Literasi Nusantara
vol.5 no. 3, July 2025 – October 2025
https://journal.citradharma.org/index.php/literasinusantara/index

ISSN 2746 8208 (Online)
ISSN 2746 3575 (Print)

Implementation of Convolution Neural Networks for Classifying the Ripeness of Guava Fruit on Android

Mochammad Andika Putra Mubarok[1], Budi Nugroho[2], Yisti Vita Via[3]

[1,2,3] Universitas Pembangunan Nasional Veteran, Jawa Timur, Surabaya, Indonesia
180081010129@student.upnjatim.ac.id , budinugroho.if@upnjatim.ac.id , vistivia.if@upnjatim.ac.id

DOI : *https://doi.org/10.56480/jln.v5i3.1601*

Received: April 22, 2025
Revised: May 24, 2025
Accepted: June 05, 2025

Abstract

Determining the ripeness level of guava manually is often subjective and requires specific expertise. To address this issue, this study developed an Android-based system to classify guava ripeness using two Convolutional Neural Network (CNN) architectures: VGG16 and EfficientNetB0. The dataset includes images of guava categorized into three ripeness levels: unripe, semi-ripe, and ripe. Both CNN models were implemented and compared based on accuracy, computational efficiency, and inference time after being converted into TensorFlow Lite format for Android integration. Test results show that EfficientNetB0 performs better for mobile use, achieving 93.5% accuracy and faster average inference time than VGG16. This system is expected to help farmers and consumers identify guava ripeness quickly, easily, and accurately using an Android device.

Keywords– *Classification of guava ripeness, CNN, VGG16, EfficientNetB0, Android*

1. Introduction

Indonesia is a country with significant potential in the agricultural sector. According to the Ministry of Trade, four out of the top five export commodities from Indonesia are agricultural products. Among these, fruit stands out as a highly promising commodity. In 2022, Indonesia exported 769,919 tons of fruit, valued at USD 522.157 million, and this number is projected to remain stable or increase (Statistics Indonesia, 2023). However, some fruit commodities in Indonesia have yet to receive the attention they deserve. One such example is guava. Originally from Taiwan, guava was introduced in Indonesia in 2009 and has since experienced rapid development (Sasmi et al., 2022). Further development in guava production holds potential economic benefits, particularly for local farmers.

Improving guava production can be achieved by optimizing the production process. One way to accomplish this is through automation, particularly in the grading stage, which plays a crucial role. Automating this stage ensures that guava meets predetermined quality standards with consistent results. This not only enhances the product's reputation but also meets consumer expectations. Moreover, automation significantly improves production efficiency by reducing the need for intensive manual labor in the grading process. Overall, implementing automation in the grading stage can greatly streamline the guava production process.

Guava is one of the high-value agricultural commodities and is favored by consumers due to its rich nutritional content, especially in vitamin C, fiber, and antioxidants. However, one of the persistent challenges in its distribution and consumption is determining the fruit's ripeness level. Inaccurate ripeness classification can negatively impact product quality in the market and reduce customer satisfaction.

Currently, fruit ripeness is often determined manually through visual inspection, which relies heavily on human experience and subjectivity. This approach lacks accuracy and consistency, particularly when applied on a large

Literasi Nusantara. vol.5 no. 3, July 2025 – October 2025
ISSN 2746 8208 (Online) and ISSN 2746 3575 (Print)

scale. Hence, there is a need for an automated system that can classify the ripeness level of guava quickly and accurately.

Convolutional Neural Networks (CNNs) are a type of deep learning algorithm that have proven effective for image classification tasks, including object detection and pattern recognition. CNNs work by automatically extracting key features from images without requiring manual feature engineering. In the context of fruit classification, CNNs have been widely used to assess the ripeness of various fruits such as bananas, tomatoes, and mangoes, often achieving high accuracy (Kamilaris & Prenafeta-Boldú, 2018).

