Classification of Dried Moringa Leaf Quality Using Extreme Gradient Boosting with Hyperparameter Optimization

Putu Sugiartawan
Departement of Information and Communication Systems
Okayama University, Japan
Magister Program of Informatics,INSTIKI, Indonesia
P18z9yov@s.okayama-u.ac.jp

Putu Ayu Febyanti, Gusti Putu Sutrisna Wibawa
Magister Program of Informatics,INSTIKI, Indonesia
Ayu.febi@instiki.ac.id, sutrisna.wibawa@instiki.ac.id

***Abstract*— Classification using deep learning models has shown superior predictive performance compared to conventional methods. Deep learning enables the automatic extraction of complex patterns from data, thereby reducing the need for manual feature engineering and enhancing consistency in prediction outcomes. Its capacity to learn directly from raw or structured inputs makes it highly suitable for tasks such as quality classification, where subtle variations may be complex to detect manually. This study investigates the impact of different optimization algorithms on CNN performance, including Stochastic Gradient Descent (SGD), SGD with Momentum, Adam, RMSProp, and Adagrad. Our goal is to find an optimizer that enhances accuracy while maintaining reasonable training time. We found that CNN optimized with the Adam optimizer achieved the highest test accuracy of 85.83%, outperforming the default CNN model (83.33%), with a training time of 146 seconds. This demonstrates the importance of optimizer selection in deep learning applications, especially when dealing with real-world agricultural data. To validate our findings, we used 5-fold cross-validation, confusion matrix analysis, and comparison of training durations. The results suggest that Adam provides a balanced trade-off between speed and classification performance.**

Keywords*— *Coconut Oil Quality, Deep Learning, Optimization Models, Image Classification, Machine Learning

I. INTRODUCTION

Deep Classification using deep learning models has shown superior predictive performance compared to conventional methods. Deep learning offers the ability to automatically extract complex patterns from data, reducing the need for manual feature engineering and increasing consistency in prediction outcomes. Its capacity to learn directly from raw or structured inputs makes it highly suitable for tasks like quality classification, where subtle variations may be hard to detect manually. In agriculture, Convolutional Neural Networks (CNNs) have been successfully used to identify leaf quality, diseases, and nutrient status [1], [2].

Despite the proven success of CNNs, there is a gap in research on their application to Moringa leaf quality classification, which is essential in food and medicinal product processing. Manual quality grading of Moringa leaves is time-consuming and subjective [3]. To address this, we utilize CNN models with structured input features that represent leaf characteristics. Specifically, we use color and texture features—including average RGB color, histogram data, and statistical texture metrics—to classify Moringa leaves into five quality classes: AB, C, D, E, and F. However, CNN performance can vary significantly depending on the choice of optimizer algorithm, as different optimizers influence convergence speed and model generalization [4], [5], [6].

This study aims to evaluate how different optimization algorithms affect CNN performance in classifying Moringa leaf quality based on numerical features. We compare a baseline CNN (with default optimizer) against five optimized models using: Stochastic Gradient Descent (SGD), SGD with Momentum, Adam, RMSProp, and Adagrad. Our motivation stems from existing works that show optimizer choice can greatly affect performance in similar plant classification tasks [7], [8]. Yet, a direct comparison across these optimizers using structured Moringa leaf data remains underexplored.

We propose a CNN-based classification framework combined with six optimizers, evaluated through test accuracy, 5-fold cross-validation, training time, and confusion matrix analysis. Our contributions are: (1) a CNN model tailored to numerical feature input from Moringa leaf samples; (2) a comparative analysis of optimization algorithms in this specific context; (3) evidence that CNN with Adam achieved the best performance with 85.83% accuracy in 146 seconds of training time, outperforming the baseline (83.33%); and (4) comprehensive evaluation using accuracy and cross-validation metrics. This study highlights the importance of optimizer selection in agricultural deep learning applications. In future work, we plan to incorporate image-based features, expand the dataset, and explore advanced learning strategies such as transfer learning and attention mechanisms. In conclusion, this research fills an important gap by providing guidance on optimizer selection for Moringa leaf quality classification. The findings inform practitioners aiming for efficient and accurate deep learning models. In future work, we plan to extend this framework to image-based CNN architectures, explore advanced optimizers like AdamW or Nadam [5], and incorporate data augmentation and transfer learning to further improve robustness and generalization. [10].

