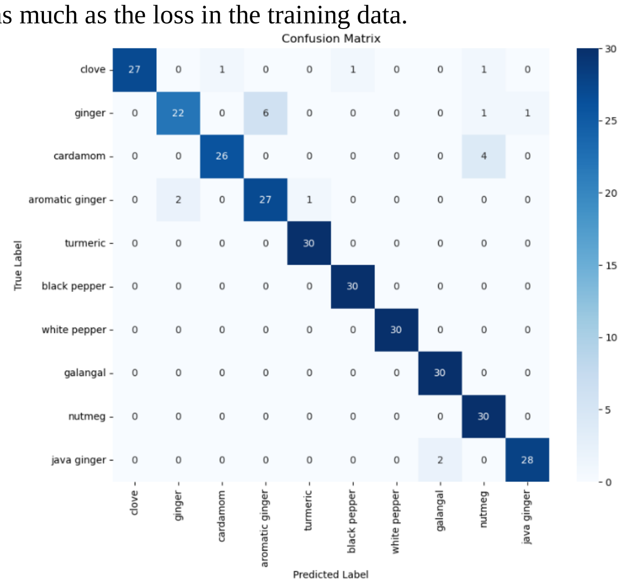


Figure 7. CNN Confusion Matrix

Figure 7 above is the result of the Confusion Matrix, which shows how the CNN model works to classify 10 types of spices. Each class has been tested with 30 samples to measure how accurate the model is in classifying every kind of spice.

The matrix shows two categories of results, namely true positive and false negative. The clove class had 27 true positives and 3 false negatives, where the samples were incorrectly predicted as cardamom, black pepper, and nutmeg. As for the ginger class, the true positive value is 22 and the false negative is 8, which is the sample wrongly predicted as aromatic ginger, nutmeg, and java ginger. Furthermore, the cardamom class has 26 true positives and 4 false negatives with a prediction error as nutmeg. In the aromatic ginger class, 27 true positives and 3 false negatives were obtained with prediction errors in the ginger and turmeric classes. Perfect results were obtained for the turmeric, black pepper, white pepper, galangal, and nutmeg classes with 30 true positives and 0 false negatives. Lastly, the java ginger class resulted in 28 true positives and 2 false negatives with a misprediction to galangal.

B. MobileNetV2 Classification

Furthermore, the second test was carried out using the MobileNetV2 architecture. The results of the test can be seen in Table 2 below. In Table 2, it can be seen that the results of the classification test using MobileNetV2 show excellent performance results, where the MobileNetV2 model can maintain a high accuracy rate of 0.99, with an average F1-Score of 0.99 when used in detecting various types of spices. Significantly, MobileNetV2 can improve the performance of the model, especially in classes that previously showed poor performance in testing using CNN without transfer learning. For example, the ginger class, which previously only obtained an F1-Score value of 81%, increased to 100% when transfer learning with MobileNetV2 was applied.

TABLE II
MOBILENETV2 TESTING RESULTS

Scenario	Precision	Recall	F1-Score	Accuracy
Clove	1.00	1.00	1.00	0.99
Ginger	1.00	1.00	1.00	0.99
Cardamom	1.00	1.00	1.00	0.99
Aromatic Ginger	0.97	1.00	0.98	0.99
Turmeric	1.00	1.00	1.00	0.99
Black Pepper	1.00	0.97	0.98	0.99
White Pepper	0.97	1.00	0.98	0.99
Galangal	1.00	1.00	1.00	0.99
Nutmeg	1.00	0.97	0.98	0.99
Clove	1.00	1.00	1.00	0.99

