Journal of Electronics, Electromedical Engineering, and Medical Informatics
Multidisciplinary: Rapid Review: Open Access Journal Vol. 7, No. 2, April 2025, pp: 330-340; eISSN: 2656-8632

RESEARCH ARTICLE OPEN ACCESS

Manuscript received December 8, 2024; Revised February 10, 2025; Accepted March 1, 2025; date of publication March 4 15, 2025
Digital Object Identifier (**DOI**): https://doi.org/10.35882/jeeemi.v7i2.652

How to cite: Kemal Crisannaufal, Wikky Fawwaz Al Maki, "Optimizing Support Vector Machine for Avocado Ripeness Classification Using Moth Flame Optimization", Journal of Electronics, Electromedical Engineering, and Medical Informatics, vol. 7, no. 1, pp. 220-230, January 2025.

Optimizing Support Vector Machine for Avocado Ripeness Classification Using Moth Flame Optimization

Kemal Crisannaufal[1], and Wikky Fawwaz Al Maki[1,2]

[1] School of Computing, Telkom University, Bandung, Indonesia
[2] HUMIC Research Center, Telkom University, Bandung, Indonesia

Corresponding author: Wikky Fawwaz Al Maki (e-mail: wikkyfawwaz@telkomuniversity.ac.id).

ABSTRACT Avocado is a fruit from Mexico and Central America that is widely distributed worldwide for production and consumption. In avocados, ripeness is crucial because it is the primary factor consumers consider, significantly influencing their purchasing decisions. The manual ripeness selection is inefficient and inconsistent, so the classification system is essential for determining ripeness due to its effectiveness and efficiency compared to manual selection. In this study, we aim to develop a model that can classify avocado ripeness using machine learning with optimization. The data consists of avocado images categorized into five ripeness stages: underripe, breaking, ripe (first stage), ripe (second stage), and overripe. We utilize a Support Vector Machine (SVM) for the classification. Instead of manually choosing the model's hyperparameters, we use Moth Flame Optimization (MFO) to optimize the SVM hyperparameters. The MFO ensures that the proposed model has optimal performance. For the input of SVM, we extract the HSV, GLCM, and HOG and apply PCA to the data. In this study, we use three SVM kernels: RBF, polynomial, and sigmoid. The MFO finds the model's hyperparameters based on kernel requirements, including C, gamma, degree, and coef0. The MFO-SVM obtains optimal performance with an accuracy of 82.55%, 82.68%, and 81.23% for SVM kernel RBF, polynomial, and sigmoid, respectively. The results show that our proposed model demonstrates adequate performance in identifying the ripeness levels of avocados. The MFO increases model performance on all evaluation metrics compared to the baseline model and can be an excellent strategy to improve model performance.

INDEX TERMS Avocado Ripeness, Classification, Moth Flame Optimization, Support Vector Machine

I. INTRODUCTION

Avocado is a fruit from Mexico and Central America, with a wide distribution of production and consumption worldwide [1]. Good taste and a variety of serving styles have increased avocado consumption globally, making avocado one of the famous "superfoods" in the world [2], [3]. In avocados, quality is the main factor that influences consumer purchasing decisions. Fruit quality standards are crucial because they affect fruit sales and marketing [2], [4]. One of the fruit quality standards is ripeness. To meet the quality standards of the avocado, especially its ripeness, the sorting process of the avocado is essential. Sorting ripeness manually by humans is inefficient as it demands extensive time and effort, and the results are inconsistent [5]. Therefore, the avocado ripeness classification system is vital for achieving more effective and efficient performance than the manual ripeness selection.

Many researchers have researched the classification of avocado ripeness. In [6], Xavier *et al.* classified avocado ripeness into five classes using AlexNet and ResNet-18. The model obtained an average accuracy of 76.9% for AlexNet and 78.4% for ResNet-18. Cruz and Ramirez [7] classified avocados into five classes based on ripeness level and avocado size using LabView and CNN. The study obtained an accuracy of 60%. In another study, Acevedo *et al.* [8] classified avocado ripeness into three classes using ANN with RGB and contrast from GLCM feature extraction, obtaining an accuracy of 88%. Although many studies have been conducted, some research still has limitations, especially regarding the model's performance.

Machine learning can be an effective method for classification. Support Vector Machine (SVM) is one of the machine learning algorithms that is robust and effective in making predictions based on data [9]. In [10], SVM was used to classify avocado ripeness based on electrical impedance spectroscopy, achieving 90% accuracy. Previous studies have also demonstrated the effectiveness of SVM in similar image classification problems. In [9], SVM obtained an accuracy of 95.97% and outperformed ResNet-50 in apple fruit classification using a bag of visual words (BoVW) with SIFT, SURF, and K-means clustering. In [11], Rahman *et al.* used SVM to classify tomato leaf diseases with GLCM feature extraction and obtained an average accuracy of 92.5% across all classes. In another study [12], Pothen and Pai classified rice leaf diseases using SVM with LBP and HOG feature extraction. HOG-SVM obtained an accuracy of 94.6%, whereas LBP-SVM obtained an accuracy of 90.23%. According to previous research, SVM provides excellent performance, so we employed the method in this study.

In machine learning for image classification, feature extraction is commonly used to capture patterns from the images. Features representing an image, such as shape, texture, and color, are often used to extract image characteristics [11-16]. Feature extraction techniques such as HSV, GLCM, and HOG have shown strong results in ripeness classification [13], [15], [16]. In [15], KNN with HSV effectively classifies the maturity of palm oil fruit with an average accuracy of 94,16%. In [16], HOG with KNN obtained 100% accuracy in classifying papaya ripeness. In [13], the combination of HSV and GLCM has excellent performance in classifying banana ripeness with 98,89% accuracy. These three methods have demonstrated effective performance in fruit ripeness classification. We adapted the methods for avocados as they have similar characteristics of changing color and texture with the ripeness phase.

In addition to feature extraction, hyperparameters are critical aspects of the machine learning model that affect its performance. Manual tuning hyperparameters is ineffective and highly resource-intensive [17]. To address this, many researchers use optimization algorithms to find the model's hyperparameters and to improve the overall model's performance [14], [18]. One of the optimization algorithms that can be used in the hyperparameters tuning process is Moth Flame Optimization (MFO). MFO is an optimization method based on the behavior of moths towards light sources [19]. The classification of fruit ripeness using MFO-SVM has not been explored widely previously. However, research [20] in predicting the mortality of Tilapia fish has shown that MFO can improve SVM model performance through hyperparameter selection. MFO was used to optimize C and gamma parameters in SVM, and it obtained better performance than SVM models without optimization with an accuracy of 99,98%. The results also surpassed SVM with other optimization algorithms, such as GA and PSO. The study shows that MFO can improve model performance by finding the optimal hyperparameter.

In this study, we aim to develop a model that can classify the ripeness of avocado fruit using the SVM method with MFO for optimization, which differentiates this study from the previous research. The MFO ensures that the proposed model demonstrates better results than the baseline model. The contributions of this study include the following:

a. Build the MFO-SVM model that can classify avocado ripeness
b. Provide knowledge about the performance comparison of the MFO-SVM and baseline SVM in various kernels
c. Provide knowledge on the effectiveness of MFO in enhancing the accuracy of the baseline model

II. MATERIALS AND METHODS

The MFO-SVM classification system was developed through several phases. These phases include preprocessing, data splitting, data augmentation, feature extraction, dimensionality reduction, model training with hyperparameter optimization, and evaluating the proposed model's performance. FIGURE 1 illustrates the development stages of the MFO-SVM classification system.

