An Ensemble Convolutional Neural Network Approach for Image Classification of Indonesian Endemic Fruits

Monica Widiasri[1*], *Joko* Siswantoro[1], and *Alexander* Kenrick Duanto[1]

[1]Informatics Engineering, Universitas Surabaya, Surabaya, Indonesia

Abstract. Indonesia has a diverse range of endemic fruits that grow in its various regions. These fruits have their own distinctive characteristics, which can sometimes lead to confusion in the sorting process. Classification can be used as a solution to this problem. Several similar studies have classified fruits; however, there has been no research specifically using deep learning methods for Indonesia's endemic fruits. The designed system is expected to classify fruits accurately based on their unique characteristics. The classification models used consist of three CNN architecture models: AlexNet, ResNet-50, and InceptionV3, which are then combined with an ensemble method. Each model is compared by evaluating the use of transfer learning and without it. The three models with the most optimal results are implemented in an ensemble application. The best results were obtained from the AlexNet model, with an accuracy of 99.67%, the InceptionV3 model, with an accuracy of 99.81%, and the ResNet-50 model, with an accuracy of 100%. All three models are implemented in an ensemble using the majority voting method. The results of the ensemble implementation yield an accuracy of 100% on the test dataset.

1 Introduction

In categorizing these fruits, the process sometimes requires a considerable amount of time. Conventional methods, such as assigning codes and matching them with a catalog book, not only require more time but also demand more effort to label each fruit individually [1]. Therefore, a new method is needed to accelerate the fruit identification process. The use of image recognition technology can help expedite this process.

Convolutional Neural Networks (CNNs) are an effective approach for classifying images. CNNs share a concept like Artificial Neural Networks (ANN), which consist of many interconnected nodes called neurons. This concept is inspired by how the human brain's nervous system works. The main difference between CNNs and ANNs is that in CNNs, neurons can optimize themselves during the learning process. CNNs are considered more suitable for image detection because the CNN system prioritizes pattern recognition in an image. These patterns make it easier for the system to encode a specific image into an architecture [2].

[*] Corresponding author: monica@staff.ubaya.ac.id

The image classification process is carried out using three models: AlexNet, InceptionV3, and ResNet-50. These models are used to generate predictions from each model. Subsequently, an ensemble process with majority voting determines the final prediction result from all models. The ensemble method is employed because it is considered to produce better performance compared to using a single algorithm [3].

The developed system is expected to produce a model that can assist in classifying Indonesia's endemic fruits. The benefit of developing this model is to facilitate the identification of Indonesian endemic fruits, thereby reducing the time required for fruit classification.

2 Methods

The research methodology employed in this study comprises the following steps: dataset collection, image preprocessing, model training, ensemble process, evaluation, and testing.

2.1 Dataset Collection

Endemic fruits are types of fruit that grow naturally and exclusively in a particular area [1]. The dataset used in this study is Ubaya-IFDS3000, which comprises 15 classes of Indonesian endemic fruits: ambarella, avocado, dragon fruit, duku, durian, guava, mangosteen, pacitan orange, persimmon, pineapple, salak, sapodilla, siam lime, soursop, and starfruit, as illustrated in Figure 1. Each class contains 200 images with five different background colors, namely white, pink, light yellow, light green, and light blue, as shown in Figure 2. The total number of images in the dataset is 3,000 images.

The image dataset will be divided into 80% for the training and evaluation process, 20% for the testing process, with a total of 2400 images and 600 images, respectively. Of the 2,400 images in training and evaluation, 70% (1,680 images) are allocated for training and 30% (720 images) for validation. Images were loaded in batches of 64, with labels assigned based on their directory structure.

The images were acquired at a resolution of 72 dpi with dimensions of 2592 × 1456 pixels and stored in the JPEG format, which is commonly utilized as a standard for digital image storage. The acquisition process was completed under lighting settings of 1050 lumens and 160 lumens, delivered from two different angles (0˚ and 45˚).