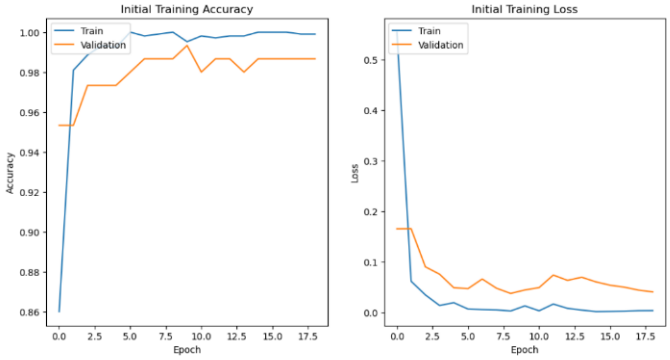


Figure 8. MobileNetV2 Performance

Figure 8 shown a graph showing the results of measurements during the model training process using MobileNetV2 transfer learning. The results consist of two metrics: accuracy and loss. The training accuracy graph shows that the model produces training accuracy values that increase sharply in just a few epochs and remain stable until the end of training. Validation accuracy is also seen to increase sharply to more than 98% in the early epochs, and it remains stable without a significant decrease. Such stability may indicate that the model can generalize well to new data.

On the other hand, in the training loss graph, it can be seen that the value drops very drastically in the first epoch and reaches a very low value in a short time, followed by a stable low value until the end of training. For the validation loss, the decrease is also quite significant, below 0.1, and continues to fall until it is close to 0. This shows that the model can learn quickly and efficiently from the data.

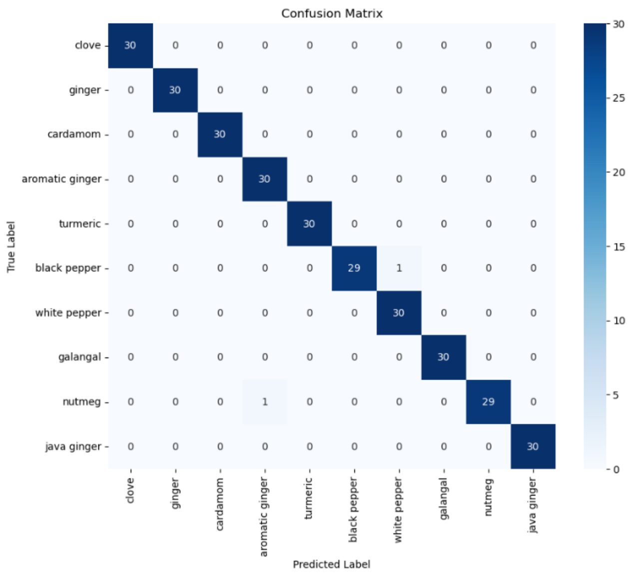


Figure 9. MobileNetV2 Confusion Matrix

Figure 9 above shows the results of the Confusion Matrix produced by the MobileNetV2 model. The model performs exceptionally well in predicting the 30 samples provided. Of the 10 spice classes predicted, 8 achieved perfect scores with 30 true positives and 0 false negatives. The remaining two classes, black pepper and nutmeg, resulted in 29 true positives and 1 false negative, with the errors showing black pepper predicted as white pepper and nutmeg as aromatic ginger.

IV. CONCLUSION

After conducting this study in the form of classification on 10 types of spice data including clove, ginger, cardamom, aromatic ginger, turmeric, black pepper, white pepper, galangal, nutmeg, and java ginger with 1500 data in total, it was obtained test results from the use of CNN and MobileNetV2 algorithms. Test results on the CNN method produce consistent values with an average accuracy of 93%. While in the test results using the MobileNetV2 architecture, constant accuracy results were obtained, namely 99%. These numbers show that MobileNetV2 can capture most of the important features of the dataset.

In addition, the results of the confusion matrix analysis also show that MobileNetV2 is better than CNN in the classification of spices. CNN has many misclassification errors in clove, ginger, cardamom, aromatic ginger, and java ginger. In MobileNetV2, the classification can produce almost perfect results for each class, and errors only occur in the black pepper and nutmeg classes.

From these two tests, it can be concluded that the MobileNetV2 architecture can significantly improve classification performance compared to simple CNN. Therefore, MobileNetV2 architecture can be suggested in the classification of spice images.

