Scientific Journal of Informatics
Vol. 11, No. 4, Nov 2024

p-ISSN 2407-7658 https://journal.unnes.ac.id/journals/sji/index e-ISSN 2460-0040

Optimization of Mango Plant Leaf Disease Classification Using Concatenation Method of MobileNetV2 and DenseNet201 CNN Architectures

Ahmad Ramadhan Auni[1*], Endang Sugiharti[2]
[1, 2]Department of Informatics Engineering, Universitas Negeri Semarang, Indonesia

Abstract.
Purpose: Mango production can be severely impacted by diseases affecting mango plants. By leveraging artificial intelligence, the agricultural sector can automate the analysis of mango leaves to monitor plant health. The goal of this research is to improve the early detection of diseases in mango leaves to allow early treatment to minimize damage to the crops.
Methods: This study employs an approach of combining two pre-trained CNN architectures, namely MobileNetV2 and DenseNet201 through concatenation method. To enhance the model's generalization ability, various image augmentation techniques were applied during the training phase.
Result: The model developed in this study achieved great performance in classifying mango leaf diseases with a testing accuracy of 99.25%. This result indicates the effectiveness of the concatenation method by outperforming the accuracy of either MobileNetV2 or DenseNet201 when implemented separately.
Novelty: This research introduces a novel strategy by concatenating two pre-trained CNN architectures for mango leaf disease classification, a method not previously explored in this context. The model developed from this study has the potential to serve as a tool for the early detection and treatment of mango leaf diseases.

Keywords: Leaf disease, Convolutional neural network, Concatenation, MobileNetV2, DenseNet201
Received October 2024 / **Revised** December 2024 / **Accepted** January 2025

INTRODUCTION

In this era, technological advancements are progressing at an unprecedented pace. These developments are making significant impacts across many sectors, including health, education, communication, and agriculture [1], [2], [3], [4]. Various innovations have been created to make human work in these fields more effective and efficient [5]. Examples of these innovations include disease diagnosis using artificial intelligence in health, online learning in education, social media that expands human communication access in communication, and plant disease detection in agriculture. In addition to detecting plant diseases, innovations in agriculture are increasingly leveraging artificial intelligence for various tasks, such as managing irrigation, analyzing soil content, and estimating crop yields [6], [7], [8].

Recent technological advancements in agriculture have simplified the work of farmers. In the past, farmers had to inspect each plant manually to identify any diseases. Now, this problem can be solved by technology that automates plant health checks [9]. This agricultural technology utilizes artificial intelligence, making plant health checks effective and efficient [10]. The task performed by artificial intelligence for automating plant health checks is called image classification [11].

Mango plant (*Mangifera indica* L.) is an economically significant fruit plant and a key commodity in the agricultural industry in various countries [12]. This is due to the many benefits of mango plants. Mango skin and leaves play an important role in traditional medicine. Countries such as India, Nigeria, and Sri Lanka utilize mangoes for their medicinal properties to address various health issues, including anemia, malaria, and diarrhea [13]. Additionally, mangoes are a rich source of carbohydrates, proteins, fats, fiber, and vitamins, all of which play a vital role in supporting human health and growth [14].

[*]Corresponding author.
Email addresses: aramadhan395@students.unnes.ac.id (Auni)[*], endangsugiharti@mail.unnes.ac.id (Sugiharti)
DOI: [10.15294/sji.v11i4.15169](https://doi.org/10.15294/sji.v11i4.15169)

All plants, including mango plants, can be susceptible to diseases that may result in crop failure. This can be very detrimental, making it important to identify diseases affecting mango plants to protect them. Analyzing the condition of the roots, stems, and leaves can detect symptoms of plant diseases. Mango plant diseases can be caused by climate changes or other factors. Currently, one way to determine if a mango plant is diseased is by analyzing leaf patterns using artificial intelligence [15], [16]. A common artificial intelligence implementation for this purpose is using Convolutional Neural Network (CNN) algorithms for mango leaf classification.

To classify plant leaf health, CNN is a widely used algorithm [17]. Some CNN architectures include InceptionV3, DenseNet201, MobileNetV2, ShuffleNetV2, EfficientNetB0 [18], [19], [20], [21], [22]. Various treatments can also be applied to the data and CNN architectures such as image augmentation, fine-tuning, transfer learning, and concatenation to achieve accurate results in plant leaf health classification [23], [24], [25]. The selection of CNN architecture and treatments depends on the characteristics of the data used [26].

MobileNetV2 is a CNN architecture specifically designed for devices with limited resources, like smartphones. It employs depthwise separable convolutions, which help to significantly lower the number of parameters and computational requirements while maintaining accuracy. Additionally, MobileNetV2 incorporates inverted residuals and linear bottlenecks that enhance the speed of the training process. This architecture proves to be highly efficient for various artificial intelligence tasks that involve image data, including image classification, object detection, and image segmentation [20].

DenseNet201 is a CNN architecture that employs dense connectivity, meaning each layer is directly linked to all the layers before it. This design enables DenseNet201 to utilize the features learned by previous layers which improve feature representation. With 201 layers, this architecture is recognized for generating richer and more detailed feature representations. DenseNet201 has demonstrated strong performance in various artificial intelligence tasks that involve image data, including image classification and object detection [19].