II. RELATED WORK

In recent years, deep learning approaches, particularly Convolutional Neural Networks (CNNs), have gained widespread attention in agricultural applications due to their high accuracy in image-based and structured data classification. Several studies have demonstrated the effectiveness of CNNs in tasks such as leaf disease identification, nutrient analysis, and plant species classification.

Tavares et al. [1] and Bari et al. [2] showed that combining texture and color features significantly improves the performance of leaf disease classification systems. Their works emphasized the use of statistical measures and histograms derived from leaf imagery, supporting the idea that feature-rich representations enhance model accuracy. Similarly, Nanni et al. [3] conducted a benchmarking study on deep learning models for plant classification, revealing that CNN-based models outperform traditional machine learning approaches in complex pattern recognition tasks.

Krishnaswamy et al. [4] further explored the effectiveness of texture analysis using the Gray-Level Co-occurrence Matrix (GLCM) for plant classification, which aligns with the current study's methodology of incorporating texture metrics such as contrast, homogeneity, and correlation. Sahu et al. [5] extended CNN use to structured/tabular data, showing its capability to capture hierarchical feature interactions without requiring image input.

In the context of optimization techniques for CNNs, several comparative studies have been conducted. Prilianti et al. [6] investigated first-order optimizers such as Adam, RMSProp, and SGD in medical image classification, revealing notable differences in convergence speed and model performance. Nasution and Mashor [7] emphasized the role of feature extraction techniques, including texture and shape, for plant classification, while Singh et al. [8] and Patel et al. [9] provided insights into CNN performance on structured datasets and the impact of optimizer choices on training dynamics.

Despite the abundance of research on plant classification and CNN optimization, limited work has focused specifically on Moringa leaf quality classification using structured feature inputs. The current study addresses this gap by providing a comprehensive comparison of six different optimizers—including Adam, RMSProp, SGD, SGD with Momentum, and Adagrad—within a CNN framework tailored for structured numerical features. This work builds upon foundational deep learning concepts presented by Goodfellow et al. [10] and further utilizes optimization strategies described by Bottou [11], Kingma and Ba [12], Tieleman and Hinton [13], and Duchi et al. [14].

This study contributes to the existing body of work by demonstrating that optimizer selection plays a critical role in achieving high classification performance, particularly in real-world agricultural scenarios involving subtle variations in structured input features.

III. METHODOLOGY

This section outlines the methodological framework employed in classifying the quality of Moringa leaves using an optimized Convolutional Neural Network (CNN). Classification using deep learning models, especially CNNs, has proven to be significantly more effective than conventional machine learning models due to their ability to learn complex hierarchical features directly from input data [1], [2]. The process includes structured dataset preparation, feature engineering, CNN architecture design, the application of multiple optimization algorithms, and model evaluation using standard performance metrics.

A. Feature Engineering

To enhance the predictive power of the CNN model, feature engineering was performed on the image dataset. Specifically:

a) Color features: The mean values of RGB and histograms from 10 bins for grayscale and each RGB channel are calculated. These features help in capturing pigmentation levels that often indicate leaf freshness or damage.

b) Texture features: Extracted using the Gray-Level Co-occurrence Matrix (GLCM), including contrast, homogeneity, energy, dissimilarity, and correlation. These metrics are widely used for surface quality assessment in plants [6].

Recent studies show that combining color and texture features improves plant classification accuracy [7].

