International Journal of Mathematics and Computer Research

ISSN: 2320-7167

Volume 13 Issue 04 April 2025, Page no. – 5107-5113

Index Copernicus ICV: 57.55, Impact Factor: 8.615

DOI: 10.47191/ijmcr/v13i4.13

Impact of Activation Function on the Performance of Convolutional Neural Network in Identifying Oil Palm Fruit Ripeness

Naufal Budiman[1], Kusworo Adi[2], Adi Wibowo[3]

[1,2,3] Department of Information Systems, School of Postgraduate, Diponegoro University, Semarang, Indonesia

ARTICLE INFO

Published Online:
22 April 2025

ABSTRACT

Activation functions play a crucial role in Convolutional Neural Networks (CNN), particularly in enabling the model to recognize and represent complex patterns in digital images. In image classification tasks, the choice of activation function can significantly impact the accuracy and overall performance of the model. The Rectified Linear Unit (ReLU) is the most commonly used activation function due to its simplicity; however, it has a limitation in discarding information from negative input values. To address this issue, alternative functions such as Leaky ReLU and Gaussian Error Linear Unit (GELU) have been developed, designed to retain information from negative inputs. This study presents a comparative analysis of three activation functions ReLU, Leaky ReLU, and GELU on a CNN model for classifying oil palm fruit ripeness levels. The results show that although all three activation functions achieved high training accuracy ReLU at 100%, Leaky ReLU at 99.93%, and GELU at 99.49%—the performance on testing data varied significantly. Leaky ReLU outperformed the others, achieving the highest test accuracy of 95.35%, an F1-score of 95.39%, and a Matthews Correlation Coefficient (MCC) of 93.28%. It also exhibited the smallest gap between training and testing accuracy (4.58%), indicating better generalization capability and a lower risk of overfitting compared to ReLU and GELU. Moreover, the model using Leaky ReLU was able to classify all three classes more evenly, particularly excelling in identifying the 'ripe' class, which tends to be more challenging. These findings highlight that Leaky ReLU is a more optimal activation function for oil palm fruit image classification, as it maintains high accuracy while effectively reducing overfitting. This study contributes to the selection of appropriate activation functions for CNN-based classification systems and opens opportunities for exploring more adaptive activation functions in future research.

Corresponding Author:
Naufal Budiman

KEYWORDS: Convolutional Neural Network, Activation Function, Image Classification

I. INTRODUCTION

The advancement of artificial intelligence technology, particularly in the field of Deep Learning, has led to significant progress across various sectors, including agriculture. One of its applications is image classification for identifying fruit ripeness levels, which plays a crucial role in improving harvest quality and production efficiency [1]. Oil palm fruit is one of Indonesia's primary commodities, and accurately determining its ripeness significantly impacts palm oil yield [2]. Convolutional Neural Networks (CNN) are among the most widely used Deep Learning (DL) architectures for image processing. CNN offer a high level of abstraction and the ability to automatically learn patterns within images [3].

Feature extraction is the most fundamental part of CNNs. This process underpins the formation of trainable multilayer networks, supported by nonlinear activation functions and downsampling techniques [4]. In this process, activation functions play a vital role by preserving important information captured by convolutional layers, discarding irrelevant data, and transforming feature data through nonlinear approaches. Activation functions are essential, as without them, the network would be unable to comprehend complex patterns in the data [5].

One of the most commonly used activation functions is the Rectified Linear Unit (ReLU), known for its simplicity and efficiency [6]. However, ReLU has limitations in handling negative values, which can lead to some neurons becoming permanently inactive—a condition known as the "dying ReLU" problem—where a significant number of neurons stop responding, ultimately hindering the learning process and potentially reducing model accuracy [7].