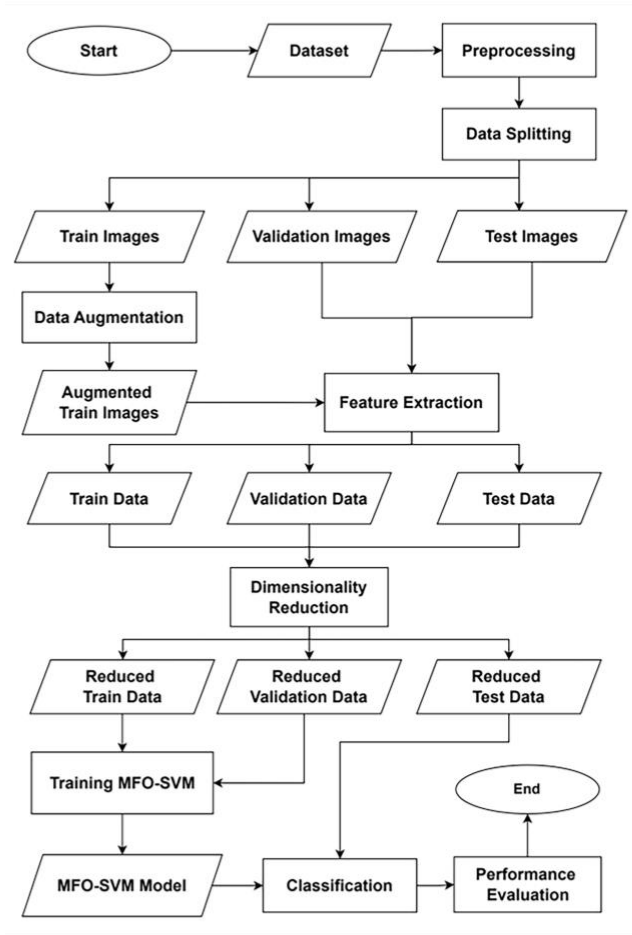


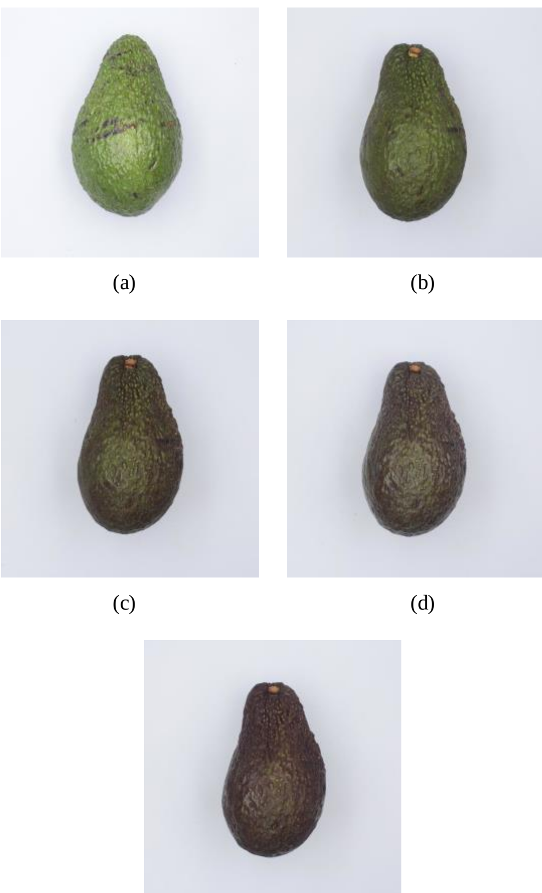
FIGURE 1. Methodology flowchart of our proposed model

A. DATASET

This study used the Hass Avocado Ripening Photographic data [21] (https://data.mendeley.com/datasets/3xd9n945v8/1), comprising 14,710 images. The dataset comprises images with sizes of 800x800 pixels. The dataset consists of five classes: the underripe class, which includes avocados with a firm texture and yellowish-green color; the breaking class, consisting of avocados with a hard texture, starting to darken; the ripe (first stage) class, which is avocado with purple spots and a softened texture; the ripe class (second stage), avocados with uniform purple color without damage (peak shelf life); and the overripe class, that is avocados that have passed the peak shelf life characterized by spots on the skin and stalk. The data distribution for each ripeness category is shown in TABLE I, while FIGURE 2 shows sample images for each class in the dataset.

TABLE I
Distribution of Dataset

Class	Number of Instances
Underripe	3568
Breaking	2228
Ripe (first stage)	2756
Ripe (second stage)	3294
Overripe	2864



(a) (b) (c) (d) (e)

FIGURE 2. (a) Underripe (b) Breaking (c) Ripe (first stage) (d) Ripe (second stage) (e) Overripe

B. PREPROCESSING

High-resolution images such as 800x800 pixels enhance visual details but are resource-intensive. Resizing image dimensions is a common technique to improve computational efficiency. Image resizing in preprocessing is an important stage to reduce computational complexity, standardize image dimensions, and retain essential spatial characteristics [22]. While smaller image sizes increase computational efficiency, they may also result in the loss of visual details. Therefore, it is important to achieve a balance between computational efficiency and retained information. In this study, we resized the images from 800x800 to 128x128 pixels, a size that improves computational efficiency while preserving most of the features in the image.

C. DATA AUGMENTATION

Data augmentation is a method that can enhance model performance by increasing the number of training data [23]. It can also be used to solve the class imbalance problem by increasing the amount of minority classes [24]. Data augmentation can be performed through two methods: geometric transformation (classic transformation) and techniques to generate new data using generative models (synthetic augmentation) [23]. In this study, we applied geometric transformations to augment the training data, especially for classes with a minority number. The transformations from image I to the transformed image T can be represented as $T(u,v) = I\big(t(u,v)\big)$, where $t(u,v)$ defines the mapping of the pixel coordinate in the original image I to the new coordinates in the transformed image T [25]. The transformations can improve the model's performance by adding more variations to the data, helping the model recognize new patterns without collecting additional data. In this study, the images were rotated, shifted, and flipped until each class in the training data had a total of 2500 images. TABLE 2 compares the number of training data before and after augmentation for each class.

TABLE 2
Comparison of the Number of Training Data Before and After Augmentation

Ripeness Classes	Original Data	Augmented Data
Underripe	2497	2500
Breaking	1560	2500
Ripe (first stage)	1929	2500
Ripe (second stage)	2306	2500
Overripe	2005	2500

D. FEATURE EXTRACTION

Feature extraction is needed to produce data features that describe the pattern or characteristics of the image. We use the avocado fruit's color, texture, and shape features through HSV, GLCM, and HOG feature extraction.