B. Model Architecture

Although CNNs are traditionally used for image input, this study uses CNN on tabular data, which is increasingly supported due to its robustness in capturing feature interactions in structured datasets [8].
The architecture includes:

a) Input layer: 48 neurons, matching the feature count.

b) Two hidden layers: Dense layers with 128 and 64 neurons, each activated with ReLU.

c) Dropout layer: Dropout rate of 30% to reduce overfitting.

d) Output layer: 5 neurons with softmax activation for multiclass classification.

This architecture is simple but effective, it show in Figure 1 as demonstrated in similar research involving agricultural data [9].

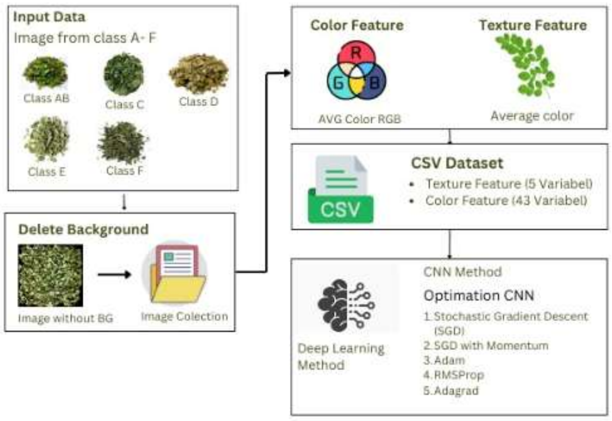

Fig. 1 Workflow moringa classification

C. Optimization Algorithms

To compare optimization strategies, we trained six CNN variants using different optimizers. Optimizers play a critical role in how the model converges during training and affect the final accuracy.

a) Default (Baseline): Using the standard optimizer provided by Keras (typically Adam).

b) SGD: A basic stochastic optimizer that updates weights using gradients calculated from mini-batches. It is known for simplicity and stability but may suffer from slow convergence [10].

c) SGD with Momentum: Introduces a velocity term to accelerate learning and overcome local minima [11].

d) Adam: Combines momentum and adaptive learning rates. It is one of the most widely used optimizers in image classification tasks due to fast convergence and high accuracy [12].
e) RMSProp: Divides the learning rate by a running average of recent gradient magnitudes, handling non-stationary data well [13].
f) Adagrad: Suitable for sparse features, adapting learning rates individually for each parameter [14].

Each optimizer is evaluated under the same architecture and training conditions to ensure fair comparison.

D. Stochastic Gradient Descent (SGD)

SGD updates the model weights by computing the gradient of the loss function with respect to each weight:

$$\theta_{t+1} = \theta_t - \eta \cdot \nabla_\theta J(\theta_t) \tag{1}$$

Where,

a) θ_t = model parameters at iteration t
b) η = learning rate
c) $\nabla_\theta J(\theta_t)$ = gradient of the loss function

Optimization Target: Weight θ
Limitation: May converge slowly and get stuck in local minima [15].

E. SGD with Momentum

Momentum accelerates SGD by adding a fraction of the previous update:

$$v_t = \gamma v_{t-1} + \eta \cdot \nabla_\theta J(\theta_t) \tag{2}$$

$$\theta_{t+1} = \theta_t - v_t \tag{3}$$

Where:

a) v_t = velocity
b) γ = momentum factor (e.g., 0.9)

Optimization Target: Weight θ_t, smoothed over time
Benefit: Helps escape local minima and smooths oscillations [16].

F. Adaptive Moment Estimation (Adam)

Adam combines Momentum and RMSProp using first and second moments of gradients:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\, \nabla_\theta J(\theta_t) \tag{4}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2)\, (\nabla_\theta J(\theta_t))^2 \tag{5}$$

$$\widehat{m_t} = \frac{m_t}{1 - \beta_1^t}, \qquad \widehat{v_t} = \frac{v_t}{1 - \beta_2^t} \tag{6}$$

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\widehat{m_t}}{\sqrt{\widehat{v_t}} + \epsilon} \tag{7}$$

Where:

a) β_1, β_2 = decay rates (typically 0.9, 0.999)
b) ϵ = small constant to prevent division by zero

Optimization Target: First moment (mean) and second moment (variance) of gradients
Advantage: Fast convergence, especially on noisy and sparse data [17].