To address this issue, several alternative activation functions have been developed, including Leaky ReLU and the Gaussian Error Linear Unit (GELU). The Leaky ReLU function allows a small portion of negative values to pass through by multiplying them with a small constant [8]. This approach helps prevent the dying ReLU phenomenon, allowing the network to learn from potentially relevant negative features, thereby improving accuracy and model generalization [9]. Meanwhile, GELU employs a more refined statistical approach by activating neurons based on the probability of their inputs following a normal distribution. Instead of directly cutting off negative values like ReLU, GELU considers the likelihood that an input contributes positively to the final outcome, resulting in smoother and more adaptive activation [10].

Although both functions have shown potential in improving model accuracy, a comprehensive evaluation of their performance is necessary, particularly in the context of oil palm fruit image classification. This study aims to compare the performance of three activation functions ReLU, Leaky ReLU, and GELU in a CNN model for classifying the ripeness of oil palm fruits. The evaluation considers not only classification accuracy but also the stability of performance on training and validation data, in order to identify potential overfitting. Through this study, it is expected that a deeper understanding of the impact of activation function selection on model performance can be obtained.

II. LITERATURE REVIEW

2.1. Convolutional Neural Network

Convolutional Neural Network (CNN) is a type of artificial neural network specifically designed to recognize visual patterns in a way that resembles the human visual perception mechanism [11]. CNN consists of several main layers that work sequentially. As shown in Figure 1, CNN begins with an input layer that receives the data, followed by convolutional layers that extract important features using filters. The output of the convolutional layers is then processed by an activation function, which introduces non-linearity into the network. This function is crucial, as it enables the network to learn and model complex relationships within the data something that cannot be achieved through linear transformations alone [12].

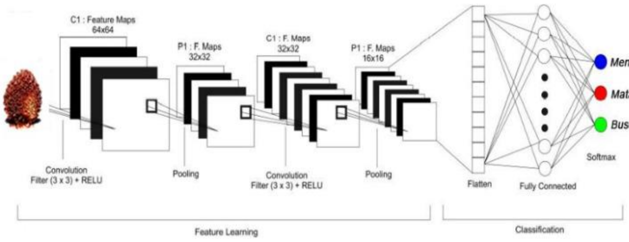

Figure 1. CNN Architecture

The activation function is one of the most crucial components in the architecture of a CNN, as it acts like a switch that determines whether a neuron should be activated or not. Without an activation function, a neural network would perform only linear computations and would be unable to capture complex patterns in visual data. Activation functions introduce non-linearity into the model, which is essential for learning intricate relationships between features in an image. Moreover, activation functions play a key role in the backpropagation process, during which the network learns and adjusts its weights through gradient calculations. The derivative of the activation function determines the magnitude of weight updates that should be made [13]. The activation function works by taking the result of the convolution operation (typically a weighted sum plus a bias), and then applying a non-linear function to it. Mathematically, this process can be expressed as:

$$h^k = f(w^k . x + b^k) \qquad (1)$$

Where x is the input from the previous layer, w^k and b^k are the weights and bias for the k-th filter, f is the activation function, and h^k is the output of the activation. The next step is feature simplification or down-sampling through the pooling layer, which serves to reduce the network's complexity, accelerate training time, and minimize the risk of overfitting. Pooling functions such as max pooling or average pooling are applied to local regions of size p × p, where p is the size of the pooling kernel [14]. Subsequently, the fully connected (FC) layer takes features from the previous layer both mid-level and low-level features and transforms them into high-level representations. This final stage resembles a traditional neural network. The classification process is carried out in the output layer using methods such as softmax, where the final result is a probability value indicating the likelihood that the input belongs to a particular class [15].

2.2. ReLU Activation Function

The Rectified Linear Unit (ReLU) activation function is a type of non-linear function designed to address the vanishing gradient problem. It operates by allowing positive input values to pass through unchanged, while negative values are converted to zero. Its simple yet effective nature enables the gradient computation to continue efficiently, thereby accelerating and stabilizing the training of artificial neural networks [6]. The equation for ReLU is shown in Equation 2. This function is widely used because it effectively mitigates the vanishing gradient issue. The graphical representation of the ReLU activation function can be seen in Figure 2.