1) HSV

HSV feature extraction can capture the color changes in each phase of avocado ripeness. HSV is a color space consisting of hue, saturation, and value. Hue is a feature that describes the chromatic properties of a color related to the dominant wavelength of the color [26]. Saturation measures the purity of color, with lower saturation indicating that the color contains grayer [26]. Value is the brightness or the intensity of the color [26]. HSV is good at handling images with different lighting conditions using color, chroma, and values [27].

This study used the histogram and color moments for each channel. Color histogram distributes the color into several ranges of values (bins), whereas color moments are a measure that can differentiate images based on their color characteristics [28]. The histogram of an image is denoted as $H = [h_1, h_{2,} \ldots, h_n]$, where n denotes the number of bins, $h_i = N_i/N$ represents the probability of a pixel in the image falling within the i-th bin, N_i refers to the total pixel count in the i-th bin, and N represents the total pixel count in the image [29]. We extracted the histogram channel H for 180 bins, whereas channels S and V are 256 bins. Additionally, the color moments were computed for each channel. The mean, standard deviation, and skewness are calculated from the images using Eq. (1), Eq. (2), and Eq. (3), respectively [28]. A total of 701 data features were obtained from HSV feature extraction.

$$M_j = \sum_{i=1}^{N} \frac{1}{N} P_{j,i} \tag{1}$$

$$\sigma_j = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (P_{j,i} - M_j)^2} \tag{2}$$

$$S_j = \sqrt[3]{\frac{1}{N} \sum_{i=1}^{N} (P_{j,i} - M_j)^3} \tag{3}$$

M_j, σ_j, and S_j represent the mean, standard deviation, and skewness of the j-th color channel, respectively. $P_{j,i}$ represents the value of the i-th pixel in the j-th color channel, and N denotes the total number of pixels in the image.

2) GRAY LEVEL CO-OCCURRENCE MATRIX (GLCM)

GLCM captures image texture changes and patterns using second-order statistics [30]. In GLCM, the image is mapped into a table representing the frequency of occurrence of pixel value pairs at a specified distance and angle [31]. The second-order statistics used in this study include ASM, energy, dissimilarity, contrast, correlation, and homogeneity. ASM and energy represent the uniformity of the image represented by Eq. (4) [30] and Eq. (5) [31]. Dissimilarity represents the diversity of textures in the image, represented by Eq. (6) [31]. Contrast measures the difference between the highest and the lowest pixel values in a group of pixels in the image, represented by Eq. (7) [31]. Correlation represents the consistency of image textures, represented by Eq. (8) [30]. Homogeneity measures the similarity of pixel pairs in the image, represented by Eq. (9) [31]. Before performing GLCM feature extraction, the image is converted into grayscale. In this study, we used four angles consisting of [0°. 45°, 90°, 135°] with a distance of [1, 2, 3, 4]. The GLCM feature extraction resulted in 6x4x4=96 data features for each image.

$$ASM = \sum_{i,j=0}^{L-1} P^2(i,j) \tag{4}$$

$$Energy = \sqrt{ASM} \tag{5}$$

$$Dissimilarity = \sum_{i,j=0}^{L-1} |i-j| P(i,j) \tag{6}$$

$$Contrast = \sum_{i,j=0}^{L-1} P(i,j)(i-j)^2 \tag{7}$$

$$Correlation = \sum_{i,j=0}^{L-1} P(i,j) \left[\frac{(i-\mu_i)((j-\mu_j)}{\sqrt{(\sigma_i{}^2)(\sigma_j{}^2)}} \right] \tag{8}$$

$$Homogenity = \sum_{i,j=0}^{L-1} \frac{P(i,j)}{1+(i-j)^2} \tag{9}$$

In Eq. (4) to Eq. (9), i and j represent the row and column indices in the GLCM, and L represents the number of gray levels in the image. $P(i,j)$ represents the elements of the GLCM at the (i,j) position. Additionally, μ_i and μ_j represent the mean values of the i and j, respectively, while σ_i and σ_j represent the standard deviation of the values i and j in the GLCM [32].

3) HISTOGRAM OF ORIENTED GRADIENTS (HOG)

HOG is a method that can capture an image's shape and appearance (texture, pattern) through simple computation, making it faster and more efficient than other feature descriptors such as SIFT and LBP [33]. In HOG, histograms are used to represent the features that capture the directional change information of the edge [34]. HOG works by dividing the image into small cells and organizing these cells into blocks [35]. The magnitude and orientation of the gradient values are calculated, and the orientation values are stored as a histogram for each block [35]. The magnitude $M(x,y)$ and orientation (x,y) of gradients are calculated using Eq. (10) and Eq. (11), respectively [35]. The resulting histograms are put together to produce a data feature vector [35]. This study calculated HOG with a cell size of 8x8 pixels and a block size of 2x2 cells, using nine orientations, resulting in 8100 data features. Before performing HOG, we convert the image's color space from RGB into grayscale. The visualization of the HOG feature extraction is shown in FIGURE 3.

$$M(x,y) = \sqrt{|G_x{}^2| + |G_y{}^2|} \tag{10}$$

$$(x,y) = \tan^{-1}\left(\frac{G_x}{G_y}\right) \tag{11}$$

G_x and G_y denote the gradient along the x and y directions, respectively.

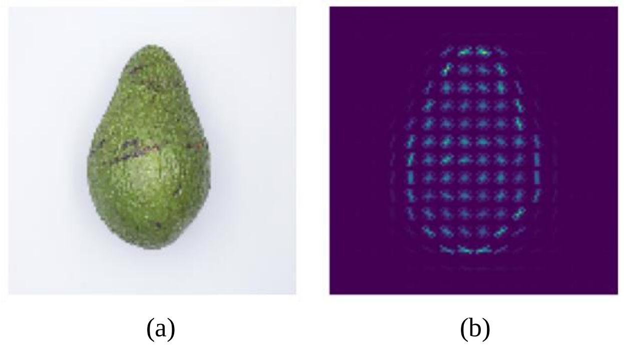


FIGURE 3. **(a) Input Image (b) Visualization of HOG Feature Extraction**

E. DIMENSIONALITY REDUCTION

In this study, the feature extraction process resulted in 8,897 features. Training and optimization on data with 8,897 features will require many resources. Therefore, we applied principal component analysis (PCA) to transform data into a lower dimension. PCA is a method for reducing high, correlated data features into fewer features, referred to as principal components [36]. PCA works by averaging the data, calculating the covariance matrix to evaluate the dependencies and correlations between data features, and decomposing the matrix using eigenvalues. The eigenvalues are sorted in descending order to determine the *n* principal components [36]. In this study, PCA reduced the data dimension from 8,897 features to 150 principal components, making training and optimization more efficient while preserving essential features from the data.

F. SUPPORT VECTOR MACHINE (SVM)

SVM is an efficient and effective method for solving classification problems [9]. Using kernel functions, SVM can efficiently solve classification problems in high-dimensional data spaces [9]. SVM can classify data with linear characteristics and non-linear characteristics [37]. In linear SVM, SVM tries to find the largest hyperplane that separates different data features, as in FIGURE 4. For data that cannot be linearly separated, SVM will calculate data in low-dimensional space and then map it to high-dimensional space using a kernel function so that the optimal hyperplane can still be constructed in high-dimensional space [38]. The kernel function will determine how the likeness between the data points is measured [39]. In this study, we used RBF, polynomial, and sigmoid kernels, calculated using Eq. (12), Eq. (13), and Eq. (14), respectively [39], [40]. The variables v_i and v_j are data points, γ represents the gamma parameter, *d* denotes the degree of the polynomial, and *r* denotes the coef0 in the scikit-learn library.