In CNN, concatenation refers to a method of merging features from various layers, resulting in a more complete representation. This technique can be applied across different CNN architectures to enhance their ability to represent plant leaf images effectively. As a result, concatenation has the potential to outperform individual CNN architectures in terms of performance [25].

Several studies have been carried out using artificial intelligence including the use in the agricultural sector. Research by Pandiyaraju et al. [27] used spatial attention-enabled ensemble classification using VGG16 and EfficientNetV2-B0 CNN architectures to classify mango leaf diseases, utilizing a dataset of 4,000 images with 500 images belonging to each class. The ensemble method used achieved higher performance than the VGG16 and EfficientNetV2-B0 that was implemented individually. The accuracy achieved using said model was 97.13%, which was higher when compared to several models, namely AlexNet, Artificial Neural Network, and VGG16. In another study, Mahbub et al. [28] used Lightweight Convolutional Neural Network (LCNN) model on a dataset also consisting of 4,000 images with 500 images belonging to each class. LCNN is specifically designed to reduce complexity while maintaining high accuracy, ideal for resource-constrained environments. The accuracy obtained using said model was 98%, outperforming several pre-trained CNN models, namely VGG16, ResNet50, ResNet101, and Xception.

Despite the impressive results achieved by previous studies, there is still potential for improvement in the accuracy of mango leaf disease classification. By using different CNN models compared to previous studies, there is a possibility of performance improvement. In particular, MobileNetV2 is known for its efficiency without degrading performance and DenseNet201 excels in capturing features through its dense connectivity structure. However, the integration of these architectures through concatenation method to combine their feature extraction capabilities has not been thoroughly studied. The approach could potentially lead to a more comprehensive feature representation which results in enhanced classification performance. Therefore, this research is focused on optimizing the accuracy of classifying mango leaf diseases by combining MobileNetV2 and DenseNet201 through concatenation method, contributing to the agricultural sector.

METHODS

This research was conducted to obtain accuracy result from the CNN model that applies concatenation method. The stages of this research are data collection, image preprocessing, data split, model implementation, model training, and model evaluation. The stages carried out in this study are presented in research flow in Figure 1.

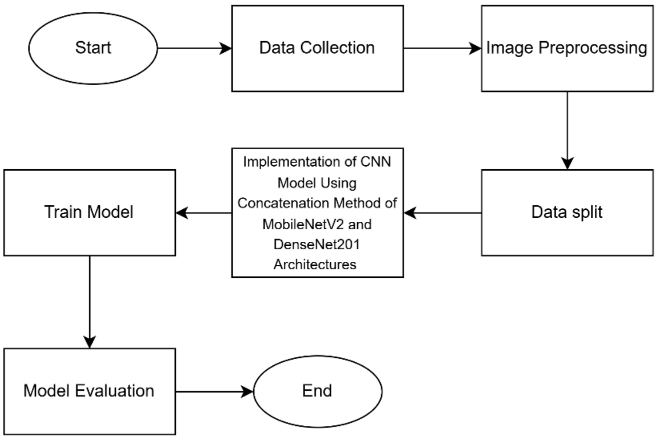

Figure 1. Research flow

Data collection

The dataset utilized in this study is the "Mango Leaf Disease Dataset" which is a public dataset obtained from the Kaggle platform. It contains images of mango plant leaves collected from Bangladesh. Initially, the dataset contained 1,800 images. However, the dataset underwent processes of image rotation and enlargement, resulting in a total of 4,000 images. The dataset is categorized into eight distinct classes, including Anthracnose, Bacterial Canker, Cutting Weevil, Die Back, Gall Midge, Healthy, Powdery Mildew, and Sooty Mould, with each class containing 500 images [29]. Sample images from the dataset can be seen in Figure 2.

Figure 2. Dataset sample

Image preprocessing

The dataset used will be going through preprocessing by using both image augmentation techniques and image resizing. Image augmentation is the process of modifying images and only applied on training data so that the model can recognize variety of images [23]. Image augmentation techniques used in this study include the following.

1. Rescale the image from [0,255] to [0,1].
2. Flip the image horizontally.
3. Flip the image vertically.
4. Rotate the image by 40°.

5. Shift the image horizontally by 20%.
6. Shift the image vertically by 20%.
7. Apply shear transformation to the image by 20%.
8. Zoom in on the image by 20%.
9. Fill empty pixels that appear after augmentation using nearest pixel value.

Additionally, the images will be resized from 240 x 320 pixels to 224 x 224 pixels to allow the model to train more efficiently.

Data split

Data collected from the dataset is split into 3 sets, namely training data, validation data, and testing data. The importance of data split lies in its ability to ensure that the model is trained effectively, validated to tune hyperparameters, and tested to evaluate performance on unseen data, to reduce overfitting and providing a reliable measure of model accuracy [30]. The ratio of data split for training, validation, and testing data is 80%:10%:10% respectively. This data split ratio is also used in research by Mahbub et al. [28] and achieved good result. The result of the data split is shown in Table 1.