$$f(x) = max(0, x) = \begin{cases} x_i, & if\ \ x_i \geq 0 \\ 0, & if\ \ x_i < 0 \end{cases} \qquad (2)$$

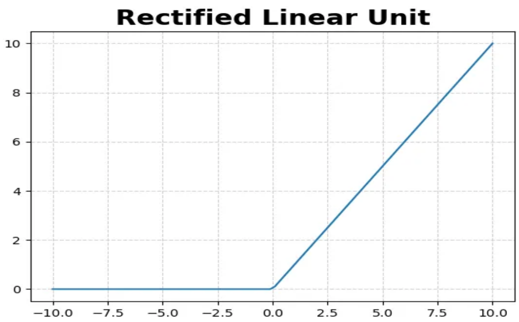

Figure 2. ReLU Activation Function

Although the ReLU function can overcome the vanishing gradient problem commonly found in activation functions such as sigmoid and tanh, it also has certain drawbacks. For input values that are negative, ReLU outputs zero. As a result, the neuron may stop learning because its gradient becomes zero continuously a condition known as the "dying neuron" problem. Moreover, since many of ReLU's outputs are zero (especially for negative inputs), the data distribution can become unbalanced and lack a zero centered mean, which may negatively impact the training process of the neural network [7].

2.3. Leaky ReLU Activation Function

Leaky ReLU is a variant of the ReLU activation function proposed to avoid the "dying ReLU" problem during neural network training. The "dying ReLU" issue occurs when weights get stuck at zero because ReLU outputs zero for all input values less than zero. Leaky ReLU addresses this by allowing a small gradient to pass through even when the input is less than zero, thereby reducing the likelihood of neurons becoming inactive during training [16]. In this way, the gradient in the negative region remains non-zero, albeit small, enabling the network to continue learning effectively and minimizing the risk of dead neurons. Leaky ReLU offers a more stable alternative to ReLU, particularly in maintaining information flow during the training process of neural networks [17]. The equation for Leaky ReLU is shown in Equation 3, and its graphical representation can be seen in Figure 3.

$$f(x) = ax + x = \begin{cases} x & if \;\; x > 0 \\ ax & if \;\; x \leq 0 \end{cases} \qquad (3)$$

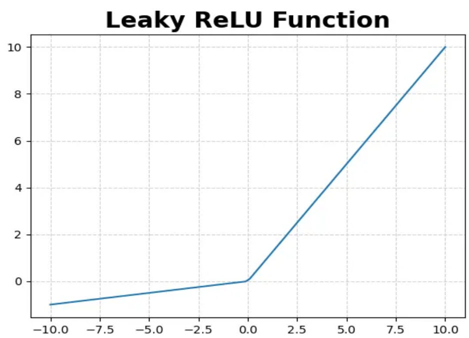

Figure 3. Leaky ReLU Activation Function

By introducing a slope parameter α in Equation 3, the Leaky ReLU activation function emerges as a solution to the issues in the negative input region. As a result, in various deep learning studies found in the literature, this function is widely used as an alternative to the ReLU activation function [6].

2.4. GeLU Activation Function

The Gaussian Error Linear Unit (GeLU) activation function is designed to combine the advantages of activation functions like ReLU (which is fast and efficient in training) and techniques like Dropout (which help reduce overfitting). Unlike ReLU, which directly sets negative values to zero, GeLU allows input values to pass through based on a probability calculated using the Gaussian (normal) distribution. In other words, each input value is not strictly activated or deactivated but is partially passed through depending on its magnitude, making it a smoother alternative to ReLU [18]. GeLU introduces a stochastic approach to determining whether a neuron should be active, similar to Dropout, while maintaining the differentiable and continuous nature of an activation function. This makes it highly effective for training deep neural networks [19]. The equation for GeLU is shown in Equation 4, and its graphical representation can be seen in Figure 4.