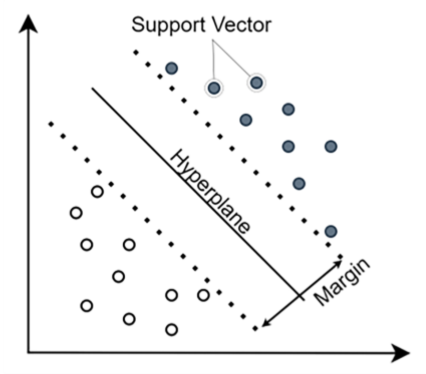


FIGURE 4. **Illustration of the SVM Hyperplane**

$$RBF = \exp\left(-\gamma\left\|v_i - v_j\right\|^2\right) \tag{12}$$

$$Poly = (\gamma v_i^T v_j + r)^d \tag{13}$$

$$Sigmoid = \tanh(\gamma v_i^T v_j + r) \tag{14}$$

G. MOTH FLAME OPTIMIZATION

This study used MFO (Moth Flame Optimization) for the hyperparameter optimization of the SVM. The MFO algorithm is inspired by moths' behavior toward light sources [19]. Within the search space, moths serve as potential optimal hyperparameters and update their positions based on a reference point represented by the best moths, the flames [19]. This updated position is determined using the logarithmic spiral represented by Eq. (15) [19].

$$S(M_{i,}F_j) = D_i . e^{bt} . \cos(2\pi t) + F_j \tag{15}$$

Where D_i represents the closeness of the *i*-th moth to the *j*-th flame, *b* represents a fixed number that determines the shape of the logarithmic spiral, and *t* is a random number in [-1, 1]. The *t* value determines how close the next position is to the flame. D_i is calculated using Eq. (16), where M_i indicates the i-th moth, and F_j indicates the j-th flame [19].

$$D_i = \left|F_j - M_i\right| \tag{16}$$

The value of *t* affects the exploration and exploitation process. Exploration is performed when the next moth's position is not between the moth and the flame. In contrast, exploitation occurs when the next moth's position is between the moth and the flame [19], as shown in FIGURE 5. Moths update their position based on *n* different positions in search space, which can degrade the exploitation of the best potential solution [19]. As the iteration progresses, the number of flames can be reduced using Eq. (17) [19].

$$flame\ no = round\left(N - l.\frac{N-1}{T}\right) \tag{17}$$

Where l denotes the current iteration, N denotes the maximum number of flames of the current iteration, and T denotes the maximum iteration.

In this study, moths will represent the set of possible optimal hyperparameters for the SVM. We used accuracy against the validation data as the fitness value. In each iteration, the moth will be evaluated through classification against the validation data, and then the flames are updated based on the best moth. The moth position is updated using the spiral movement function to reach a better solution marked by the flames. This process continues until the maximum iteration is reached. The moth that gives the best accuracy on the validation data during iteration will be the most optimal hyperparameter set for MFO-SVM. The process of optimizing the SVM hyperparameter by MFO is shown in FIGURE 6.

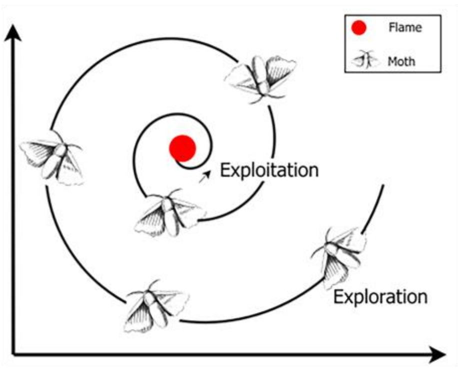


FIGURE 5. Illustration of the Exploration and Exploitation Process in MFO

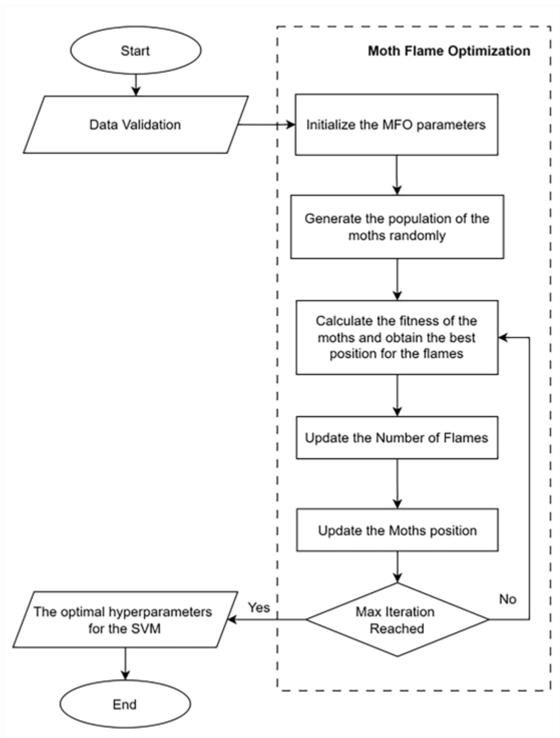


FIGURE 6. MFO Algorithm for SVM Hyperparameter Optimization

H. PERFORMANCE EVALUATION

We computed four main evaluation metrics in classification to evaluate the model. Accuracy measures how well the model predicts all data by calculating the ratio between correct predictions in all ripeness categories and total instances tested, calculated using Eq. (18) [41]. Precision evaluates the ratio between the accurate predictions for a specific ripeness category and the total predictions for that category, calculated using Eq. (19) [41]. Recall evaluates the ratio between the accurate predictions for a specific ripeness category and the total data in that category, calculated using Eq. (20) [41]. F1-score provides information about the balance between precision and recall in one metric, calculated using Eq. (21) [41].

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \tag{18}$$

$$Precision = \frac{TP}{TP+FP} \tag{19}$$

$$Recall = \frac{TP}{TP+FN} \tag{20}$$

$$F1-Score = 2x\frac{Precision\ x\ Recall}{Precision + Recall} \tag{21}$$

In the equations, TP denotes true positive, TN denotes true negative, FP denotes false positive, and FN denotes false negative.

III. RESULT

This study classified avocado ripeness into five stages. The data were split with a proportion of 70:15:15 for training, validation, and testing. The split process resulted in 10,297 training images, 2,207 validation images, and 2,206 test images. We resized the images into 128x128 pixels and applied data augmentation to training data. After that, we extracted HSV, GLCM, and HOG feature extraction and obtained 8,897 features. We reduced the features to 150 using PCA to improve the model training efficiency. The SVM performs classification with hyperparameters selected using the MFO algorithm.

We evaluated the baseline SVM model (SVM with default hyperparameters) to compare its result with the optimized model. In the baseline model, the RBF kernel is superior to polynomial and sigmoid kernels. The RBF and sigmoid baseline model obtained performance above 80% for each evaluation metric with an accuracy of 81.60% for the RBF kernel and 80.92% for the sigmoid kernel. In contrast, the polynomial kernel obtained an accuracy of 67.54%, with accuracy, recall, and F1-score below 70%. This result indicates that the baseline SVM with a polynomial kernel has not yet performed well in identifying avocado ripeness. The MFO optimization process will improve the polynomial SVM performance. TABLE 3 presents the evaluation results of the baseline SVM model.