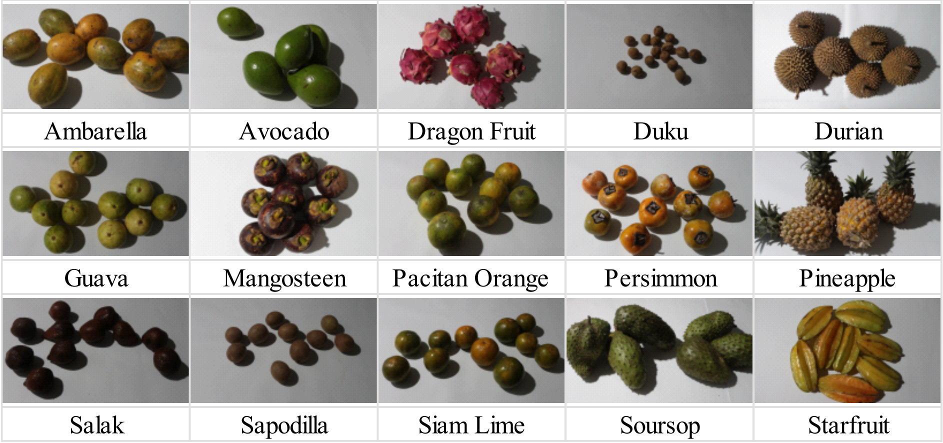


Fig. 1. Image samples of the Ubaya-IFDS3000 dataset

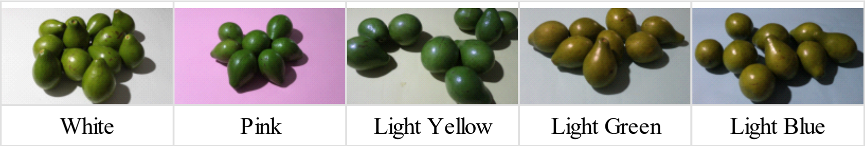


Fig. 2. Background variation samples from the Ubaya-IFDS3000 dataset

2.2 Image Preprocessing

The image to be processed will be resized to 227×227×3, and the pixel values will be normalized to a range of 0.0–1.0. For the training process, image augmentation is performed by randomly zooming, rotating, and flipping the image horizontally, to enhance dataset variability and reduce overfitting [4].

2.3 Model Training

The classification model training process is formed using transfer learning or a pre-trained architecture. The model architecture uses the Ensemble CNN method, which combines the AlexNet, ResNet-50, and InceptionV3 models. Each model is compared by evaluating its use of transfer learning and its performance without it. The three models with the most optimal results are implemented in an ensemble application.

2.3.1 AlexNet

AlexNet is a convolutional neural network (CNN) architecture comprising five convolutional layers and three fully connected layers [5], with an input size of 227 × 227 × 3, representing RGB channels. The convolutional layers utilize ReLU activation and max pooling operations to reduce spatial dimensions while preserving essential features. The final fully connected layers employ a softmax activation function to generate classification outputs. A detailed structure of the AlexNet architecture is presented in Figure 3.

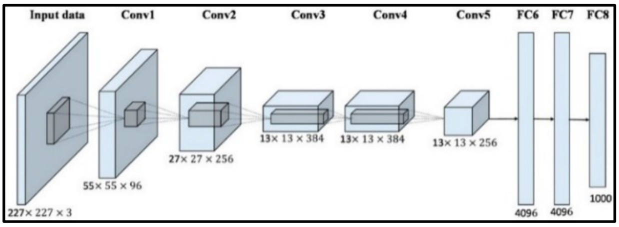


Fig. 3. AlexNet Architecture [6]

2.3.2 ResNet50

ResNet50 is a deep CNN consisting of 50 layers that employs residual blocks to address the degradation problem in very deep networks [7]. The model takes an input image of 227 × 227 × 3. The first layer applies a 7 × 7 convolution (stride = 2) with 64 filters, followed by a 3 × 3 max pooling (stride = 2).

The network then proceeds through four convolutional stages, each composed of residual blocks repeated 3, 4, 6, and 3 times, respectively. Finally, the features are passed through an average pooling layer and a fully connected layer with softmax activation for classification (Figure 4).

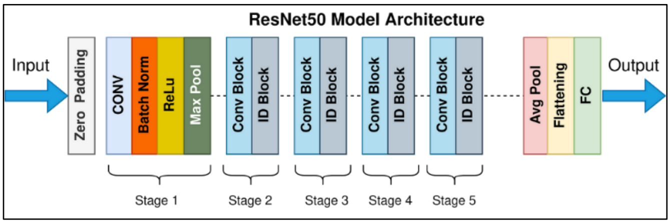


Fig. 4. ResNet50 Architecture [8]

2.3.3 InceptionV3

InceptionV3, developed by Google as an extension of GoogLeNet, employs a series of inception modules [9]. Each module consists of parallel convolutional layers whose outputs are concatenated at the end of the block. To reduce computational cost, the architecture incorporates dimensionality reduction using 1 × 1 convolutions before larger convolutions.