$$f(x) = x.\frac{1}{2}[1 + erf\,\frac{x}{\sqrt{2}}] \qquad (4)$$

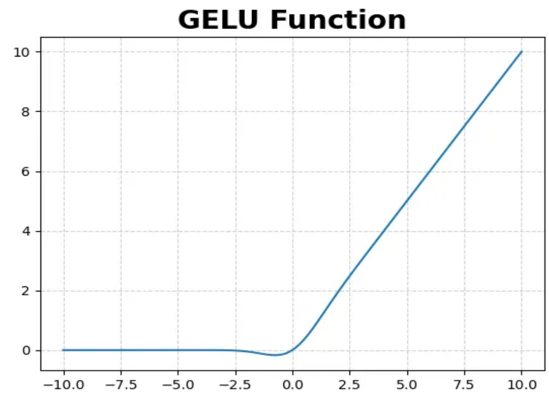

Figure 4. GeLU Activation Function

GELU is widely used in fields such as computer vision, natural language processing (NLP), and speech recognition due to its ability to enhance model performance and reduce overfitting. However, its main drawbacks are the heavier computational cost and more complex optimization process compared to other activation functions like ReLU [20].

III. RESEARCH METHODS

This study will analyze the impact of using different activation functions in the Convolutional Neural Network (CNN) model designed for identifying the ripeness of oil palm fruit. The activation functions compared are ReLU (Rectified Linear Unit), Leaky ReLU, and GeLU (Gaussian Error Linear Unit). The goal of this experiment is to evaluate the model's performance in classifying the ripeness levels of oil palm fruit based on the type of activation function used in each convolution and fully connected layer. The architecture structure of the CNN remains the same, consisting of 4 convolutional layers, each with 48 filters of size 3×3, followed by a 3×3 pooling layer with a stride of 1×1 after each convolution. The model is also equipped with a fully connected layer consisting of 242 neurons and using a dropout rate of 29.54% to reduce overfitting. The model optimization is performed using the Adam algorithm with a learning rate of 0.000484397 and 25 epochs. By maintaining consistent structure and parameters, this study will measure the impact of each activation function on accuracy, loss, and training stability in classifying oil palm fruit images into three categories: ripe, unripe, and rotten. The CNN model parameters to be used are shown in Table 1.

Table 1. CNN Model Parameters

Parameters	Value
Number of Convolutions	4
Number of Filters	48 per layer
Kernel Size	3x3
Pooling Size	3x3
Convolution Stride	2x2
Pooling Stride	1x1
Neurons in FC Layer	242
Dropout Rate	29.54%
Optimizer	Adam
Learning Rate	0.000484397
Epoch	25

The dataset used in this study consists of photos of oil palm fruit from plantations, categorized into ripe, unripe, and rotten conditions. The dataset was directly captured using a smartphone camera at the oil palm plantation of the Agricultural Training Center in Jambi, consisting of 3,974 images of oil palm fruit. This includes 1,316 images of ripe oil palm fruit, 1,355 images of unripe oil palm fruit, and 1,303 images of rotten oil palm fruit.

IV. RESULT

In this study, an evaluation was conducted on three types of activation functions ReLU (Rectified Linear Unit), Leaky ReLU, and GeLU (Gaussian Error Linear Unit) within a CNN model used for identifying the ripeness level of oil palm fruit. The objective of this comparison is to determine the effect of each activation function on the model's ability to learn and generalize to new data, particularly because activation functions play a crucial role in introducing non-linearity into the network and significantly influence the backpropagation process.

The results show that all three models achieved high training accuracy: ReLU achieved 100%, Leaky ReLU reached 99.93%, and GeLU achieved 99.49%. The f1-score and MCC (Matthews Correlation Coefficient) also reflected strong training performance, with ReLU scoring 100% for both f1-score and MCC, Leaky ReLU achieving an f1-score of 99.93% and MCC of 99.89%, and GeLU scoring an f1-score of 99.49% and MCC of 99.24%. The training performance results for the three activation functions are presented in Table 2.