TABLE 3
Performance of Baseline SVM Model

Kernel	Accuracy	Precision	Recall	F1-Score
RBF	81.60%	81.83%	81.60%	81.54%
Polynomial	67.54%	75.13%	67.54%	67.34%
Sigmoid	80.92%	81.05%	80.92%	80.86%

This study also evaluated the effect of HSV, GLCM, and HOG feature extraction on the baseline model performance. Extracting HSV, GLCM, or HOG features individually results in HOG and HSV demonstrating a more significant influence with an accuracy of 76.79% for HOG and 75.88% for HSV using the RBF kernel. In comparison, GLCM obtained an accuracy of 65.23%. Using two of three considered feature extractions gives better results than a single feature extraction technique. Combining two of three features demonstrates that HSV+HOG significantly influences the model's performance with an accuracy of 81.60%, followed by GLCM+HOG with an accuracy of 78.56% and HSV+GLCM with an accuracy of 76.93% using the RBF kernel. HSV+HOG performs similarly to the HSV+GLCM+HOG in the RBF kernel, but the HSV+GLCM+HOG obtained better accuracy for polynomial and sigmoid kernels. TABLE 4 compares the performance of individual feature extraction and combined feature extraction methods.

TABLE 4
Performance of Baseline SVM Model Across Different Feature Extraction Techniques

Feature Extraction	SVM Kernel	Baseline Model Performance			
		Accuracy	Precision	Recall	F1-Score
HSV	RBF	75.88%	75.93%	75.88%	75.81%
	Polynomial	73.16%	73.95%	73.16%	73.24%
	Sigmoid	60.06%	60.47%	60.06%	59.79%
GLCM	RBF	65.23%	64.50%	65.23%	64.13%
	Polynomial	55.12%	60.41%	55.12%	55.83%
	Sigmoid	34.13%	38.85%	34.13%	33.47%
HOG	RBF	76.79%	76.65%	76.79%	76.34%
	Polynomial	52.13%	64.77%	52.13%	47.57%
	Sigmoid	73.48%	73.11%	73.48%	73.09%
HSV+ GLCM	RBF	76.93%	77.08%	76.93%	76.90%
	Polynomial	74.71%	75.72%	74.71%	74.77%
	Sigmoid	58.66%	59.12%	58.66%	58.42%
HSV +HOG	RBF	81.60%	81.79%	81.60%	81.53%
	Polynomial	66.73%	74.47%	66.73%	66.40%
	Sigmoid	80.64%	80.77%	80.64%	80.58%
GLCM+ HOG	RBF	78.56%	78.67%	78.56%	78.33%
	Polynomial	53.67%	66.29%	53.67%	49.56%
	Sigmoid	76.20%	76.25%	76.20%	75.97%
HSV+ GLCM+ HOG	RBF	81.60%	81.83%	81.60%	81.54%
	Polynomial	67.54%	75.13%	67.54%	67.34%
	Sigmoid	80.92%	81.05%	80.92%	80.86%

In this study, MFO optimizes SVM's hyperparameters. It explores the C parameter within the range of [0.0001, 10000], gamma within the range of [0.0001, 10], degree within the range of [2, 6], and coef0 within the range of [-1, 1]. The number of moths used in this study was 50, with 15 iterations to search the optimal hyperparameter. A total of 750 possible hyperparameters will be evaluated over 15 iterations, with each moth updating its position in search of the optimal hyperparameters. The moth will represent the set of hyperparameters to be evaluated against the validation data. In each iteration, the hyperparameters will be updated based on flames (the best hyperparameters), which provides the best accuracy against the validation data. Moth will attempt to maximize the model's accuracy, and at the end of the process, the best hyperparameter will be returned to be used by the SVM model. TABLE 5 shows the optimization results generated by MFO for each kernel.

TABLE 5
Results of MFO Optimization for SVM Model

Kernel	C	Gamma	Degree	Coef0
RBF	8.07887	0.0001	-	-
Polynomial	0.54199	0.00021	4	0.87164
Sigmoid	12.53155	0.0001	-	-0.8974

In the RBF kernel, PCA+SVM obtained an accuracy of 81.64%, whereas PCA+MFO-SVM obtained an accuracy of 82.55%. The PCA+MFO-SVM achieved 0.95% higher than the baseline SVM model and 0.91% higher than the PCA+SVM model. TABLE 6 shows the results of MFO-SVM with the RBF kernel.

TABLE 6
Performance of MFO-SVM with RBF Kernel

Methods	Model Performance			
	Accuracy	Precision	Recall	F1-Score
PCA+SVM	81.64%	81.94%	81.64%	81.54%
PCA+MFO-SVM	82.55%	82.71%	82.55%	82.50%

In the polynomial kernel, PCA improves the accuracy of the baseline model by 10.88% to 78.42%. SVM with MFO achieved an accuracy of 82.68%, an increase of 4.26% compared to the PCA+SVM model. TABLE 7 shows the results of MFO-SVM with the polynomial kernel.

TABLE 7
Performance of MFO-SVM with Polynomial Kernel

Methods	Model Performance			
	Accuracy	Precision	Recall	F1-Score
PCA+SVM	78.42%	79.49%	78.42%	78.33%
PCA+MFO-SVM	82.68%	82.82%	82.68%	82.61%

In the sigmoid kernel, PCA+SVM obtained an accuracy of 68.36%, whereas PCA+MFO-SVM achieved an accuracy of 81.23%, 12.87% higher than the PCA+SVM model and 0.31% than the baseline model. Although the MFO-SVM model on the sigmoid kernel has almost a similar performance as the baseline model, MFO-SVM uses a much smaller number of features, making it more efficient. TABLE 8 presents the results of the sigmoid MFO-SVM model.

The MFO-SVM with a polynomial kernel achieved the best result compared to other kernels. The polynomial MFO-SVM can accurately predict 506 out of 535 underripe classes, 241 out of 334 breaking classes, 353 out of 414 ripe (first stage) classes, 367 out of 494 ripe (second stage) classes, and 357 out of 429 overripe avocados. Overall, the model can correctly predict 1,824 out of 2,206 data. FIGURE 7 shows the confusion matrix for polynomial MFO-SVM.