Table 1. Result of data split

Training Data Percentage	Validation Data Percentage	Testing Data Percentage	Training Data Amount	Validation Data Amount	Testing Data Amount	Total
80%	10%	10%	3,200	400	400	4,000

Model implementation

The model used concatenation method of two pre-trained CNN architectures, namely MobileNetV2 and DenseNet201. CNN is a type of deep learning architecture designed for processing digital images and other two-dimensional grid data. The primary objective of CNN is to automatically identify features from input data through convolutional layers to allow the model to recognize complex structures and patterns in digital images. A typical CNN architecture consists of several key components, including convolutional layers, pooling layers, activation layers, and fully connected layers. The convolutional layer employs filters that traverse the input image to capture essential features such as edges and lines, Meanwhile, the pooling layer reduces the spatial dimension of the features extracted by the convolutional layer, which helps to minimize the number of parameters in the model. The activation layer introduces nonlinearity into the model. Finally, the fully connected layer serves as the final layer of the CNN, connecting the extracted features to specific classes or labels. Illustration of the CNN structure can be seen in Figure 3 [31].

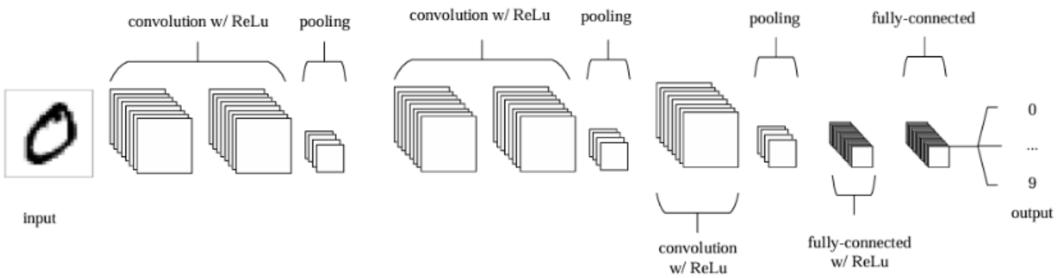

Figure 3. CNN structure [31]

MobileNetV2 is a pre-trained CNN architecture specifically designed for mobile devices and environments with limited resources. It employs depthwise separable convolutions which enhance efficiency and decrease the number of parameters. Additionally, it features innovations such as the inverted residual block and the linear bottleneck technique. These features enhance feature extraction and allow deeper layers without increasing the model's complexity. The computational cost of depthwise separable convolutions can be seen in Equation 1 [20].

$$Cost = h_i \,.\, \omega_i \,.\, d_i(k^2 + d_j) \tag{1}$$

In Equation 1, h_i denotes the height dimension of the feature map at layer i, while ω_i refers to the width dimension of the feature map at the same layer. Next, d_i indicates the number of input channels at layer i

and k represents the kernel size of the convolutional layer. Additionally, d_j represents the output channels at layer j. MobileNetV2 architecture can be seen in Figure 4.

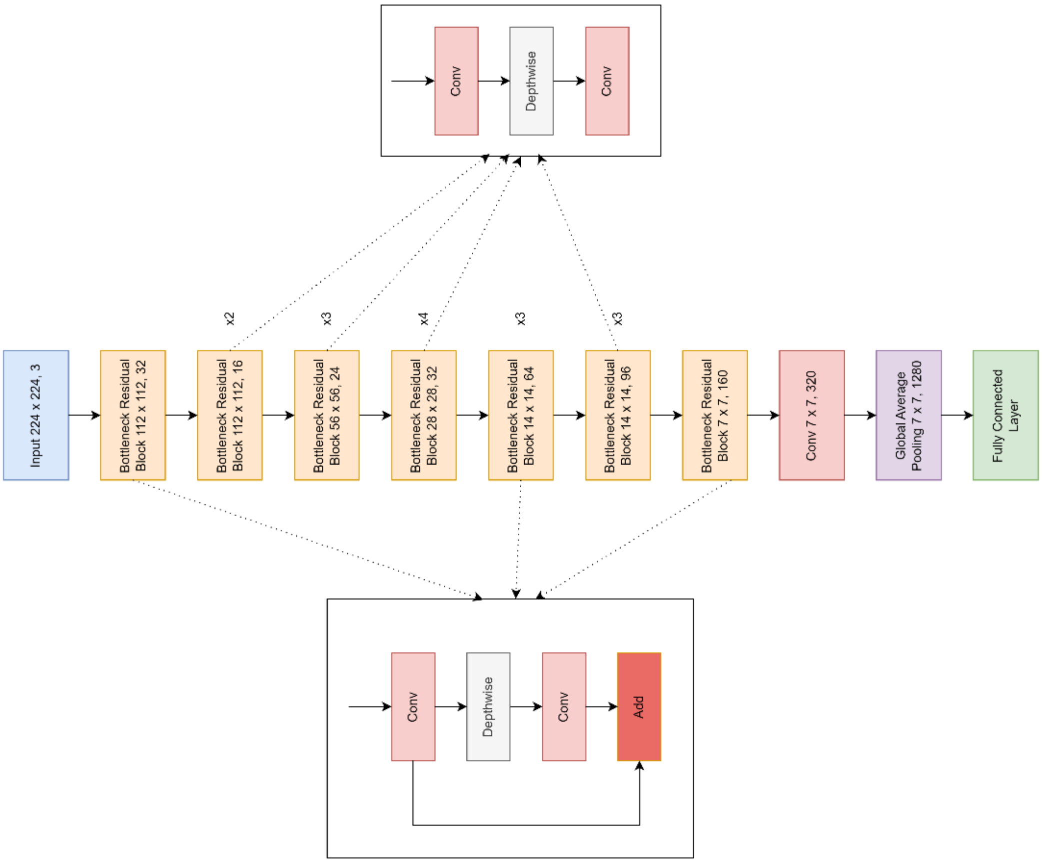

Figure 4. MobileNetV2 architecture

DenseNet201 is a pre-trained CNN architecture that is part of DenseNet recognized for its dense connectivity. In this architecture, each layer receives input from all preceding layers, improving the flow of information throughout the network. DenseNet201 consists of 201 layers and is particularly effective in tasks related to image processing. Its distinctive dense block structure ensures that inputs from any layer are directly connected to all subsequent layers, resulting in optimizing the model's performance. DenseNet201 architecture can be seen in Figure 5.