Table 2. Training Result

	ReLU	Leaky ReLU	GeLU
Accuracy	100%	99.93%	99.49%
F1-Score	100%	99.93%	99.49%
Recall	100%	99.93%	99.49%
Precision	100%	99.93%	99.49%
MCC	100%	99.89%	99.24%

Nevertheless, the extremely high performance on the training data indicates a potential risk of overfitting, particularly in models using the ReLU and GeLU activation functions. While these models demonstrated excellent capability in recognizing patterns within the training set, they struggled to maintain the same level of performance on the validation data. The training and validation graphs for each activation function can be seen in Figures 5 through 10.

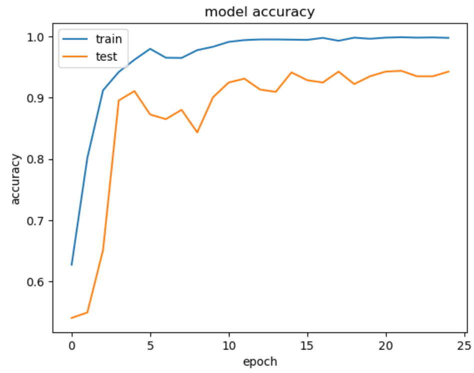

Figure 5. Training and Validation Model Accuracy Using ReLU

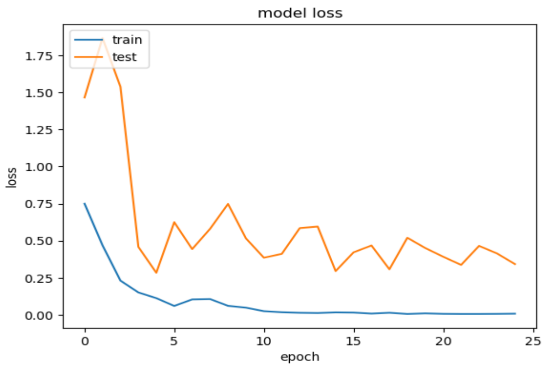

Figure 6. Training and Validation Model Loss Using ReLU

The model utilizing the ReLU activation function demonstrated very high training accuracy, reaching nearly 1.0 within the first few epochs. However, the validation accuracy exhibited considerable fluctuations and never matched the level of training accuracy. This pattern indicates a clear sign of overfitting, where the model becomes too tailored to the training data and struggles to generalize effectively to unseen data. The loss curves further reinforce this observation. While the training loss consistently decreased, the validation loss showed large fluctuations and lacked a steady downward trend. This suggests that although the model was learning well on the training set, it had difficulty maintaining performance on the validation set, highlighting the limitations of ReLU in preventing overfitting in this specific task.

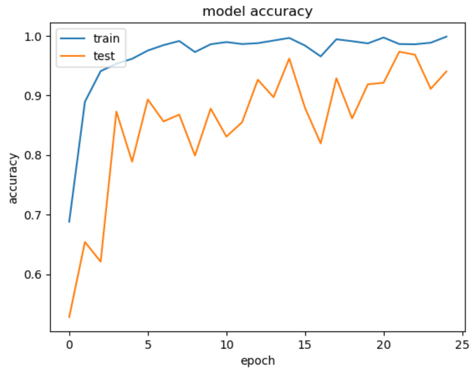

Figure 7. Training and Validation Model Accuracy Using Leaky ReLU

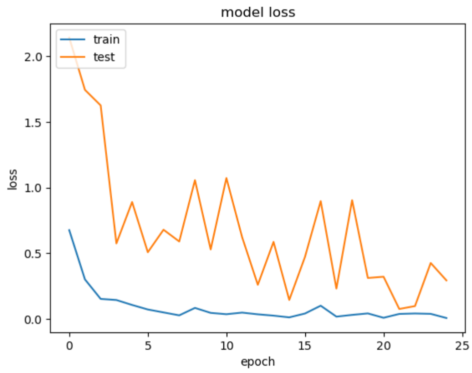

Figure 8. Training and Validation Model Loss Using Leaky ReLU

The performance of the model using Leaky ReLU showed promising results. Training accuracy increased rapidly and stabilized close to 1.0, similar to the model with ReLU. However, the most notable difference lies in the validation accuracy, which increased more steadily, despite some minor fluctuations. This suggests that Leaky ReLU helps the model learn more stably and maintain a better balance between training and validation, effectively reducing the risk of overfitting. The training loss decreased smoothly, and although validation loss also exhibited fluctuations, their amplitude was smaller compared to the ReLU and GeLU graphs. This visual pattern supports the quantitative results, indicating that Leaky ReLU provides better generalization — the model not only learns from the training data but also retains consistent performance on validation data.