TABLE 8
Performance of MFO-SVM with Sigmoid Kernel

Methods	Model Performance			
	Accuracy	Precision	Recall	F1-Score
PCA+SVM	68.36%	69.04%	68.36%	68.56%
PCA+MFO-SVM	81.23%	81.44%	81.23%	81.21%

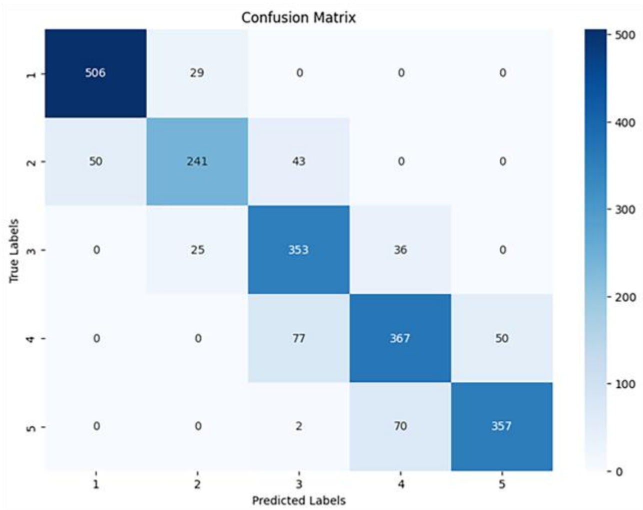


FIGURE 7. **The Confusion Matrix of MFO-SVM with Polynomial Kernel**

IV. DISCUSSION

This study observed five stages of avocado ripeness. We employed manual feature extraction and machine learning techniques to identify the avocado ripeness. We used HSV, GLCM, and HOG to extract patterns from the images. Based on TABLE 4, selecting feature extraction techniques is crucial because it affects the classification performance. Using a single feature from the image, such as color or texture, is insufficient to capture the characteristics of avocado ripeness. Extracting the color and texture features in combination provides better performance than a single feature extraction. Extracting HSV, GLCM, and HOG in combination obtained the best result that provides characteristics of the avocado's color, texture, and shape that can better identify its ripeness. This combination more effectively identifies the avocado ripeness patterns than extracting each feature individually.

The feature extraction process resulted in 8,897 features, which will require many resources for training and optimization. In this study, PCA was applied before the optimization process with MFO. Although the features were significantly reduced to 150, the optimized model's accuracy is better than the baseline model using all data. This result shows that PCA simplifies model complexity by reducing feature dimensions and improving classification performance through the properly tuned model. PCA reduces the computational load and improves the efficiency of the model.

The hyperparameters significantly influence the model's effectiveness. Hyperparameter values such as C and gamma that are too large or too small can cause the model to be overfitting or underfitting, so the model's performance is not optimal [42]. In the PCA+SVM model on the sigmoid kernel, the default hyperparameters are unsuitable for the data used, so the model tends to be underfitting. MFO can solve this problem by finding the hyperparameter combination that gives the best performance to the model. In this study, the MFO-SVM model can increase the PCA+SVM model's accuracy by 0.91%, 4.26%, and 12.87% in RBF, polynomial, and sigmoid kernel, respectively, and increase the baseline SVM model's accuracy by 0.95%, 15.14%, and 0.31% in RBF, polynomial and sigmoid kernels, respectively. In the sigmoid kernel, MFO-SVM performs similarly to the baseline model performance. However, MFO-SVM is still better because it is more efficient by using a much smaller number feature. The optimized model outperforms the baseline model for each kernel. MFO consistently improves model performance by more systematically selecting optimal hyperparameters for the model than using manual search.

The proposed model has performed well in solving the avocado ripeness problem. The MFO-SVM with a polynomial kernel has the highest accuracy compared to other kernels. Based on FIGURE 7, our proposed model better identifies the underripe ripeness stage of avocado fruit. The underripe avocados are easier to identify due to their consistent green color and contrast compared to other ripeness classes. However, due to the minimal visual difference, our proposed model still needs to work on distinguishing other ripeness classes. In this study, the dataset consisted of images of avocados observed daily to capture changes in ripeness, so the differences between the classes were subtle and more difficult to distinguish visually. As a result, most prediction errors occurred because the model predicted one level higher or lower than the actual ripeness stage. This study still needs to improve classification accuracy, especially in ripeness classes with minimal visual differences between classes.

Since the classification of avocado fruit ripeness has been discussed, we compared our study with several previous studies, as shown in TABLE 9. This study has surpassed the results of [6] using AlexNet and ResNet-18, which were evaluated with a similar dataset. The study also split the data with a ratio of 75:15:15 for training, validation, and testing. ResNet-18 provided the best result, with an average accuracy of 78.4%. Our proposed method obtained 4% higher accuracy than that study. The improved accuracy in a similar dataset shows that our approach through machine learning and feature extraction techniques is more effective in classifying avocado ripeness. The performance of our proposed model also surpasses that of [7], where in that study, researchers classified avocados based on their ripeness level using LabView and

CNN with an accuracy of 60%. In addition to classifying ripeness, that research also classifies avocados based on their size, which needs to be discussed in this study. This study also had better results compared to [43], which classified avocado maturity into three classes using KNN with L*a*b* and texture features, which obtained 81.9% accuracy.

Nevertheless, this study cannot outperform the research [8] and [44]. In [8], the researchers classified avocados into three ripeness classes using RGB and the contrast feature from GLCM with ANN and obtained 88% accuracy. Although the study obtained higher accuracy than our model, it classified ripeness into fewer classes. Similarly, our study did not surpass [44], which classified avocado ripeness into five classes using a Multi-Channel Hybrid Deep Neural Network (MCHDNN) combining VGG-16 and EfficientNetB0. The study obtained an accuracy of 90.18%. However, our model was able to outperform the single-channel models of both VGG-16 and EfficientNetB0 in [44], where the single-channel models obtained an accuracy of 82.62% for VGG-16 and 82.47% for EfficientNetB0.

Our proposed MFO-SVM model with HSV, GLCM, and HOG feature extraction has demonstrated effective results in classifying avocado ripeness. Compared to previous studies, our proposed model obtained competitive performance. Although our result did not outperform the performance in some studies, our research provides new insights into the effectiveness of the MFO-SVM for avocado ripeness classification.

IV. CONCLUSION

The MFO-SVM model was developed to classify avocados' ripeness into five stages. This study finds that the MFO-SVM model performs well in classifying avocado ripeness. MFO-SVM with the polynomial kernel using PCA obtained the best performance with an accuracy of 82.68%, followed by the RBF kernel with an accuracy of 82.55%, and the sigmoid kernel obtained an accuracy of 81.23%. This study finds that the combination of HSV, GLCM, and HOG feature extraction performs better than the individual feature extractions. In addition, we also find that the PCA simplifies the model by reducing the features to fewer numbers, making training more efficient with better performance than using all features through optimization. This study also finds that MFO consistently found optimal hyperparameters, thus improving performance for each SVM kernel. Hyperparameters search using MFO is proven to be one of the promising strategies for improving overall model performance. Although our model obtained good results, it still has limitations in predicting the ripeness stage if the visual variations between the ripeness stages are minimal. In the future, we will explore different feature extraction and optimization techniques and consider using deep learning to improve the model's accuracy.