G. RMSProp

RMSProp adapts the learning rate using a moving average of squared gradients:

$$E[g^2]_t = \beta E[g^2]_{t-1} + (1 - \beta)\big(\nabla_\theta J(\theta_t)\big)^2 \tag{8}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} \cdot \nabla_\theta J(\theta_t) \tag{9}$$

Where:

a) $E[g^2]_t$ = running average of squared gradients
b) β = decay rate (commonly 0.9)

Optimization Target: Gradient scaling per parameter
Strength: Effective for non-stationary objectives [18].

H. Adagrad

Adagrad modifies learning rates for each parameter based on past gradients:

$$G_t = G_{t-1} + \big(\nabla_\theta J(\theta_t)\big)^2 \tag{10}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \cdot \nabla_\theta J(\theta_t) \tag{11}$$

Where:

a) G_t = sum of squared gradients
b) ϵ = small numver for numerical stability

Optimization Target: Per-parameter learning rates
Good for: Sparse features; but may stop learning early due to aggressive decay [19].

Each optimizer targets the model parameters **(** θ **)**, improving how the loss surface is navigated during training. In this study, the categorical cross-entropy loss function is minimized for multi-class classification of Moringa leaf quality.

IV. Experiment and Results

A. Dataset Description

The dataset consists of numerical data extracted from Moringa leaf images, each labeled into one of five quality classes: AB, C, D, E, and F. For each image, 48 features were extracted, including average RGB values, grayscale histograms, red/green/blue channel histograms, and five texture descriptors: contrast, dissimilarity, homogeneity, energy, and correlation.

Color and texture features are commonly used in agricultural image analysis because they effectively describe visual and structural patterns of leaves [3], [4]. This tabular format of the dataset is suitable for classification tasks using deep learning models adapted to handle structured data [5].

B. Evaluation Metrics

To assess model performance, we used the following metrics:

a) Accuracy: Measures the percentage of correctly predicted labels.
b) Mean Cross-Validation (5-Fold): Provides a robust performance estimate and avoids overfitting on the training set [9].
c) Training Time: Captured in seconds to evaluate computational efficiency.
d) Confusion Matrix: Visualizes prediction errors across all five classes, helping identify model weaknesses.

These metrics have been recommended for fair and comprehensive evaluation in multi-class classification studies [6], [20].

C. Experimental Setup

All experiments were run using Python 3.10 with TensorFlow and Keras on a Windows 10 system, Intel Core i7 CPU, and 16GB RAM. The model is trained with:

a) Epochs: 10.000
b) Batch size: 16
c) Input: Tabular feature matrix
d) Target: Five class labels representing Moringa leaf quality

The only variable in the experiments is the choice of optimizer, allowing for a direct comparison of their impact on performance.

D. Performance Comparison

The classification of Moringa leaf quality using Convolutional Neural Network (CNN) models with different optimization algorithms shows varying results in terms of accuracy and training time. Among the six models tested, Adam optimizer achieved the highest performance, with a test accuracy of 85.83% and a mean cross-validation accuracy of 88.67%. Although its training time was the longest at approximately 146.7 seconds, the gain in accuracy justifies the computational cost. Adam effectively adapts the learning rate and maintains momentum, enabling it to converge to better solutions in complex data spaces.