InceptionV3 also integrates an auxiliary classifier to stabilize training and reduce the vanishing gradient problem during backpropagation [10]. This auxiliary branch provides intermediate outputs used only in training to improve gradient flow, as illustrated in Figure 5.

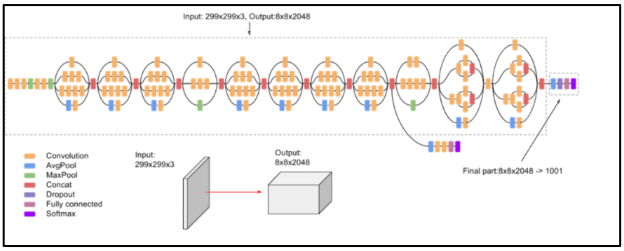


Fig. 5. InceptionV3 Architecture

2.3.4 Hyperparameter Tuning

Each model was trained twice to compare the performance of transfer learning against training with the original architecture. Additional layers were incorporated into the models, including a global average pooling or flatten layer, dropout layers, a dense layer with ReLU activation, and a final dense layer with 15 units using softmax activation. The values for the dropout rate and the dense layer configuration with ReLU activation were determined through hyperparameter tuning. The values and parameters used for hyperparameter tuning are presented in Table 1.

Table 1. Hyperparameter tuning parameters and values

Parameters	**Values**
Units in *dense layer* (ReLU)	128–512 units and 32–128 units
Dropout layer rate	0.0–0.6
Learning rate	Selected from 0.001, 0.0001, and 0.00001

During hyperparameter tuning, each model was trained for 25 epochs. To optimize the training process, callbacks were implemented, including an early stopping mechanism. The early stopping was activated from the 5th epoch with a patience value of 3 to prevent overfitting and unnecessary computations.

2.4 Ensemble Process

The ensemble process was performed after training the three models and selecting the best configurations from six hyperparameter tuning searches. Each model generated a probability array for all classes, followed by a thresholding step to ensure the validity of predictions. Majority voting is an ensemble process that determines the final prediction based on the highest number of votes, where each vote corresponds to the output of an individual model that has been trained beforehand [11]. The final class, which is the result of the classification, is determined by selecting the class label with the highest number of valid votes. The workflow of the majority voting ensemble process is illustrated in Figure 6.

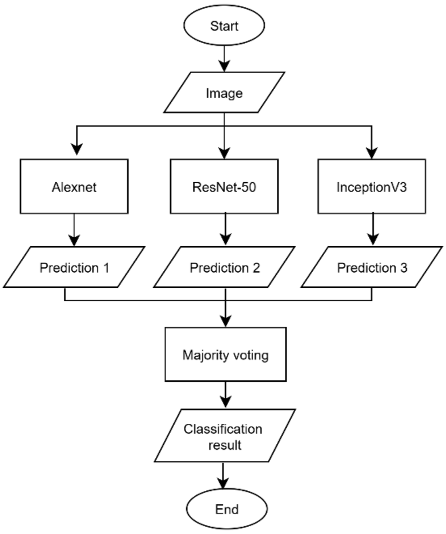


Fig. 6. Ensemble Workflow Design

2.5 Model Evaluation

In the evaluation process, evaluation data is obtained after each model is trained. The evaluation method used is to examine the accuracy, precision, recall, F1-score, and loss of each model.

2.6 Model Testing

Testing was conducted on a locally run website. The model deployment used the Flask library to create a local server. During the testing process, several code adjustments were made to ensure the ensemble model could perform classification.

3 Result and Discussion

The results show that for all models, the trainable layers with transfer learning (T) outperform training from scratch (NT), as shown in Table 2. Significant improvements were observed in the AlexNet model compared to the InceptionV3 and ResNet-50 models, with an increase in accuracy, precision, recall, and F1-score reaching 13%. Similarly, the loss value can be reduced significantly by using trainable models, especially the AlexNet model. The classification performance of all transfer learning models achieved excellent accuracy, precision, recall, and F1-score values, and the best was achieved by the ResNet-50 model.