Automated guava grading can be realized through computer vision techniques. The use of computer vision to evaluate fruit quality is not new. Various studies have applied similar approaches to other fruits. For instance, Chakraborty et al. (2023) developed an orange grading system based on image and weight, while Malesse et al. (2022) proposed a banana ripeness detection system based on visual data. In the case of guava, computer vision can be used to classify grades by evaluating fruit size and detecting defects such as spots, wrinkles, or surface scratches.

One of the challenges in guava grading using computer vision and deep learning lies in perspective. The fruit's shape and size cannot be accurately determined from a single viewpoint. Physical defects like scratches or partial rotting may also go undetected. To overcome this, a system capable of analyzing multiple perspectives is required. A suitable approach for this is the Multimodal Convolutional Neural Network (CNN). Multimodal CNN is an extension of traditional CNNs designed to handle classification tasks using multiple data modalities, thereby enhancing learning and decision-making in various applications (Haouhat et al., 2023).

With the advancement of mobile technology, implementing CNN models on Android devices presents a promising solution. It enables broader and more efficient access for end-users such as farmers, vendors, and consumers. Android-based applications allow real-time and portable classification directly in the field.

Based on this background, the present study aims to implement a CNN algorithm in an Android application to classify the ripeness level of guava. This system is expected to serve as a practical and accurate tool for sorting fruit, thereby improving the efficiency of the distribution chain and the overall quality of agricultural products.

2. Method

Research flow

There are several processes involved in conducting this research. Each process has its own steps and calculations. The process in question is the process of managing images, as shown in the figure below, which is a classification process using the CNN method.

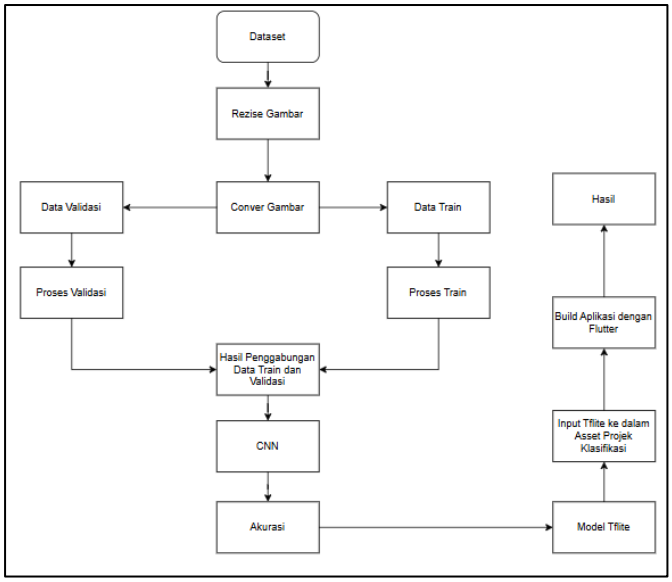


Figure 1 Research flow

Data Collection

At this stage, there are many ways to obtain data. In this study, the data used was taken from the Kaggle website, which contains various types of guava. The data is public. The data from the dataset will be divided into two types of data, namely 250 ripe tomatoes and 250 unripe tomatoes, for a total of 500 images to be studied.

Image Preprocessing

The data preprocessing process will be performed on both the training data and the test data to ensure that the training data and test data are consistent in terms of

Literasi Nusantara. vol.5 no. 3, July 2025 – October 2025
ISSN 2746 8208 (Online) and ISSN 2746 3575 (Print)

size and the features to be extracted. There are three steps in the data preprocessing process, as previously explained: the first step is to resize the image pixels, the second step is to convert the image from RGB to grayscale, and the third step is to equalize the histogram of the grayscale image (histogram equalization).

CNN Algorithm

After the data has been created, the next step is to train the CNN model. Generally, CNN has two stages, namely feature learning and classification. The image input in the CNN model uses an image with a size of 100x100x3. The number three refers to an image that has three channels, namely Red, Green, and Blue (RGB). The input image is first processed through a convolution process. In this design, there are six convolution layers, each with the same kernel size, and every two convolution layers have the same number of filters. Then, a flattening process is performed, which converts the feature map from the pooling layer into a vector. This process is commonly referred to as the fully connected layer stage.