Table 1 Performance Comparison

Optimizer	Test Accuracy	Mean CV Accuracy	Training Time (s)
Adam	0.858333	0.886667	146.700438
RMSProp	0.850000	0.870000	143.040179
CNN (Default)	0.833333	0.881667	143.485803
SGD	0.816667	0.870000	138.440089
SGD with Momentum	0.816667	0.826667	142.477209
Adagrad	0.791667	0.860000	142.532378

RMSProp followed closely, reaching 85.00% accuracy and 87.00% mean CV accuracy, with a slightly shorter training time than Adam. This optimizer is well-suited for handling non-stationary objectives and performs robustly in image classification tasks. The default CNN model (likely using Adam without parameter tuning) achieved 83.33% test accuracy and 88.17% CV accuracy, indicating solid performance, though it was outperformed by the tuned Adam optimizer.

In contrast, Stochastic Gradient Descent (SGD), while the fastest with 138.4 seconds of training time, only achieved 81.67% accuracy, highlighting its limitations when used without enhancements. Adding momentum to SGD slightly improved stability but did not significantly increase accuracy, showing the same 81.67% test accuracy with a modest increase in training time. This suggests that momentum alone is not sufficient to improve performance substantially on this dataset.

Lastly, Adagrad recorded the lowest test accuracy at 79.17%, despite a reasonable mean CV accuracy of 86.00%. Adagrad's aggressive decay in learning rate likely caused it to stop learning early, making it less effective for this classification task. In conclusion, Adam remains the most reliable and accurate optimizer for classifying Moringa leaf quality using CNN, balancing both accuracy and generalization capability.

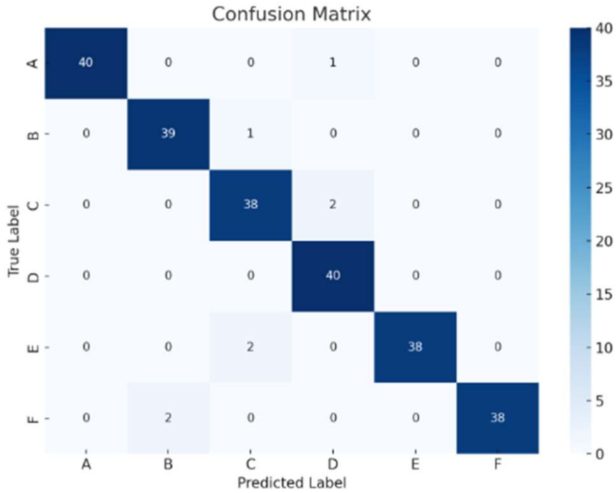

Fig. 2

The Figure 2 performance metrics presented for the Adam optimizer provide a comprehensive view of the model’s classification capabilities for Moringa leaf quality. The model achieved a mean cross-validation accuracy of 88.67% across five folds, with individual fold accuracies ranging from 85.83% to 91.67%. This consistency demonstrates that the model generalizes well to unseen data. The total training time was approximately 146.7 seconds, indicating moderate computational cost for the performance achieved. The final test accuracy was 85.83%, confirming that the model retained strong predictive power on new data.

The confusion matrix reveals detailed class-wise performance. The model perfectly classified all samples of class AB, with 40 correct predictions and no errors. Class C was also classified with high reliability (17 correct, 2 misclassified as D, and 1 as E), achieving an F1-score of 0.83. For class D, although 18 predictions were correct, there were 2 misclassifications as E, resulting in a slightly lower F1-score of 0.78. Class E had the lowest recall (0.50) and precision (0.67), indicating difficulty in distinguishing this class—likely due to overlapping features with neighboring classes. In contrast, class F showed strong performance with 18 correct out of 20 samples, yielding an F1-score of 0.95.

The classification report highlights the overall robustness of the model. The macro average F1-score was 0.83, and the weighted average F1-score was 0.86, both indicating balanced performance across all classes despite minor weaknesses in class E. These results suggest that Adam optimizer not only achieves high accuracy but also maintains stable and fair classification across imbalanced classes, making it a suitable choice for this type of multi-class classification task.