Table 2. Comparison of Model Performance in Percentage (%)

Model	Layers	Accuracy	Precision	Recall	F1-*Score*	Loss
AlexNet	T	99.67	99.67	99.67	99.67	3.76
	NT	86.67	87.71	86.67	86.61	40.81
InceptionV3	T	99.83	99.84	99.83	99.83	3.21
	NT	96.17	96.44	96.17	96.14	9.91
ResNet-50	T	**100.00**	**100.00**	**100.00**	**100.00**	**1.92**
	NT	99.67	99.67	99.67	99.67	1.89

Table 3. Testing Scenarios Using Model Combination Variations

Combination	AlexNet	InceptionV3	ResNet-50
1	T	T	T
2	**T**	**T**	**NT**
3	T	NT	T
4	T	NT	NT
5	NT	T	T
6	NT	T	NT
7	NT	NT	T
8	NT	NT	NT

An evaluation of the ensemble process was conducted to determine whether this method successfully improved performance in image classification. The scenarios of model combinations for ensemble testing are presented in Table 3. These scenarios were designed to test various combinations to identify the most optimal one. The results of the ensemble testing are presented in Table 4.

Table 4. Evaluation Results from Model Combinations

Combination	*Accuracy*	*Precision*	*Recall*	*F1-score*	*Avg. Loss*	*Time (s)*
1	100.00 %	100.00 %	100.00 %	100.00 %	3.01%	44.67
2	**100.00 %**	**100.00 %**	**100.00 %**	**100.00 %**	**3.00%**	44.57
3	100.00 %	100.00 %	100.00 %	100.00 %	5.24%	44.97
4	100.00 %	100.00 %	100.00 %	100.00 %	5.23%	43.35
5	99.83%	99.84%	99.83%	99.83%	15.79%	**43.27**
6	99.83%	99.84%	99.83%	99.83%	15.78%	43.99
.	99.83%	99.84%	99.83%	99.83%	18.02%	45.62
8	99.83%	99.84%	99.83%	99.83%	18.01%	43.43

Based on Table 4, the second ensemble model combination achieved the best results with accuracy, resistance, recall, and F1 score of 100%. The best ensemble model consisted of AlexNet (T), InceptionV3 (T), and ResNet-50 (NT). The best testing time was achieved in the fifth combination, which took 43.27 seconds. However, the average loss across all models was 3.00%, indicating that each model could still produce prediction errors. This can be attributed to the implementation of majority voting to determine the final prediction, meaning at least two models successfully made correct predictions. Overall, the testing times of the combinations were relatively similar, averaging around ±44 seconds.

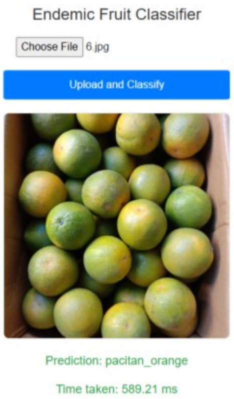
This study's performance was evaluated against that of other research that used the same dataset (Ubaya-IFDS3000 dataset). With k-NN and LDA, the system's performance results using the ensemble learning method yielded the highest accuracy of 97.80% [1]. In a related study, image classification accuracy was 98.03% using ensemble learning with MPEG-7 Visual Descriptors and Extreme Learning Machine [12]. Utilizing MPEG-7 Color and Texture Features Fusion and the Support Vector Machine classifier enhanced with the Grey Wolf Optimizer, research was able to attain the highest accuracy of 99.21% [13]. It can be concluded that this study using the ensemble CNN majority voting approach outperforms previous studies using the same data set, achieving the highest classification accuracy of 100%.

Table 5. Method Comparison with Previous Studies

Methods	Best Accuracy
Ensemble using k-NN and LDA [1]	97.80%
Ensemble using ELMs optimization [12]	98.03%
Enhanced SVM based on GWO using CS + SC [13]	99.21%
Ensemble using CNN model	**100.00%**

A basic web-based application was developed to implement the entire ensemble approach. Users can input an image to this application, and the three models are used to process it for prediction. The final prediction result is produced by applying the ensemble procedure with majority vote once all model predictions have been collected. As seen in Figure 7, the classification results are given beneath the preview of the submitted image.

The system's ability to achieve 100% accuracy could be attributed to the use of a controlled dataset, which is a further limitation of this study. Ensemble models can be trained and tested on a more diverse and randomized dataset of endemic fruits, minimizing the risk of overfitting. Furthermore, the method could be developed into a web application to make it easier to classify Indonesia's endemic fruits.