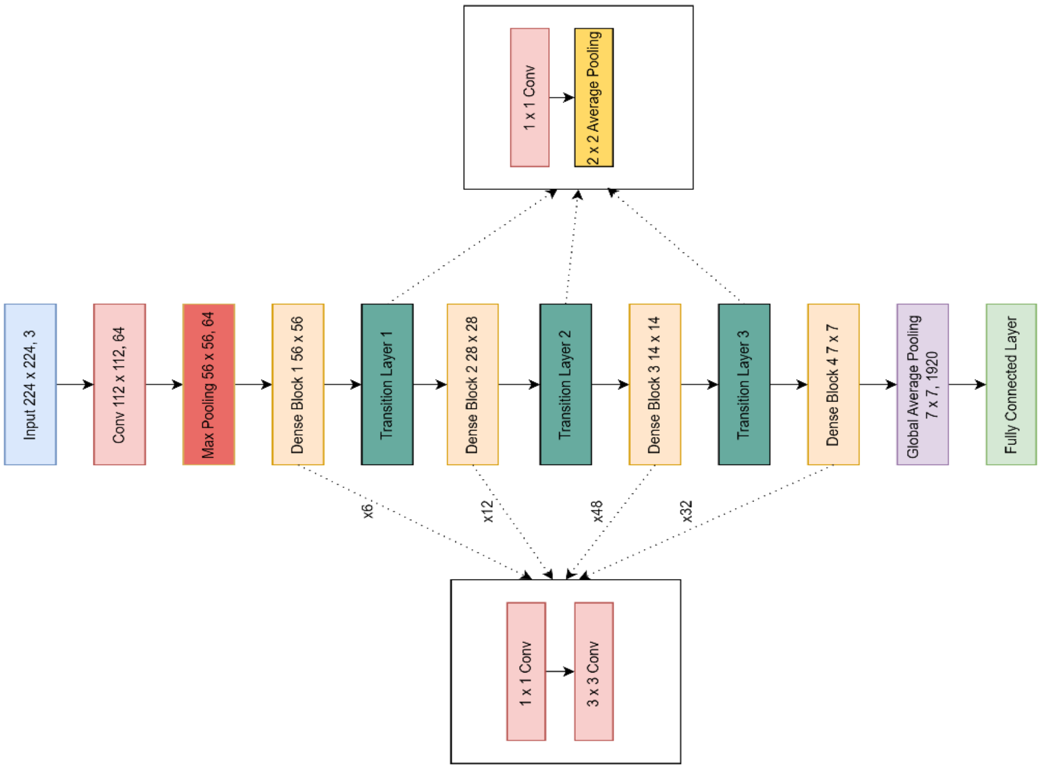

Figure 5. DenseNet201 architecture

In a dense block, the layers are directly connected, meaning that each layer within the block is connected to every subsequent layer. Each layer receives input from the features extracted by all the previous layers. This characteristic allows each layer in the dense block to have access to all the information generated by the preceding layers, enabling effective feature representation learning. The calculation of output in dense block can be seen in Equation 2 [19].

$$x_l = H_l([x_0, x_1, \dots, x_{l-1}]) \tag{2}$$

In Equation 2, x_l represents the output of layer l, H_l represents the operation applied to layer l, and $[x_0, x_1, \dots, x_{l-1}]$ represents the concatenation of the outputs of all previous layers in the block. The illustration of dense block 1 can be seen in Figure 6.

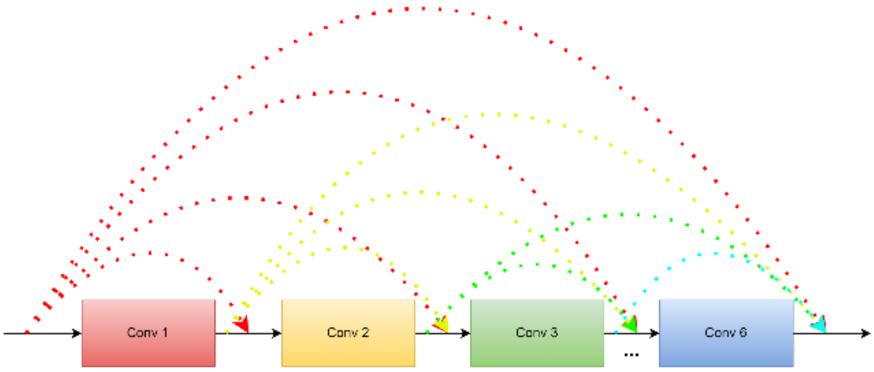

Figure 6. Dense block 1 illustration

Concatenation in neural networks combines outputs from multiple layers, forming a larger vector or matrix that incorporates information from all layers. This method enhances feature representation and model performance by integrating features from various layers. The general process includes defining input and processing layers, combining features using a concatenate layer, and passing the combined features to a fully connected layer for final prediction. Using the concatenation method for different CNN architectures has the potential to improve the model because it allows the model to leverage the strengths of multiple architectures simultaneously. By combining features from MobileNetV2, known for its efficiency, and DenseNet201, known for its dense connectivity, the model can capture a broader range of patterns within the data. This enables the model to benefit from both architectures. Consequently, this approach can enhance the model's accuracy. In this research, the steps for implementing concatenation method of MobileNetV2 and DenseNet201 are as follows.