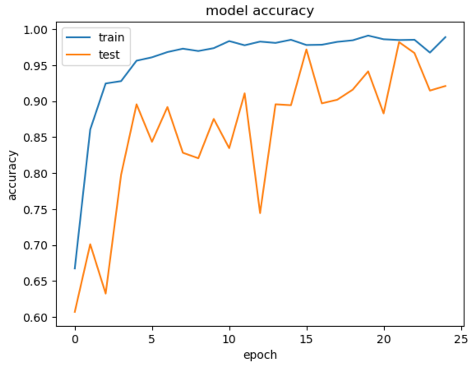

Figure 9. Training and Validation Model Accuracy Using GeLU

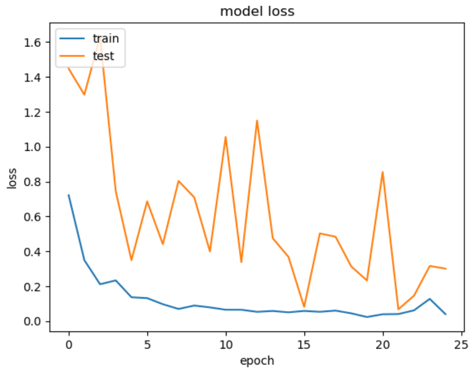

Figure 10. Training and Validation Model Loss Using GeLU

The performance of the model using the GeLU (Gaussian Error Linear Unit) activation function shows a pattern similar to the previous two models in terms of training accuracy, which quickly rises toward the maximum value. However, the validation accuracy appears to be less stable compared to both Leaky ReLU and even ReLU. The validation accuracy fluctuates more sharply, indicating that the model struggles to maintain consistent performance on test data. Similarly, the validation loss displays higher fluctuations and does not exhibit a stable downward trend, further reinforcing the indication that GeLU yields less consistent performance compared to ReLU and Leaky ReLU. This may be attributed to the more complex mathematical nature of GeLU, which could make the model more sensitive to noise in the data, resulting in instability during validation.

When tested on the test dataset, the model's performance began to show significant differences. The model using the Leaky ReLU activation function recorded the best performance, with a test accuracy of 95.35%, an F1-score of 95.39%, and an MCC (Matthews Correlation Coefficient) of 93.28%. On the other hand, the ReLU-based model achieved a test accuracy of 94.62%, an F1-score of 94.65%, and an MCC of 92.27%, while GeLU recorded the lowest test accuracy of 93.40%, with an F1-score of 93.40% and an MCC of 90.57%. In addition, Leaky ReLU also exhibited the smallest gap between training and test accuracy, approximately 4.58%, compared to 5.38% for ReLU and 6.09% for GeLU. This gap reflects the degree of overfitting to the training data — the smaller the gap, the better the model's generalization capability to new data. Therefore, Leaky ReLU proves to be more effective in reducing overfitting compared to both ReLU and GeLU. The results of the testing phase are presented in Table 3.

Table 3. Testing Accuracy of Each Function Activation

	ReLU	Leaky ReLU	GeLU
Accuracy	94.62%	95.35%	93.40%
F1-Score	94.65%	95.39%	93.40%
Recall	94.62%	95.35%	93.40%
Precision	95.35%	95.91%	94.35%
MCC	92.27%	93.28 %	90.57%

Another factor contributing to the superior performance of Leaky ReLU lies in its inherent ability to address the dying ReLU problem, a condition where neurons become permanently inactive. By allowing a small gradient for negative input values, Leaky ReLU ensures that learning can still occur even when a unit receives negative inputs. This characteristic enhances the model's ability to represent features, especially during long training processes or when dealing with complex datasets such as oil palm fruit image classification.