BIBLIOGRAPHY

[1] M. I. Al-Amin, F. N. Sidiq, D. R. Ramdania, N. Fajar, Y. A. Gerhana, and M. Harika, "Spices Image Classification Using Support Vector Machine," in *2022 10th International Conference on Cyber and IT Service Management, CITSM 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 1–4. doi: 10.1109/CITSM56380.2022.9935856.

[2] R. Widyani, M. Nilamcaya, and D. Iftitah, "The Benefits of Indonesian Spices for Livestock Health As A Legacy of Our Ancestors Local Wisdom," *IOP Conf Ser Earth Environ Sci*, vol. 1020, no. 1, pp. 1–11, May 2022, doi: 10.1088/1755-1315/1020/1/012032.

[3] R. Velázquez, A. Rodríguez, A. Hernández, R. Casquete, M. J. Benito, and A. Martín, "Spice and Herb Frauds: Types, Incidence, and Detection: The State of the Art," *Foods*, vol. 12, no. 18, pp. 1–38, Sep. 2023, doi: 10.3390/foods12183373.

[4] I. P. W. Prasetia and I. M. G. Sunarya, "Image Classification of Spices Basic Ingredients for Making Bumbu Bali Base Genep Based on Deep Learning," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 13, no. 1, pp. 79–90, Mar. 2024, doi: 10.23887/janapati.v13i1.67967.

[5] Y. Zhang and X. Zheng, "Development of Image Processing Based on Deep Learning Algorithm," in *2022 IEEE Asia-Pacific Conference on Image Processing, Electronics and Computers, IPEC 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 1226–1228. doi: 10.1109/IPEC54454.2022.9777479.

[6] D. K. Ponmozhi and R. Jose, "A REVIEW ON SPICES CLASSIFICATION AND RECOGNITION," *Transtellar Publications*, vol. 12, no. 1, pp. 25–32, 2022, [Online]. Available: www.tjprc.org

[7] C. Li, J. Feng, L. Hu, J. Li, and H. Ma, "Review of Image Classification Method Based on Deep Transfer Learning," in *Proceedings - 2020 16th International Conference on Computational Intelligence and Security, CIS 2020*, Institute of Electrical and Electronics Engineers Inc., Nov. 2020, pp. 104–108. doi: 10.1109/CIS52066.2020.00031.

[8] A. Hosna, E. Merry, J. Gyalmo, Z. Alom, Z. Aung, and M. A. Azim, "Transfer learning: a friendly introduction," *J Big Data*, vol. 9, no. 102, p. 1, Oct. 2022, doi: 10.1186/s40537-022-00652-w.

[9] Parjito, F. Ulum, K. Muludi, Z. Abidin, R. M. Alma, and Permata, "Classification of Ornamental Plants with Convolutional Neural Networks and MobileNetV2 Approach," in *Proceedings - ISMODE 2022: 2nd International Seminar on Machine Learning, Optimization, and Data Science*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 534–539. doi: 10.1109/ISMODE56940.2022.10180988.

[10] D. Watni and S. Chawla, "Impact of various image formats supported by android smartphones on image steganography: a preliminary study," *Multimed Tools Appl*, vol. 83, no. 11, pp. 33031–33050, Mar. 2023, doi: 10.1007/s11042-023-16343-4.

[11] X. Pei *et al.*, "Robustness of machine learning to color, size change, normalization, and image enhancement on micrograph datasets with large sample differences," *Mater Des*, vol. 232, pp. 1–13, 2023, doi: 10.1016/j.matdes.2023.112086.

[12] D. R. Nayak, N. Padhy, P. K. Mallick, M. Zymbler, and S. Kumar, "Brain Tumor Classification Using Dense Efficient-Net," *Axioms*, vol. 11, no. 1, pp. 1–13, Jan. 2022, doi: 10.3390/axioms11010034.

[13] A. AKILLI and H. ATIL, "Evaluation of Normalization Techniques on Neural Networks for the Prediction of 305-Day Milk Yield," *Turkish Journal of Agricultural Engineering Research*, vol. 1, no. 2, pp. 354–367, 2020, doi: 10.46592/turkager.2020.v01i02.011.