1. Define the input layer.
2. Load the MobileNetV2 architecture and configure it to take input from the input layer.
3. Load the DenseNet201 architecture and configure it to take input from the input layer.
4. Use flatten layers to transform the outputs from both architectures into one-dimensional vectors.
5. Use a concatenate layer to merge the outputs processed by the flatten layer.
6. Apply a dense layer to prevent overfitting.
7. Apply an output layer that uses the softmax activation function to classify images into one of 8 different classes.

The pseudocode implementing concatenation method of MobileNetV2 and DenseNet201 in this research is shown below.

```
BEGIN
        SET n_classes = 8
        SET input_shape = (224, 224, 3)

        CREATE input_layer with shape = input_shape

        LOAD MobileNetV2 model with input = input_layer
        FREEZE all layers in MobileNetV2

        LOAD DenseNet201 model with input = input_layer
        FREEZE all layers in DenseNet201
        mobilenet_output = FLATTEN output of MobileNetV2
        densenet_output = FLATTEN output of DenseNet201

        concatenated_features = CONCATENATE mobilenet_output and
        densenet_output

        x = ADD Dense layer with 64 neurons
        output = ADD Output Dense layer with n_classes neurons

        RETURN output
END
```

The model implementing concatenation method of MobileNetV2 and DenseNet201 can be seen in Figure 7.

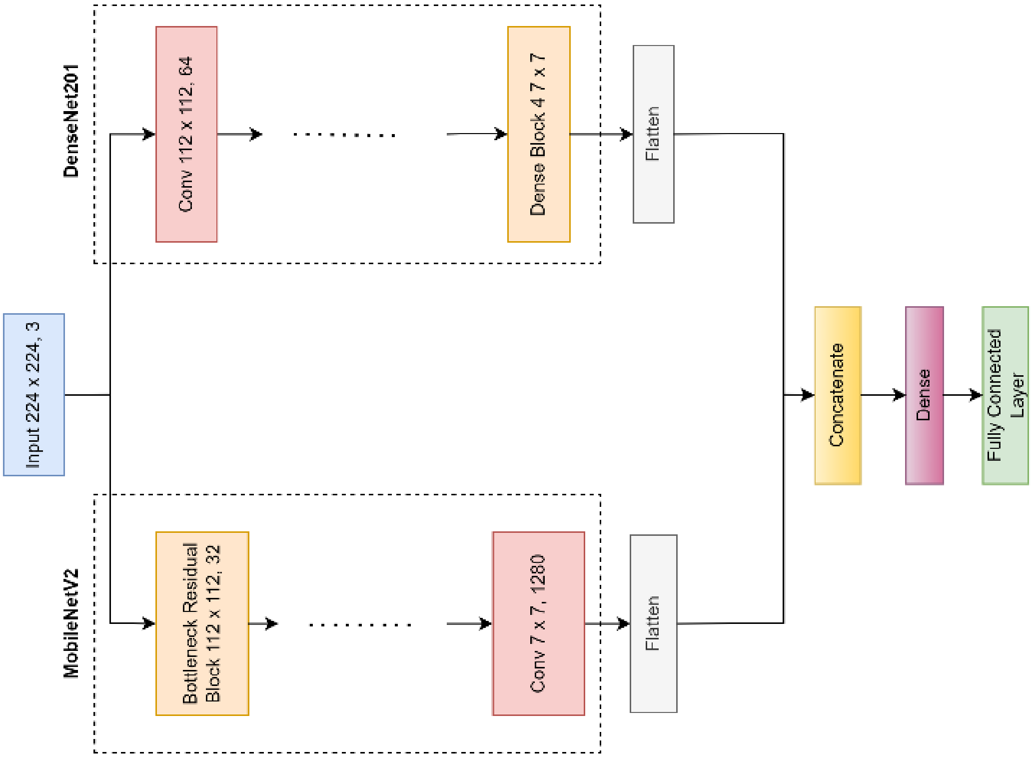

Figure 7. Model implementing concatenation method of MobileNetV2 and DenseNet201

Train model

The model will be trained using the Keras library by configuring some hyperparameters. Hyperparameters are crucial as they significantly influence the model's learning process and performance [32]. The hyperparameters used can be seen in Table 2.

Table 2. Hyperparameters

Hyperparameter	Value/Type
Optimizer	Adam
Loss function	Categorical crossentropy
Metrics	Accuracy
Epoch	50

Model evaluation

At this stage an evaluation is carried out from the previous stage, namely the testing stage. The authors use a confusion matrix that serves as foundation to calculate the accuracy of the model. Confusion matrix can be seen in Table 3 [33].