In contrast, ReLU, which lacks a gradient for negative values, risks causing a number of neurons to become inactive and stop learning. Meanwhile, GeLU, with its smooth and probabilistic nature, tends to be too "soft" in learning sharp features, which may reduce the model's discriminative power on high-contrast images like oil palm fruit ripeness. Overall, these findings reinforce that Leaky ReLU is a superior activation function in the context of classifying oil palm fruit ripeness using CNN, as it strikes a better balance between high accuracy and strong generalization capabilities. Although all three activation functions demonstrated signs of overfitting, Leaky ReLU proved to be the most resilient in minimizing its impact and delivering more stable and consistent predictions across all three target classes.

V. CONCLUSION

This study evaluated the impact of three activation functions—ReLU, Leaky ReLU, and GeLU—on the performance of a Convolutional Neural Network (CNN) model in classifying the ripeness levels of oil palm fruit. Based on the training and testing results, it was found that although all three activation functions achieved high accuracy on the training data, only Leaky ReLU demonstrated the best balance between training accuracy and generalization capability on unseen data. The model using Leaky ReLU achieved the highest test accuracy of 95.35%, along with an F1-score of 95.39% and a Matthews Correlation Coefficient (MCC) of 93.28%. Additionally, Leaky ReLU showed the lowest accuracy gap between training and testing (4.58%), indicating the least degree of overfitting compared to ReLU and GeLU when evaluated on real-world data.

Leaky ReLU advantage in this research is attributed to its ability to mitigate the dying neuron problem by introducing a small gradient for negative inputs. This allows the model to continue learning effectively, even in the presence of complex image features, such as those found in the oil palm fruit dataset. In contrast, ReLU poses a greater risk of neuron inactivation, while GeLU, although smoother in mathematical formulation, tends to be too soft in distinguishing sharp features that are essential for image classification tasks.

Thus, it can be concluded that Leaky ReLU is the most optimal activation function for developing CNN models aimed at classifying oil palm fruit ripeness. It not only delivers accurate results but also enhances the model's stability and consistency in recognizing patterns from new data, making it highly recommended for real-world applications in precision agriculture and deep learning-based digital image processing.

REFERENCES

1. S. Espinoza, C. Aguilera, L. Rojas, dan P. G. Campos, "Analysis of Fruit Images with Deep Learning: A Systematic Literature Review and Future Directions," *IEEE Access*, no. October 2023, hal. 1–1, 2023, doi: 10.1109/access.2023.3345789.
2. S. Srivastava, A. V. Divekar, C. Anilkumar, I. Naik, V. Kulkarni, dan V. Pattabiraman, "Comparative analysis of deep learning image detection algorithms," *J. Big Data*, vol. 8, no. 1, 2021, doi: 10.1186/s40537-021-00434-w.
3. J. Naranjo-Torres, M. Mora, R. Hernández-García, R. J. Barrientos, C. Fredes, dan A. Valenzuela, "A review of convolutional neural network applied to fruit image processing," *Appl. Sci.*, vol. 10, no. 10, 2020, doi: 10.3390/app10103443.
4. X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, dan M. Parmar, *A review of convolutional neural networks in computer vision*, vol. 57, no. 4. Springer Netherlands, 2024.
5. Y. Jiang, J. Xie, dan D. Zhang, "An Adaptive Offset Activation Function for CNN Image Classification Tasks," *Electron.*, vol. 11, no. 22, hal. 1–11, 2022, doi: 10.3390/electronics11223799.
6. S. KILIÇARSLAN, K. ADEM, dan M. ÇELİK, "An overview of the activation functions used in deep learning algorithms," *J. New Results Sci.*, vol. 10, no. 3, hal. 75–88, 2021, doi: 10.54187/jnrs.1011739.
7. Y. Bai, "RELU-Function and Derived Function Review," *SHS Web Conf.*, vol. 144, hal. 02006, 2022, doi: 10.1051/shsconf/202214402006.
8. Y. Sun, "The role of activation function in image classification," *2021 IEEE 3rd Int. Conf. Commun. Inf. Syst. Comput. Eng. CISCE 2021*, no. Cisce, hal.