TABLE 9
Evaluation Comparison of the MFO-SVM Model with Existing Research

Authors	Method	Result
Xavier *et al.* [6]	AlexNet and ResNet-18	The model obtained an average accuracy of 78.4% for ResNet-18 and 76.9% for AlexNet. Achieved high results when applying a margin of error of one ripeness level.
Cruz and Ramirez [7]	LabView and CNN	The model achieved 60% accuracy, with lighting conditions remaining a challenge.
Acevedo *et al.* [8]	ANN using RGB and GLCM contrast features	The model obtained a high accuracy of 88%. Among the three classes, green, unripe, and ripe, the prediction of the green class achieves 100% accuracy.
Vazquez *et al.* [43]	KNN using L*a*b* and texture features	The model achieved an accuracy of 81.9%, with the a* channel in L*a*b* being the most influential feature.
Nuanmeesri [44]	Multi-Channel Hybrid Deep Neural Networks (MCHDNN) combining VGG-16 and EfficientNetB0	The model obtained an effective result with an accuracy of 90.18%. Among the five classes, firm, breaking, ripe, overripe, and rotten, the ripe class has the highest classification accuracy.
This study	MFO-SVM using HSV, GLCM, and HOG features	Using the polynomial kernel, the model obtained an accuracy of 82.68%. Most errors occurred because the model predicted one level higher or lower than the actual ripeness stage.

ACKNOWLEDGMENT

The authors are grateful to Telkom University for supporting this research.

REFERENCES

[1] N. J. Salazar-López *et al.*, "Avocado fruit and by-products as potential sources of bioactive compounds," *Food Research International*, vol. 138, p. 109774, 2020, doi: https://doi.org/10.1016/j.foodres.2020.109774.
[2] P. Rodríguez, I. Soto, J. Villamizar, and A. Rebolledo, "Fatty Acids and Minerals as Markers Useful to Classify Hass Avocado Quality: Ripening Patterns, Internal Disorders, and Sensory Quality," *Horticulturae*, vol. 9, no. 4, Apr. 2023, doi: 10.3390/horticulturae9040460.
[3] K.-M. Huang, Z. Guan, T. Blare, and A. M. Hammami, "Global Avocado Boom," *Choices*, vol. 38, pp. 1–9, 2023.
[4] M. Ahmed *et al.*, "Few shot learning for avocado maturity determination from microwave images," *J Agric Food Res*, vol. 15, Mar. 2024, doi: 10.1016/j.jafr.2024.100977.
[5] V. L. Tran, T. N. C. Doan, F. Ferrero, T. Le Huy, and N. Le-Thanh, "The Novel Combination of Nano Vector Network Analyzer and Machine Learning for Fruit Identification and Ripeness Grading," *Sensors*, vol. 23, no. 2, Jan. 2023, doi: 10.3390/s23020952.
[6] P. Xavier, P. M. Rodrigues, and C. L. M. Silva, "Shelf-Life Management and Ripening Assessment of 'Hass' Avocado (Persea americana) Using Deep Learning Approaches," *Foods*, vol. 13, no. 8, Apr. 2024, doi: 10.3390/foods13081150.

[7] J. E. de la Cruz and O. J. Vera Ramirez, "Convolutional neural networks for the Hass avocado classification using LabVIEW in an agro-industrial plant," in *2020 IEEE XXVII International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, 2020, pp. 1–4. doi: 10.1109/INTERCON50315.2020.9220246.

[8] C. A. Jaramillo-Acevedo, W. E. Choque-Valderrama, G. E. Guerrero-Álvarez, and C. A. Meneses-Escobar, "Hass avocado ripeness classification by mobile devices using digital image processing and ANN methods," *International Journal of Food Engineering*, vol. 16, no. 12, Dec. 2020, doi: 10.1515/ijfe-2019-0161.

[9] S. Adige, R. Kurban, A. Durmuş, and E. Karaköse, "Classification of apple images using support vector machines and deep residual networks," *Neural Comput Appl*, vol. 35, no. 16, pp. 12073–12087, Jun. 2023, doi: 10.1007/s00521-023-08340-3.

[10] M. Islam, K. Wahid, and A. Dinh, "Assessment of ripening degree of avocado by electrical impedance spectroscopy and support vector machine," *J Food Qual*, vol. 2018, 2018, doi: 10.1155/2018/4706147.

[11] S. U. Rahman, F. Alam, N. Ahmad, and S. Arshad, "Image processing based system for the detection, identification and treatment of tomato leaf diseases.," *Multimed Tools Appl*, vol. 82, no. 6, pp. 9431–9445, Mar. 2023, doi: 10.1007/s11042-022-13715-0.

[12] M. E. Pothen and M. L. Pai, "Detection of Rice Leaf Diseases Using Image Processing," in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, 2020, pp. 424–430. doi: 10.1109/ICCMC48092.2020.ICCMC-00080.

[13] J. Enríquez and M. Macas, "Artificial Intelligence-Based Banana Ripeness Detection," in *Applied Technologies*, Botto-Tobar, Miguel, Z. V. Marcelo, S. Montes Le'on, P. Torres-Carrion, and B. Durakovic, Eds., Cham: Springer Nature Switzerland, 2023, pp. 197–211. doi: https://doi.org/10.1007/978-3-031-24985-3_15.

[14] Z. Xu, Q. Zhao, Y. Zhang, Y. Zhang, and T. Shen, "Apple Grading Method Based on GA-SVM," in *The 10th International Conference on Computer Engineering and Networks*, L. Qi, L. Xiadong, S. Tao, and Q. Xuesong, Eds., Singapore: Springer Singapore, 2021, pp. 79–89. doi: https://doi.org/10.1007/978-981-15-8462-6_9.

[15] M. Y. Pusadan, I. Safitri, and Wirdayanti, "The Image Extraction Using the HSV Method to Determine the Maturity Level of Palm Oil Fruit with the k-nearest Neighbor Algorithm," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 6, pp. 1448–1456, Dec. 2023, doi: 10.29207/resti.v7i6.5558.

[16] S. K. Behera, A. K. Rath, and P. K. Sethy, "Maturity status classification of papaya fruits based on machine learning and transfer learning approach," *Information Processing in Agriculture*, vol. 8, no. 2, pp. 244–250, Jun. 2021, doi: 10.1016/j.inpa.2020.05.003.

[17] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, 2020, doi: https://doi.org/10.1016/j.neucom.2020.07.061.

[18] A. J. Makrufi and W. F. A. Maki, "Support Vector Machine with Firefly Optimization Algorithm for Apple Fruit Disease Classification," *MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, vol. 22, no. 1, pp. 177–188, 2022, doi: https://doi.org/10.30812/matrik.v22i1.2365.

[19] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl Based Syst*, vol. 89, pp. 228–249, 2015, doi: https://doi.org/10.1016/j.knosys.2015.07.006.

[20] A. A. Ewees, A. A. Hemedan, A. E. Hassanien, and A. T. Sahlol, "Optimized support vector machines for unveiling mortality incidence in Tilapia fish," *Ain Shams Engineering Journal*, vol. 12, no. 3, pp. 3081–3090, Sep. 2021, doi: 10.1016/j.asej.2021.01.014.

[21] P. Xavier, P. Rodrigues, and C. L. M. Silva, "'Hass' Avocado Ripening Photographic Dataset," *Mendeley Data*, vol. 1, 2024, doi: 10.17632/3xd9n945v8.1.

[22] P. P. Tumpa and Md. S. Islam, "Lightweight Parallel Convolutional Neural Network With SVM Classifier for Satellite Imagery Classification," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 11, pp. 5676–5688, 2024, doi: 10.1109/TAI.2024.3423813.

[23] S. Jain, G. Seth, A. Paruthi, U. Soni, and G. Kumar, "Synthetic data augmentation for surface defect detection and classification using deep learning," *J Intell Manuf*, vol. 33, no. 4, pp. 1007–1020, Apr. 2022, doi: 10.1007/s10845-020-01710-x.