Table 3. Confusion matrix

Actual	Predicted	
	Positive (P)	Negative (N)
Positive (P)	TP	FN
Negative (N)	FP	TN

In Table 3, True Positive (TP) refers to cases where positive instances are correctly identified, while True Negative (TN) represents correctly identified negative instances. False Positive (FP) occurs when negative instances are mistakenly classified as positive and False Negative (FN) refers to positive instances incorrectly classified as negative. Accuracy is calculated as the ratio of correctly classified data points to the total number of data points in the test dataset which will be calculated through Equation 3 [34].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (3)$$

RESULTS AND DISCUSSIONS

The images used as training data were augmented and resized by using Keras library. Examples of images obtained by image augmentation techniques and resizing can be seen in Figure 8.

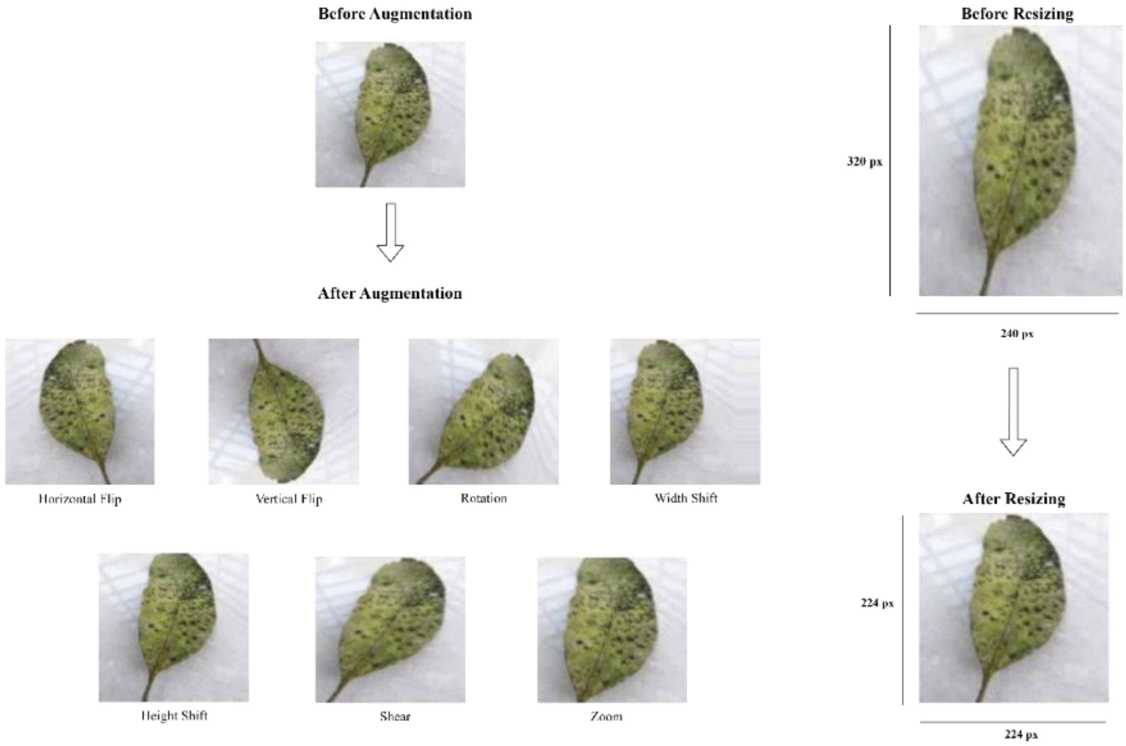

Figure 8. Image augmentation and resizing

After image augmentation and resizing, the CNN model will be trained. There are three models that will be trained. Each model uses the hyperparameters that has been mentioned in Table 2. The first model is built using MobileNetV2 while the second model utilizes DenseNet201. The third model combines both architectures by applying the concatenation method to merge the features of MobileNetV2 and DenseNet201. The models achieved by training can be evaluated by using testing data to get the accuracy of the models. Accuracy comparison of the results of testing for each model can be seen in Table 4.

Table 4. Accuracy comparison of models

Model	Accuracy
MobileNetV2	98%
DenseNet201	98.75%
Concatenation of MobileNetV2 and DenseNet201	99.25%

Table 4 shows that each model resulted in high accuracy result. However, the model that implemented concatenation method of MobileNetV2 and DenseNet201 resulted in higher accuracy than MobileNetV2 and DenseNet201 that were implemented separately. When the models were implemented separately, they struggled to capture the features of the data due to the limitations of each architecture. MobileNetV2, which was designed for efficiency, focuses on reducing computation with its layers, which can potentially compromise its ability to capture complex features. Meanwhile, DenseNet201, while effective at feature representation due to its dense connections, may not extract diverse features when implemented individually, especially in the context of a dataset that requires more intricate learning. By combining the outputs of both models through concatenation method, the model leverages the strengths of each architecture, enabling better generalization and feature capturing, which leads to improved accuracy.

Based on the research conducted in this study, the model achieved a high accuracy result and showed improvement compared to previous studies. Comparison of the accuracy obtained from this study with previous studies can be seen in Table 5.

Table 5. Comparison of accuracy result

Algorithm	Reference	Accuracy
Spatial Attention Enabled Ensemble Classification of VGG16 and EfficientNetV2-B0	[27]	97.13%
Lightweight CNN	[28]	98%
Proposed method (concatenation of MobileNetV2 and DenseNet201)		99.25%

Table 5 shows that the model developed in this study achieved higher accuracy result than previous studies by Pandiyaraju et al. [27] and Mahbub et al. [28]. Specifically, the model that used concatenation method of MobileNetV2 and DenseNet201 resulted in an accuracy that is 1.25% higher compared to Lightweight CNN used by Mahbub et al. [28].