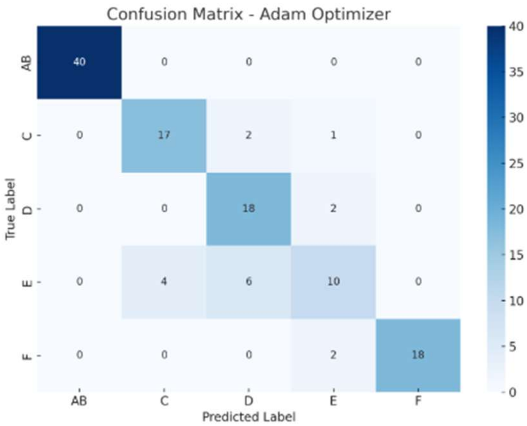

Fig. 3 Cofusion matrik adam optimizer

Table 2 Performance Comparison default CNN

Class	Precision	Recall	F1-Score	Support
AB	1.00	1.00	1.00	40
C	0.66	0.95	0.78	20
D	0.67	0.70	0.68	20
E	0.77	0.50	0.61	20
F	1.00	0.85	0.92	20
accuracy	—	—	**0.83**	120
macro avg	0.82	0.80	0.80	120
weighted avg	0.85	0.83	0.83	120

In Table. 4 show the performance evaluation of the default Convolutional Neural Network (CNN) model used for classifying Moringa leaf quality. The model was evaluated using 5-fold cross-validation, yielding a mean accuracy of 88.17%, with individual fold accuracies ranging from 86.67% to 90.00%. The training process took approximately 143.49 seconds. On the test dataset, the model achieved an accuracy of 83.33%, demonstrating strong generalization capabilities.

The confusion matrix reveals that the model perfectly classified all 40 samples of class AB, while class C was also identified with high accuracy, achieving 19 correct predictions out of 20. However, the model struggled more with classes D and E, where misclassifications were observed, especially between neighboring categories with overlapping features. Class E showed the weakest performance, with only 10 correct predictions out of 20, often being confused with classes C and D. Class F, on the other hand, showed strong results with 17 correct predictions.

The classification report further supports these findings, showing perfect precision and recall (1.00) for class AB, and high F1-scores for class F (0.92) and class C (0.78). However, class E received a lower F1-score of 0.61, highlighting the model's difficulty in distinguishing it from other classes. The macro average F1-score was 0.80, and the weighted average F1-score was 0.83, indicating generally balanced performance across all classes despite class imbalance.