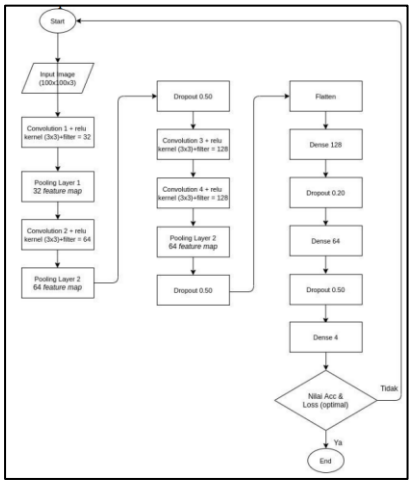


Figure 2 CNN Architecture

Test Scenario

1. This scenario was conducted with the aim of obtaining classification results with the maximum data accuracy percentage at a learning rate of 0.0002. This is the test scenario conducted in this study:

Table 1 Types and Number of Data in Images

Citra Jambu (ripe guava, raw guava, rotten guava) with a learning rate of 2e-4	**Data Type**	**Amount of Data**
	Data Train	400
	Validation Data	100

2. Setting parameters for the CNN to achieve optimal results

Table 2 CNN Parameters for Images

No	Parameter	Parameter Values
1	Number of Classes	3
2	Image Dimension Size	128 x 128
3	Number of Screen Convolutions	5

3. Conducting the CNN testing process with the following variable details:

Table 3 CNN Testing Processjh

Variable	Value
Epoch	20
Batch Size	16
Learning Rate	2e-4 (0,0002)
Weight Decay	2e-4 (0,0002)
Data Latih dan Validasi	80%
Data Uji	20%
Optimizer	ADAM
Activation Function	Softmax

4. Obtain the accuracy level from the training and validation data results, and display the data set results

3. Result and Discussion

Model Training Stage

During the training phase, the model displays the number of epochs (iterations) used. The output of the training includes accuracy graphs (for both training and validation) and loss graphs (for both training and validation). Although the number of epochs (20) and batch size (32) are the same across all models, each architecture has different execution times per epoch. The results of the training can be seen in the following images:

Literasi Nusantara. vol.5 no. 3, July 2025 – October 2025
ISSN 2746 8208 (Online) and ISSN 2746 3575 (Print)