CONCLUSION

In conclusion, using the CNN model that applies the concatenation method of MobileNetV2 and DenseNet201 architectures results in good performance for classifying mango leaf diseases. The model achieved a high testing accuracy of 99.25%, outperforming MobileNetV2 and DenseNet201 when implemented separately. This improvement in classification accuracy contributes to advancing agriculture by enabling the early detection and management of mango leaf diseases, which can minimize crop losses and enhance productivity. Moreover, the combination of MobileNetV2 and DenseNet201 through the concatenation method also highlights the potential for performance improvements by integrating CNN architectures, contributing to broader field of artificial intelligence and its applications.

REFERENCES

[1] Y.-C. Shen, M.-Y. Wang, and Y.-C. Yang, "Discovering the potential opportunities of scientific advancement and technological innovation: A case study of smart health monitoring technology," *Technol Forecast Soc Change*, vol. 160, p. 120225, Nov. 2020, doi: 10.1016/j.techfore.2020.120225.

[2] W. Ng, *New Digital Technology in Education*. Cham: Springer International Publishing, 2015. doi: 10.1007/978-3-319-05822-1.

[3] I. Diaz, D. S. Chiaburu, R. D. Zimmerman, and W. R. Boswell, "Communication technology: Pros and cons of constant connection to work," *J Vocat Behav*, vol. 80, no. 2, pp. 500–508, Apr. 2012, doi: 10.1016/j.jvb.2011.08.007.

[4] F. da Silveira, F. H. Lermen, and F. G. Amaral, "An overview of agriculture 4.0 development: Systematic review of descriptions, technologies, barriers, advantages, and disadvantages," *Comput Electron Agric*, vol. 189, p. 106405, Oct. 2021, doi: 10.1016/j.compag.2021.106405.

[5] A. Bergek, M. Hekkert, S. Jacobsson, J. Markard, B. Sandén, and B. Truffer, "Technological innovation systems in contexts: Conceptualizing contextual structures and interaction dynamics," *Environ Innov Soc Transit*, vol. 16, pp. 51–64, Sep. 2015, doi: 10.1016/j.eist.2015.07.003.

[6] T. Talaviya, D. Shah, N. Patel, H. Yagnik, and M. Shah, "Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides," Jan. 01, 2020, *KeAi Communications Co.* doi: 10.1016/j.aiia.2020.04.002.

[7] R. Khan, N. Dhingra, and N. Bhati, "Role of Artificial Intelligence in Agriculture: A Comparative Study," in *Transforming Management with AI, Big-Data, and IoT*, Cham: Springer International Publishing, 2022, pp. 73–83. doi: 10.1007/978-3-030-86749-2_4.

[8] N. N. Misra, Y. Dixit, A. Al-Mallahi, M. S. Bhullar, R. Upadhyay, and A. Martynenko, "IoT, Big Data, and Artificial Intelligence in Agriculture and Food Industry," *IEEE Internet Things J*, vol. 9, no. 9, pp. 6305–6324, May 2022, doi: 10.1109/JIOT.2020.2998584.

[9] A. Rajbongshi, T. Khan, M. M. Rahman, A. Pramanik, S. M. T. Siddiquee, and N. R. Chakraborty, "Recognition of mango leaf disease using convolutional neural network models: A transfer learning approach," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 23, no. 3, pp. 1681–1688, Sep. 2021, doi: 10.11591/ijeecs.v23.i3.pp1681-1688.

[10] N. C. Eli-Chukwu, "Applications of Artificial Intelligence in Agriculture: A Review," 2019. [Online]. Available: www.etasr.com