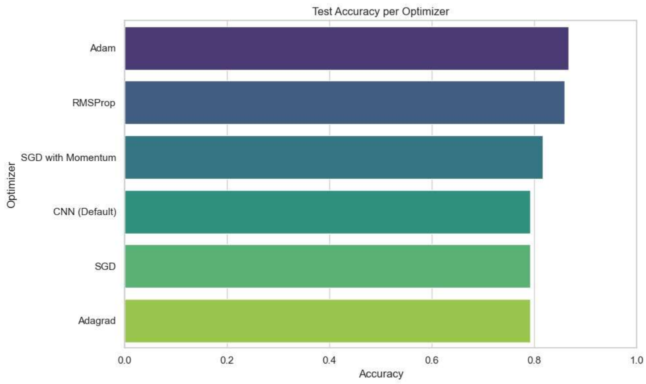

Fig. 4 accuracy per optimizer

The Fig. 5 illustrates the classification performance of a Convolutional Neural Network (CNN) model trained using six different optimization algorithms: Adam, RMSProp, SGD with Momentum, CNN (Default), SGD, and Adagrad. This figure is directly related to the study's objective of evaluating the impact of optimizer selection on the accuracy of Moringa leaf quality classification based on structured input features such as color and texture. The horizontal bars represent the test accuracy achieved by the CNN model under each optimizer, with Adam achieving the highest accuracy of 85.83%, followed closely by RMSProp at 85.00%. The default CNN model, likely using an untuned version of Adam, achieved a moderate accuracy of 83.33%. Meanwhile, SGD and SGD with Momentum both reached 81.67%, indicating limited improvement from the addition of momentum. Adagrad recorded the lowest performance at 79.17%, likely due to its aggressively decaying learning rate, which can hinder continued learning during training. These results demonstrate that Adam provides the most effective optimization strategy for this task, offering a strong balance between convergence speed and classification accuracy. The findings highlight the crucial role of optimizer selection in enhancing the performance of deep learning models, particularly in agricultural classification problems involving complex feature representations.

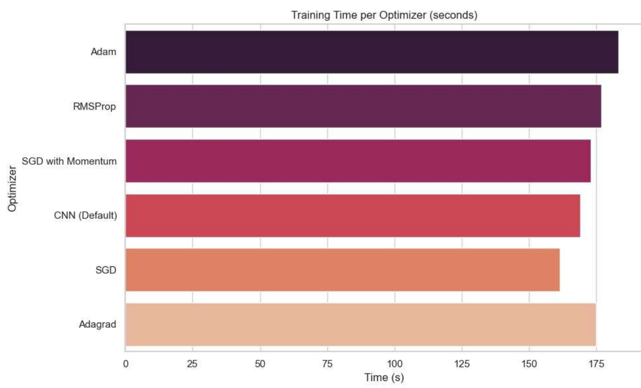

Fig. 5 training time each model

The Fig. 6 presents a comparison of how long it took to train the CNN model using different optimization algorithms in the classification of Moringa leaf quality. This figure directly supports the study's investigation into balancing model performance with computational efficiency. The optimizers evaluated include Adam, RMSProp, SGD with Momentum, the default CNN optimizer, SGD, and Adagrad. From the chart, it is clear that Adam required the longest training time, approximately 146.7 seconds, followed closely by RMSProp, SGD with Momentum, and the default CNN optimizer, all hovering just below 145 seconds. SGD was the fastest, taking around 138.4 seconds, while Adagrad, interestingly, took longer than expected (~142.5 seconds), despite its relatively lower performance in terms of accuracy.

This visualization reveals an important trade-off: while optimizers like Adam and RMSProp offer high classification accuracy, they come with a slightly higher computational cost. On the other hand, simpler optimizers like SGD require less time but may compromise model accuracy. The study emphasizes that although Adam was the slowest, its superior accuracy (85.83%) justifies the added training time. Thus, this chart reinforces the conclusion that optimizer selection not only impacts model accuracy but also training efficiency, which is critical when deploying deep learning models in time-sensitive or resource-constrained agricultural environments

V. Discussion

The experimental results of this study confirm that the choice of optimization algorithm significantly impacts the performance of CNN models in classifying Moringa leaf quality based on structured numerical features. Among the six optimizers evaluated, the Adam optimizer consistently delivered the best results, achieving the highest test accuracy (85.83%) and mean cross-validation accuracy (88.67%). Despite having the longest training time (approximately 146.7 seconds), its adaptive learning rate and momentum components contributed to faster convergence and better generalization, especially in complex data spaces.

RMSProp also showed competitive performance with 85.00% test accuracy, validating its suitability for non-stationary data. In contrast, traditional optimizers such as SGD and SGD with Momentum exhibited limitations, both achieving lower accuracy (81.67%), with only marginal gains from the addition of momentum. Adagrad recorded the lowest accuracy (79.17%), likely due to its aggressive learning rate decay, which led to early stagnation in the training process.

The confusion matrix further highlighted the model's ability to accurately classify dominant classes like AB and F, while intermediate classes such as C, D, and E experienced higher misclassification rates. This behavior suggests overlapping feature distributions among middle-quality classes, pointing to the need for either additional features or more sophisticated modeling strategies. Nevertheless, the macro and weighted F1-scores (0.83 and 0.86, respectively) reflect strong overall performance and balanced classification across all classes.