[14] E. A. Tjoa, I. P. Y. N. Suparta, R. Magdalena, and N. K. CP, "The use of CLAHE for improving an accuracy of CNN architecture for detecting pneumonia," *SHS Web of Conferences*, vol. 139, pp. 1–7, 2022, doi: 10.1051/shsconf/202213903026.

[15] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/j.gltp.2022.04.020.

[16] H. Naveed, S. Anwar, M. Hayat, K. Javed, and A. Mian, "Survey: Image mixing and deleting for data augmentation," *Eng Appl Artif Intell*, vol. 131, May 2024, doi: 10.1016/j.engappai.2023.107791.

[17] J. Gupta, S. Pathak, and G. Kumar, "Deep Learning (CNN) and Transfer Learning: A Review," *J Phys Conf Ser*, vol. 2273, no. 1, pp. 1–10, 2022, doi: 10.1088/1742-6596/2273/1/012029.

[18] N. I. R. Yassin, "Fundus Images Classification of Diabetic Retinopathy using MobileNetV2," *International Journal of Computer Science and Mobile Computing*, vol. 12, no. 5, pp. 54–63, May 2023, doi: 10.47760/ijcsmc.2023.v12i05.006.

[19] R. Indraswari, R. Rokhana, and W. Herulambang, "Melanoma image classification based on MobileNetV2 network," *Procedia Comput Sci*, vol. 197, pp. 198–207, 2022, doi: 10.1016/j.procs.2021.12.132.

[20] Y. Wang, Z. Xiao, and G. Cao, "A convolutional neural network method based on Adam optimizer with power-exponential learning rate for bearing fault diagnosis," *Journal of Vibroengineering*, vol. 24, no. 4, pp. 666–678, Feb. 2022, doi: 10.21595/jve.2022.22271.

[21] J. Ghosh and S. Gupta, "ADAM Optimizer and CATEGORICAL CROSSENTROPY Loss Function-Based CNN Method for Diagnosing Colorectal Cancer," *2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES)*, pp. 470–474, 2023, doi: 10.1109/CISES58720.2023.10183491.

[22] P. Das, S. Gupta, J. Patra, and B. Mondal, "ADAMAX optimizer and CATEGORICAL CROSSENTROPY loss function-based CNN method for diagnosing Lung cancer," *7th International Conference on Trends in Electronics and Informatics, ICOEI 2023 - Proceedings*, pp. 806–810, 2023, doi: 10.1109/ICOEI56765.2023.10126046.

[23] A. J. E. Taiwo, E. A. Olajubu, and G. A. Aderounmu, "Development of Threat Hunting Model Using Machine Learning Algorithms for Cyber Attacks Mitigation," *Proceedings - 2022*

International Conference on Computational Science and Computational Intelligence, CSCI 2022, pp. 1010–1015, 2022, doi: 10.1109/CSCI58124.2022.00179.

[24] P. Fränti and R. Mariescu-Istodor, "Pattern Recognition Letters," *Pattern Recognit Lett*, vol. 167, pp. 115–121, Mar. 2023, doi: 10.1016/j.patrec.2023.02.005.

[25] N. G. Sorboni, J. Wang, and M. R. Najafi, "Fusion of Google Street View, LiDAR, and Orthophoto Classifications Using Ranking Classes Based on F1 Score for Building Land-Use Type Detection," *Remote Sens (Basel)*, vol. 16, no. 11, pp. 1–27, Jun. 2024, doi: 10.3390/rs16112011.

[26] F. Najjar, R. T. Naha, M. M. Feridani, O. Dekhil, and K. Zhang, "Change Address Detection in Bitcoin using Hierarchical Clustering," *Proceedings - 2024 IEEE World Forum on Public Safety Technology, WFPST 2024*, pp. 42–48, 2024, doi: 10.1109/WFPST58552.2024.00015.

[27] G. K. Gupta and D. K. Sharma, "A Review of Overfitting Solutions in Smart Depression Detection Models," *2022 9th International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 145–151, 2022, doi: 10.23919/INDIACom54597.2022.9763147.