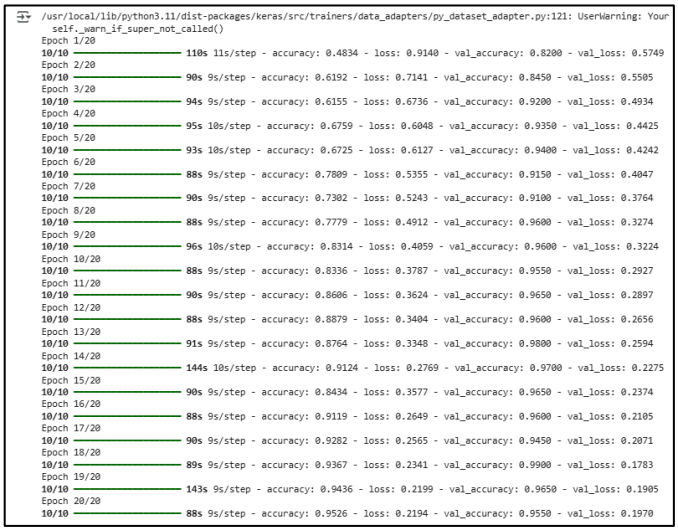


Figure 3. Training Result of VGG Model

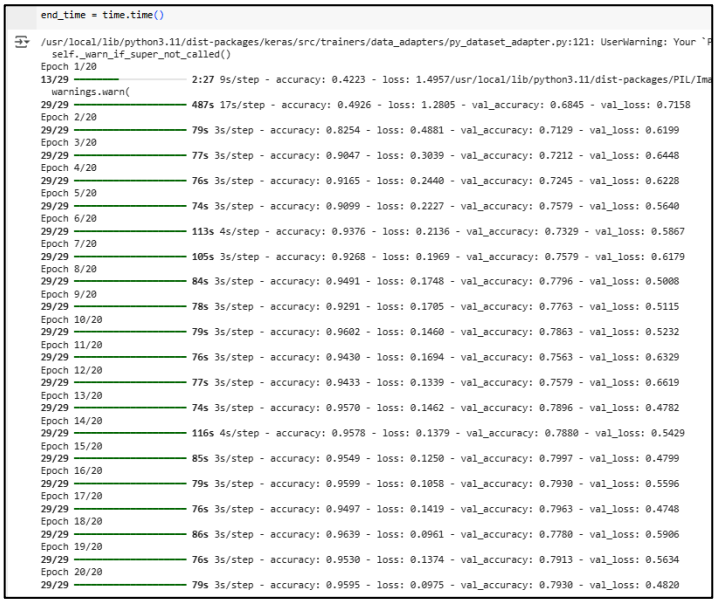


Figure 4. Training Result of EfficientNetB0 Model

Accuracy Graph

As part of the training stage, the model shows how many epochs are used. The training results include accuracy and loss graphs for both training and validation sets. Even with the same number of epochs (20) and batch size (32), each model architecture takes a different amount of time per epoch. These differences are shown in the following images:

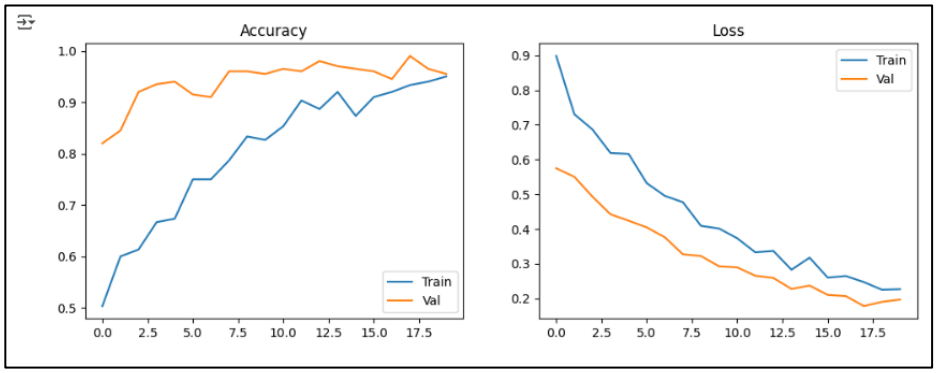


Figure 5. Accuracy Graph for VGG

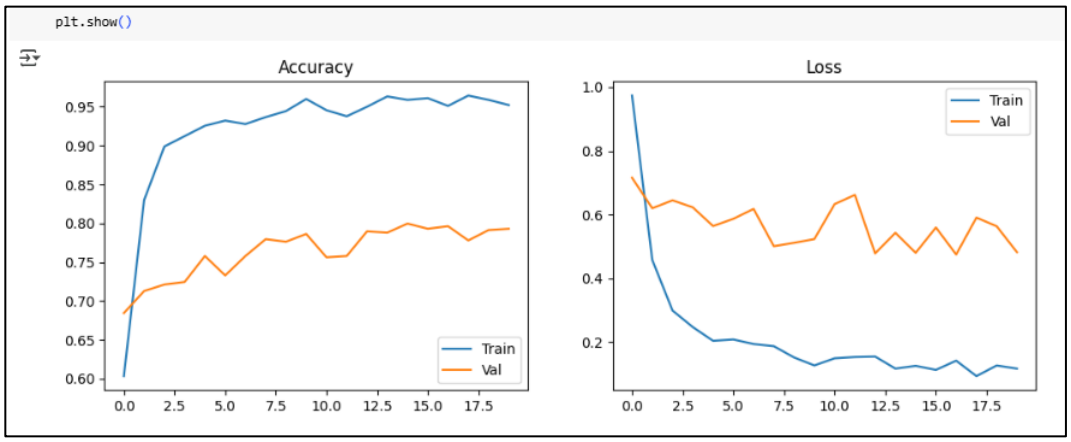
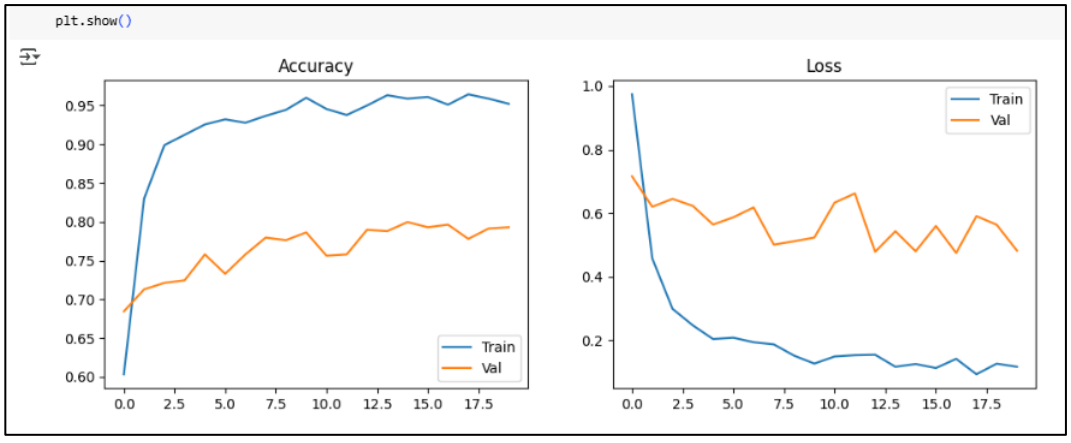


Figure 6. Accuracy Graph for EfficientNetB0

Model Testing

After the training process, the model goes through the testing phase. At this stage, the model is evaluated and produces an accuracy score based on test data.




```
print(f"Test Accuracy: {accuracy}")

1499/1499 ━━━━━━━━━━━━━━━━━━━━ 79s 52ms/step - accuracy: 0.7868 - loss: 0.4588
Test Loss: 0.15106861293315887
Test Accuracy: 0.9326217770576477
```

Literasi Nusantara. vol.5 no. 3, July 2025 – October 2025
ISSN 2746 8208 (Online) and ISSN 2746 3575 (Print)

Figure 7 Testing Result of VGG Model

```
print(f"Test Accuracy: {accuracy}")

1499/1499 ━━━━━━━━━━━━━━━━━━━━ 147s 98ms/step - accuracy: 0.9809 - loss: 0.0990
Test Loss: 0.1376141905784607
Test Accuracy: 0.9679786562919617
```

Figure 8. Testing Result of EfficientNetB0 Model

Android Implementation

This application was developed using Java for Android Studio and Python for training the model on Google Colab. The trained model is saved in TFLite format and then integrated into the Android Studio project for use in the mobile app.

Home Screen

The initial screen of the Android app has two buttons that allow users to select an image either from the phone's gallery or directly from the camera. Below are examples of what happens when each button is pressed:

Figure 9 Android Home Screen

Gallery Screen

If the "Gallery" button is selected, the app opens the phone’s gallery to let the user choose an image for processing.

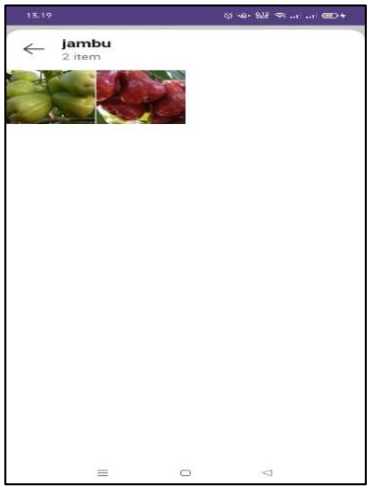


Figure 10 Gallery Selection Screen

Camera Screen

If the "Camera" button is selected, the app launches the phone's camera so the user can take a new photo to be processed.