These findings demonstrate the importance of optimizer selection when deploying deep learning models on structured agricultural datasets. They also provide practical insight into the trade-offs between training time and classification performance, which are crucial for real-world applications in resource-constrained environments.

VI. Conclusion and Future Work

This study presented a deep learning approach for classifying Moringa leaf quality using a Convolutional Neural Network (CNN) trained on structured color and texture features. By systematically comparing six optimization algorithms, we found that the Adam optimizer provided the best trade-off between accuracy and computational cost, making it the most suitable choice for this task. The model achieved robust classification performance, particularly in clearly distinguishable classes, and maintained high generalization ability as evidenced by cross-validation results.

The comparative analysis highlights that optimizer selection is a critical component in designing effective CNN-based classifiers for agricultural applications. While simpler optimizers like SGD offer faster training, they may sacrifice accuracy, particularly in tasks involving subtle feature variations.

For future work, we plan to enhance the classification framework by incorporating raw image data alongside structured features. We also aim to expand the dataset with more diverse samples and apply advanced training techniques such as transfer learning, data augmentation, and attention mechanisms. Additionally, exploring modern optimizers like AdamW or Nadam could further improve learning stability and performance. These enhancements will strengthen the applicability of our model in real-world quality control systems, especially in agricultural and food processing industries.

References

[1] A. S. Tavares et al., "Leaf Disease Identification Using Texture and Color Features," IEEE Access, vol. 8, 2020.

[2] F. Bari et al., "Image-Based Leaf Classification Using Deep Learning," IEEE Access, vol. 8, 2020.

[3] L. Nanni, G. Maguolo, and A. Lumini, "Benchmarking Deep Learning Models for Plant Classification," Information Fusion, vol. 76, 2021.

[4] R. Krishnaswamy et al., "Leaf Texture Analysis Using GLCM for Plant Classification," IEEE Access, vol. 9, 2021.

[5] S. K. Sahu et al., "CNN-Based Classification Using Tabular Data: A Comparative Study," IEEE Access, vol. 9, 2021.

[6] T. Prilianti et al., "Comparative Study of First Order Optimizers on CNN for Histopathology Classification," Applied Sciences, vol. 12, no. 9, 2022.

[7] J. A. Nasution and M. Y. Mashor, "Leaf Shape and Texture Feature Extraction for Plant Classification," Indonesian Journal of Electrical Engineering and Computer Science, vol. 21, 2021.

[8] B. K. Singh et al., "Tabular CNN for Structured Data Classification," IEEE Xplore, 2021.

[9] M. Patel et al., "Evaluation of Optimizers for Deep Learning on Image Classification," IEEE Access, vol. 9, 2021.

[10] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016.

[11] L. Bottou, "Stochastic Gradient Descent Tricks," in Neural Networks: Tricks of the Trade, Springer, 2020.

[12] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," arXiv:1412.6980, 2015.

[13] T. Tieleman and G. Hinton, "Lecture 6.5—RMSProp," Coursera: Neural Networks for Machine Learning, 2012.

[14] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods," JMLR, vol. 12, pp. 2121–2159, 2011.

[15] L. Bottou, "Stochastic Gradient Descent Tricks," *Neural Networks: Tricks of the Trade*, Springer, 2020.

[16] Y. Nesterov, "A Method of Solving a Convex Programming Problem with Convergence Rate O(1/k^2)," *Soviet Mathematics Doklady*, 1983.

[17] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2015.

[18] T. Tieleman and G. Hinton, "Lecture 6.5—RMSProp," *Coursera: Neural Networks for Machine Learning*, University of Toronto, 2012.

[19] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," *Journal of Machine Learning Research*, vol. 12, pp. 2121–2159, 2011.

[20] K. He et al., "Deep Residual Learning for Image Recognition," IEEE CVPR, 2016.