275–278, 2021, doi: 10.1109/CISCE52179.2021.9445868.

9. A. Mujhid, S. Surono, N. Irsalinda, dan A. Thobirin, “Comparison and Combination of Leaky ReLU and ReLU Activation Function and Three Optimizers on Deep CNN for COVID-19 Detection,” *Front. Artif. Intell. Appl.*, vol. 358, hal. 50–57, 2022, doi: 10.3233/FAIA220369.
10. M. Lee, “GELU Activation Function in Deep Learning: A Comprehensive Mathematical Analysis and Performance,” hal. 1–19, 2023, [Daring]. Tersedia pada: http://arxiv.org/abs/2305.12073.
11. L. Chen, S. Li, Q. Bai, J. Yang, S. Jiang, dan Y. Miao, “Review of Image Classification Algorithms Based on Convolutional Neural Networks,” *Remote Sens.*, vol. 13, no. 22, 2021, doi: 10.3390/rs13224712.
12. R. H. K. Emanuel, P. D. Docherty, H. Lunt, dan K. Möller, “The effect of activation functions on accuracy, convergence speed, and misclassification confidence in CNN text classification: a comprehensive exploration,” *J. Supercomput.*, vol. 80, no. 1, hal. 292–312, 2024, doi: 10.1007/s11227-023-05441-7.
13. A. S. Tomar, A. Sharma, A. Shrivastava, A. S. Rana, dan P. Yadav, “A Comparative Analysis of Activation Function, Evaluating their Accuracy and Efficiency when Applied to Miscellaneous Datasets,” *Proc. 2nd Int. Conf. Appl. Artif. Intell. Comput. ICAAIC 2023*, no. Icaaic, hal. 1035–1042, 2023, doi: 10.1109/ICAAIC56838.2023.10140823.
14. L. Alzubaidi *dkk*, *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*, vol. 8, no. 1. Springer International Publishing, 2021.
15. S. Khan, H. Rahmani, S. A. A. Shah, dan M. Bennamoun, “A Guide to Convolutional Neural Networks for Computer Vision,” *Synth. Lect. Comput. Vis.*, vol. 8, no. 1, hal. 1–207, 2018, doi: 10.2200/s00822ed1v01y201712cov015.
16. R. Parhi dan R. D. Nowak, “The Role of Neural Network Activation Functions,” *IEEE Signal Process. Lett.*, vol. 27, hal. 1779–1783, 2020, doi: 10.1109/LSP.2020.3027517.
17. A. Maniatopoulos dan N. Mitianoudis, “Learnable Leaky ReLU (LeLeLU): An Alternative Accuracy-Optimized Activation Function,” *Inf.*, vol. 12, no. 12, 2021, doi: 10.3390/info12120513.
18. M. Lee, “Mathematical Analysis and Performance Evaluation of the GELU Activation Function in Deep Learning,” *J. Math.*, vol. 2023, 2023, doi: 10.1155/2023/4229924.
19. A. Nguyen, K. Pham, D. Ngo, T. Ngo, dan L. Pham, “An analysis of state-of-the-art activation functions for supervised deep neural network,” *Proc. 2021 Int. Conf. Syst. Sci. Eng. ICSSE 2021*, hal. 215–220, 2021, doi: 10.1109/ICSSE52999.2021.9538437.
20. C. C. Nworu, E. J. Ekpenyong, J. Chisimkwuo, dan C. N. Onyeukwu, “The Effects of Modified ReLU Activation Functions in Image Classification Biomedical Engineering and Medical Devices The Effects of Modified ReLU Activation Functions in Image Classification,” vol. 7, no. November, hal. 0–5, 2022, doi: 10.35248/2475-7586.22.07.237.