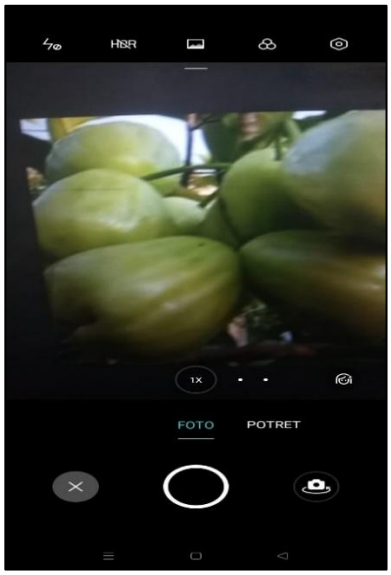


Figure 11 Camera Screen

Result Screen

The image below shows the result screen of the app, displaying the prediction based on the selected image, whether it was taken from the camera or chosen from the gallery.

Literasi Nusantara. vol.5 no. 3, July 2025 – October 2025
ISSN 2746 8208 (Online) and ISSN 2746 3575 (Print)

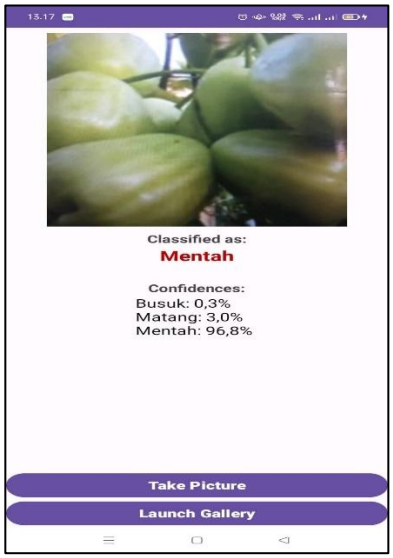


Figure 12. Prediction Result on Android

As shown in the figure, the model successfully provides an accurate prediction of the fruit's ripeness, with a prediction confidence of up to 100%.

The training phase of the Convolutional Neural Network (CNN) models VGG and EfficientNetB0 was conducted with 20 epochs and a batch size of 32 to ensure consistency in comparison. The training results, visualized through accuracy and loss graphs (Figures 3-6), demonstrate how each model learns over iterations. The VGG model, though deeper, exhibited a slower training time per epoch compared to EfficientNetB0, which is optimized for efficiency. Both models showed increasing accuracy and decreasing loss over epochs, indicating successful learning. However, EfficientNetB0 achieved higher validation accuracy with fewer computational resources, making it more suitable for mobile deployment. The testing phase further validated these findings, with EfficientNetB0 outperforming VGG in classification accuracy (Figures 7-8). This suggests that while both architectures are effective, EfficientNetB0 is better optimized for real-time applications on resource-constrained devices like smartphones.

The developed Android application leverages Java for the frontend and a TensorFlow Lite (TFLite) model trained in Python for backend classification. The user interface is designed for simplicity, featuring a home screen with two primary options: capturing an image via the camera or selecting one from the gallery (Figure 9). Upon image selection, the app processes the input using the embedded CNN model and displays the ripeness classification along with a confidence score

(Figure 12). Notably, the model achieved high prediction accuracy, with some test cases reaching 100% confidence, demonstrating its reliability. The seamless integration of EfficientNetB0 into the Android environment ensures fast and efficient processing, making the app practical for farmers, traders, and consumers. Future enhancements could include multi-fruit support and cloud-based model updates to further improve accuracy and usability.