[24] C. Guo, F. Benitez-Quiroz, Q. Feng, and A. Martinez, "Structural Similarity: When to Use Deep Generative Models on Imbalanced Image Dataset Augmentation," Mar. 2023, [Online]. Available: http://arxiv.org/abs/2303.04854

[25] R. Balestriero, I. Misra, and Y. Lecun, "A Data-Augmentation Is Worth A Thousand Samples: Analytical Moments And Sampling-Free Training," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2022, pp. 19631–19644. Accessed: Dec. 20, 2024. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/7c080cab957edab671ac49ae11e51337-Paper-Conference.pdf

[26] D. Giuliani, "Metaheuristic Algorithms Applied to Color Image Segmentation on HSV Space," *J Imaging*, vol. 8, no. 1, Jan. 2022, doi: 10.3390/jimaging8010006.

[27] H. Kim, H. Lee, S. Ahn, W. K. Jung, and S. H. Ahn, "Broken stitch detection system for industrial sewing machines using HSV color space and image processing techniques," *J Comput Des Eng*, vol. 10, no. 4, pp. 1602–1614, Aug. 2023, doi: 10.1093/jcde/qwad069.

[28] W. K. Mutlag, S. K. Ali, Z. M. Aydam, and B. H. Taher, "Feature Extraction Methods: A Review," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Aug. 2020. doi: 10.1088/1742-6596/1591/1/012028.

[29] J. Han and K.-K. Ma, "Fuzzy color histogram and its use in color image retrieval," *IEEE Transactions on Image Processing*, vol. 11, no. 8, pp. 944–952, 2002, doi: 10.1109/TIP.2002.801585.

[30] Q. Liu, Y. Song, Q. Tang, X. Bu, and N. Hanajima, "Wire rope defect identification based on ISCM-LBP and GLCM features," *Visual Computer*, vol. 40, no. 2, pp. 545–557, Feb. 2024, doi: 10.1007/s00371-023-02800-6.

[31] A. K. Aggarwal, "Learning Texture Features from GLCM for Classification of Brain Tumor MRI Images using Random Forest Classifier," *WSEAS TRANSACTIONS ON SIGNAL PROCESSING*, vol. 18, pp. 60–63, Apr. 2022, doi: 10.37394/232014.2022.18.8.

[32] I. Nurcahyati, T. H. Saragih, A. Farmadi, D. Kartini, and Muliadi, "Classification of Lung Disease in X-Ray Images Using Gray Level Co-Occurrence Matrix Method and Convolutional Neural Network," *Journal of Electronics, Electromedical Engineering, and Medical Informatics*, vol. 6, no. 4, pp. 332–342, Aug. 2024, doi: 10.35882/jeeemi.v6i4.457.

[33] Greeshma K V and Dr. J. Viji Gripsy, "Image Classification using HOG and LBP Feature Descriptors with SVM and CNN," *International Journal of Engineering Research & Technology (IJERT)*, vol. 8, no. 04, 2020.

[34] L. Zhang, W. Zhou, J. Li, J. Li, and X. Lou, "Histogram of Oriented Gradients Feature Extraction Without Normalization," in *2020 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, 2020, pp. 252–255. doi: 10.1109/APCCAS50809.2020.9301715.

[35] U. D. Dixit, M. S. Shirdhonkar, and G. R. Sinha, "Automatic logo detection from document image using HOG features," *Multimed Tools Appl*, vol. 82, no. 1, pp. 863–878, Jan. 2023, doi: 10.1007/s11042-022-13300-5.

[36] E. Odhiambo Omuya, G. Onyango Okeyo, and M. Waema Kimwele, "Feature Selection for Classification using Principal Component Analysis and Information Gain," *Expert Syst Appl*, vol. 174, Jul. 2021, doi: 10.1016/j.eswa.2021.114765.

[37] X. Wu, T. Oli, J. H. Qian, V. Taylor, M. C. Hersam, and V. K. Sangwan, "An Autotuning-based Optimization Framework for Mixed-kernel SVM Classifications in Smart Pixel Datasets and Heterojunction Transistors," Jun. 2024, [Online]. Available: http://arxiv.org/abs/2406.18445

[38] L. Zhu and P. Spachos, "Support vector machine and YOLO for a mobile food grading system," *Internet of Things*, vol. 13, Mar. 2021, doi: 10.1016/j.iot.2021.100359.

[39] S. M. Al-azzawi, M. A. Deif, H. Attar, A. Amer, and A. A. A. Solyman, "Hyperparameter Optimization of Regression Model for Electrical Load Forecasting During the COVID-19 Pandemic Lockdown Period," *International Journal of Intelligent Engineering and Systems*, vol. 16, no. 4, pp. 239–253, 2023, doi: 10.22266/ijies2023.0831.20.

[40] M. Hussain, S. K. Wajid, A. Elzaart, and M. Berbar, "A Comparison of SVM Kernel Functions for Breast Cancer Detection," in *2011 Eighth International Conference Computer Graphics, Imaging and Visualization*, 2011, pp. 145–150. doi: 10.1109/CGIV.2011.31.

[41] W. F. A. Maki, K. Mauladan, and I. B. Muktyas, "Date Fruit Classification using K-Nearest Neighbor with Principal Component

Analysis and Binary Particle Swarm Optimization," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 7, no. 6, pp. 1456–1463, Dec. 2023, doi: 10.29207/resti.v7i6.4839.

[42] M. Pan *et al.*, "Photovoltaic power forecasting based on a support vector machine with improved ant colony optimization," *J Clean Prod*, vol. 277, Dec. 2020, doi: 10.1016/j.jclepro.2020.123948.

[43] I. Arzate-Vázquez *et al.*, "Image Processing Applied to Classification of Avocado Variety Hass (Persea americana Mill.) During the Ripening Process," *Food Bioproc Tech*, vol. 4, no. 7, pp. 1307–1313, Oct. 2011, doi: 10.1007/s11947-011-0595-6.

[44] S. Nuanmeesri, "Utilization of Multi-Channel Hybrid Deep Neural Networks for Avocado Ripeness Classification," *Engineering, Technology and Applied Science Research*, vol. 14, no. 4, pp. 14862–14867, Aug. 2024, doi: 10.48084/etasr.7651.

AUTHOR BIOGRAPHY

Kemal Crisannaufal is an undergraduate student in Informatics at the School of Computing at Telkom University. He has joined as a practicum assistant in the Data Structure course, guiding on algorithms. During his studies, he also completed several machine learning, data science, and software engineering projects. He is interested in research on image processing, machine learning, and computer vision.

Wikky Fawwaz Al Maki received the Dr. Eng. degree in Integrated Science and Engineering from Ritsumeikan University, Japan in 2009. He also received his B.Eng. degree from University of Indonesia in 2004 and M.Eng. degree from Ritsumeikan University in 2007. He is currently a lecturer at School of Computing, Telkom University. His research areas are digital image processing, signal processing, stochastic systems, artificial intelligence, pattern recognition, and computer vision. He is affiliated with the Indonesia Data Scientist Society as a member. He has served as an invited reviewer for several international conferences and journals.