Endemic Fruit Classifier

6.jpg

Prediction: pacitan_orange

Time taken: 589.21 ms

Fig. 7. Testing of a basic web-based classification system

4 Conclusion

A classification model for Indonesian endemic fruits was successfully developed using an ensemble CNN majority voting approach. The retrained layers outperformed transfer learning, with AlexNet achieving 99.67% accuracy, InceptionV3 achieving 99.83% accuracy, and ResNet-50 achieving 100%. Furthermore, the ensemble model can be trained using a more diverse dataset of endemic fruits and integrated in a web application to further contribute to the classification of Indonesian endemic fruits.

References

1. J. Siswantoro, H. Arwoko, and M. Widiasri, Indonesian fruits classification from image using MPEG-7 descriptors and ensemble of simple classifiers, J. Food Process Eng., **43**, 7, (2020). https://doi.org/10.1111/jfpe.13414.
2. K. O'Shea and R. Nash, An Introduction to Convolutional Neural Networks, ArXiv e-prints, (2015).
3. R. Cohen and P. L. Cook, CHAPTER Conclusions, Eff. Mergers, 349–352, (2020). https://doi.org/10.4324/9781315016603-59.
4. M. Elgendi, M. U. Nasir, Q. Tang, D. Smith, J. P. Grenier, C. Batte, B. Spieler, W. D. Leslie, C. Menon, R. R. Fletcher, N. Howard, R. Ward, W. Parker, and S. Nicolaou, The Effectiveness of Image Augmentation in Deep Learning Networks for Detecting COVID-19: A Geometric Transformation Perspective. Frontiers in medicine, **8**, 629134, (2021). https://doi.org/10.3389/fmed.2021.629134
5. A. Krizhevsky, I, Sutskever, and G. E.Hinton, ImageNet classification with deep convolutional neural networks. Communications of the ACM, 60, 84 - 90, (2012).
6. X. Han, Y. Zhong, L. Cao, and L. Zhang, Pre-Trained AlexNet Architecture with Pyramid Pooling and Supervision for High Spatial Resolution Remote Sensing Image Scene Classification. Remote Sensing, **9(8)**, 848, (2017). https://doi.org/10.3390/rs9080848
7. K. He, X. Zhang, S. Ren and J. Sun, Deep Residual Learning for Image Recognition, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 770-778, (2016). https://doi.org/10.1109/CVPR.2016.90.
8. R. Gomes, & C. Kamrowski, J. Langlois, P. Rozario, I. Dircks, K. Grottodden, M. Martinez, W. Tee, K. Sargeant, C. LaFleur, and M. Haley, A Comprehensive Review of Machine Learning Used to Combat COVID-19, Diagnostics, **12,** 1853, (2022). https://doi.org/ 10.3390/diagnostics12081853.
9. O. Iparraguirre-Villanueva, V. Guevara-Ponce, O. Paredes, F. Sierra-Liñan, J. Zapata-Paulini, and M. Cabanillas-Carbonell, Convolutional Neural Networks with Transfer Learning for Pneumonia Detection, Int. J. Adv. Comput. Sci. Appl., **13**, Mar. (2022). https://doi.org/10.14569/IJACSA.2022.0130963.
10. A. M. Alhassan and N.I. Altmami, IV3TM: Inception V3 enabled bidirectional long short-term memory network for brain tumor classification. PloS one, **20** (10), e0335397, (2025). https://doi.org/10.1371/journal.pone.0335397
11. D. Patil and J. Patil, Malicious URLs Detection Using Decision Tree Classifiers and Majority Voting Technique, Cybern. Inf. Technol., 11–29, **18**, (2018). https://doi.org/10.2478/cait-2018-0002.
12. J. Siswantoro, H. Arwoko, and M. Siswantoro, Fruits Classification from Image using MPEG-7 Visual Descriptors and Extreme Learning Machine, 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI), 682-687, (2020). https://doi.org/10.1109/ISRITI51436.2020.9315523.

13. J. Siswantoro, Enhanced Support Vector Machine Based on Grey Wolf Optimizer for Fruits Image Classification using MPEG-7 Color and Texture Features Fusion, Int. J. Intell. Eng. Syst., **17**, 3, 1–11, (2024). https://doi.org/10.22266/ijies2024.0630.01.