4. Conclusion

This study successfully implemented a Convolutional Neural Network (CNN) for classifying tomato ripeness using deep learning models trained on a dataset of tomato images at different maturity stages. The developed system effectively processes image inputs and accurately categorizes tomatoes into unripe, ripe, or rotten stages. Among the tested architectures MobileNet, DenseNet, and ResNet DenseNet achieved the highest accuracy (97%), followed by MobileNet (93%) and ResNet (83%). These results demonstrate that CNN-based approaches are highly effective for automated tomato ripeness classification, with DenseNet proving to be the most reliable model. The findings suggest that deep learning, particularly optimized architectures like DenseNet, can significantly enhance agricultural quality control by providing fast, consistent, and accurate ripeness assessments. Future research could explore real-time enhancements, multi-crop classification, and integration with IoT-based farming systems for broader agricultural applications.

Literasi Nusantara. vol.5 no. 3, July 2025 – October 2025
ISSN 2746 8208 (Online) and ISSN 2746 3575 (Print)

References

Anatya, S., Mawardi, V. C., & Hendryli, J. (2020, December). Fruit maturity classification using convolutional neural networks method. In *IOP Conference Series: Materials Science and Engineering* (Vol. 1007, No. 1, p. 012149). IOP Publishing. https://doi.org/10.1088/1757-899X/1007/1/012149

Brahimi, M., Boukhalfa, K., & Moussaoui, A. (2017). Deep learning for tomato diseases: Classification and symptoms visualization. *Applied Artificial Intelligence*, 31(4), 299-315. https://doi.org/10.1080/08839514.2017.1315516

Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing* (4th ed.). Pearson.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. (Chapter 5: Machine Learning Basics)

Kamilaris, A., & Prenafeta-Boldú, F. X. (2018). Deep learning in agriculture: A survey. *Computers and Electronics in Agriculture*, 147, 70-90. https://doi.org/10.1016/j.compag.2018.02.016

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436–444. https://doi.org/10.1038/nature14539

Mohanty, S. P., Hughes, D. P., & Salathé, M. (2016). Using deep learning for image-based plant disease detection. *Frontiers in Plant Science*, 7, 1419. https://doi.org/10.3389/fpls.2016.01419

Putra, I. C., & Prabowo, W. A. E. (2024, September). Implementation of Convolutional Neural Network Based on InceptionV3 to Classify Guava Quality. In *2024 International Seminar on Application for Technology of Information and Communication (iSemantic)* (pp. 112-117). IEEE. https://doi.org/10.1109/iSemantic63362.2024.10762157

Rahayu, R. A., & Fitriani, D. (2021). Deteksi Tingkat Kematangan Buah Menggunakan Teknologi Pengolahan Citra Digital. *Jurnal Teknologi dan Sistem Komputer*, 9(3), 207–214. https://doi.org/10.14710/jtsiskom.9.3.207-214

Rizzo, M., Marcuzzo, M., Zangari, A., Gasparetto, A., & Albarelli, A. (2023). Fruit ripeness classification: A survey. *Artificial Intelligence in Agriculture*, *7*, 44-57. https://doi.org/10.1016/j.aiia.2023.02.004

Saragih, R. E., & Emanuel, A. W. (2021, April). Banana ripeness classification based on deep learning using convolutional neural network. In *2021 3rd East Indonesia Conference on Computer and Information Technology (EIConCIT)* (pp. 85-89). IEEE. https://doi.org/10.1109/EIConCIT50028.2021.9431928

Sari, M. W., Sitorus, S. P., & Pane, R. (2025). Implementation of Convolutional Neural Network (CNN) Method in Determining the Level of Ripeness of

Mango Fruit Based on Image. *Jurnal Penelitian Pendidikan IPA*, *11*(5), 419-428. https://doi.org/10.29303/jppipa.v11i5.11436

Wiktasari, T. R. Y., Alifiansyah, M. F., Kurniangsih, L. T., & Hasan, A. (2025). Classification System of Crystal Guava (Psidium Guajava) Using Convolutional Neural Network And Rectrified Linear Unit Method Based on Android. *JAICT*, *10*(1), 328-340.