[11] Dr. S. Shakya, “Analysis of Artificial Intelligence based Image Classification Techniques,” *Journal of Innovative Image Processing*, vol. 2, no. 1, pp. 44–54, Mar. 2020, doi: 10.36548/jiip.2020.1.005.
[12] S. K. Mitra, “Mango production in the world – present situation and future prospect,” *Acta Hortic*, no. 1111, pp. 287–296, Feb. 2016, doi: 10.17660/ActaHortic.2016.1111.41.
[13] S. Z. Hussain, B. Naseer, T. Qadri, T. Fatima, and T. A. Bhat, “Mango (Mangifera Indica)-Morphology, Taxonomy, Composition and Health Benefits,” in *Fruits Grown in Highland Regions of the Himalayas*, Cham: Springer International Publishing, 2021, pp. 245–255. doi: 10.1007/978-3-030-75502-7_19.
[14] M. H. A. Jahurul *et al.*, “Mango (Mangifera indica L.) by-products and their valuable components: A review,” Sep. 15, 2015, *Elsevier Ltd*. doi: 10.1016/j.foodchem.2015.03.046.
[15] L. Li, S. Zhang, and B. Wang, “Plant Disease Detection and Classification by Deep Learning—A Review,” *IEEE Access*, vol. 9, pp. 56683–56698, 2021, doi: 10.1109/ACCESS.2021.3069646.
[16] R. Saleem, J. H. Shah, M. Sharif, and G. J. Ansari, “Mango Leaf Disease Identification Using Fully Resolution Convolutional Network,” *Computers, Materials and Continua*, vol. 69, no. 3, pp. 3581–3601, 2021, doi: 10.32604/cmc.2021.017700.
[17] G. Shrestha, Deepsikha, M. Das, and N. Dey, “Plant Disease Detection Using CNN,” in *2020 IEEE Applied Signal Processing Conference (ASPCON)*, IEEE, Oct. 2020, pp. 109–113. doi: 10.1109/ASPCON49795.2020.9276722.
[18] A. Demir, F. Yilmaz, and O. Kose, “Early detection of skin cancer using deep learning architectures: resnet-101 and inception-v3,” in *2019 Medical Technologies Congress (TIPTEKNO)*, IEEE, Oct. 2019, pp. 1–4. doi: 10.1109/TIPTEKNO47231.2019.8972045.
[19] G. Huang, Z. Liu, G. Pleiss, L. van der Maaten, and K. Q. Weinberger, “Convolutional Networks with Dense Connectivity,” *IEEE Trans Pattern Anal Mach Intell*, vol. 44, no. 12, pp. 8704–8716, Dec. 2022, doi: 10.1109/TPAMI.2019.2918284.
[20] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “Mobilenetv2 Inverted residuals and linear bottlenecks,” *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4510–4520, 2018.
[21] V. K. Gunjan, S. Kumar, M. D. Ansari, and Y. Vijayalata, “Prediction of Agriculture Yields Using Machine Learning Algorithms,” 2022, pp. 17–26. doi: 10.1007/978-981-16-6407-6_2.
[22] V. Y, N. Billakanti, K. Veeravalli, A. D. R. N, and L. Kota, “Early Detection of Casava Plant Leaf Diseases using EfficientNet-B0,” in *2022 IEEE Delhi Section Conference (DELCON)*, IEEE, Feb. 2022, pp. 1–5. doi: 10.1109/DELCON54057.2022.9753210.
[23] A. Mikolajczyk and M. Grochowski, “Data augmentation for improving deep learning in image classification problem,” in *2018 International Interdisciplinary PhD Workshop (IIPhDW)*, IEEE, May 2018, pp. 117–122. doi: 10.1109/IIPHDW.2018.8388338.
[24] F. Zhuang *et al.*, “A Comprehensive Survey on Transfer Learning,” *Proceedings of the IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021, doi: 10.1109/JPROC.2020.3004555.
[25] N. Noreen, S. Palaniappan, A. Qayyum, I. Ahmad, M. Imran, and M. Shoaib, “A Deep Learning Model Based on Concatenation Approach for the Diagnosis of Brain Tumor,” *IEEE Access*, vol. 8, pp. 55135–55144, 2020, doi: 10.1109/ACCESS.2020.2978629.
[26] L. D. Nguyen, R. Gao, D. Lin, and Z. Lin, “Biomedical image classification based on a feature concatenation and ensemble of deep CNNs,” *J Ambient Intell Humaniz Comput*, vol. 14, no. 11, pp. 15455–15467, Nov. 2023, doi: 10.1007/s12652-019-01276-4.
[27] V. Pandiyaraju, S. Venkatraman, A. Abeshek, S. A. Aravintakshan, and K. S. Pavan, “Mango Leaf Disease Detection and Classification Using Spatial Attention Enabled Ensemble Classification,” in *2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS)*, IEEE, Apr. 2024, pp. 1–8. doi: 10.1109/ADICS58448.2024.10533513.
[28] N. I. Mahbub, F. Naznin, M. I. Hasan, S. M. R. Shifat, M. A. Hossain, and M. Z. Islam, “Detect Bangladeshi Mango Leaf Diseases Using Lightweight Convolutional Neural Network,” in *3rd International Conference on Electrical, Computer and Communication Engineering, ECCE 2023*, Institute of Electrical and Electronics Engineers Inc., 2023. doi: 10.1109/ECCE57851.2023.10101648.
[29] S. I. Ahmed *et al.*, “MangoLeafBD: A comprehensive image dataset to classify diseased and healthy mango leaves,” *Data Brief*, vol. 47, p. 108941, Apr. 2023, doi: 10.1016/j.dib.2023.108941.
[30] V. R. Joseph, “Optimal ratio for data splitting,” *Statistical Analysis and Data Mining: The ASA Data Science Journal*, vol. 15, no. 4, pp. 531–538, Aug. 2022, doi: 10.1002/sam.11583.
[31] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” Nov. 2015, [Online]. Available: http://arxiv.org/abs/1511.08458

[32] F. Hutter, J. Lücke, and L. Schmidt-Thieme, “Beyond Manual Tuning of Hyperparameters,” *KI - Künstliche Intelligenz*, vol. 29, no. 4, pp. 329–337, Nov. 2015, doi: 10.1007/s13218-015-0381-0.
[33] O. Caelen, “A Bayesian interpretation of the confusion matrix,” *Ann Math Artif Intell*, vol. 81, no. 3–4, pp. 429–450, Dec. 2017, doi: 10.1007/s10472-017-9564-8.
[34] B. H. Reddy and P. R. Karthikeyan, “Classification of Fire and Smoke Images using Decision Tree Algorithm in Comparison with Logistic Regression to Measure Accuracy, Precision, Recall, F-score,” in *14th International Conference on Mathematics, Actuarial Science, Computer Science and Statistics, MACS 2022*, Institute of Electrical and Electronics Engineers Inc., 2022. doi: 10.1109/MACS56771.2022.10022449.